A User Comment Draft (UCD)

NTCIP 1203 v04.05

National Transportation Communications for ITS Protocol

Object Definitions for Dynamic Message Signs (DMS)

September 12, 2024

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

Revision Notes:

a) *This revision reflects all proposed changes from v04.03. Test procedures have not yet been updated*

Published by

American Association of State Highway and Transportation Officials (AASHTO)
555 Twelve Street, NW, Suite 1000
Washington, DC 20004

Institute of Transportation Engineers (ITE)
1627 I ("Eye") Street, N.W., Suite 550
Washington, D.C. 20006-4007

National Electrical Manufacturers Association (NEMA)
1300 North 17th Street, Suite 900
Rosslyn, Virginia 22209-3806

Revision History for NTCIP 1203 v05
to be removed once approved

Revision	Date	Author	Description (latest on top)
05	09/12/2024	Vaughn	Implemented comments from received from WG members
04	07/15/2024	Vaughn	Implemented additional WG input from May meeting
03	05/02/2024	Vaughn	Implemented initial set of WG comments for sections 1-5 and Annex A
02	03/18/2024	Vaughn	Second draft based on coordinating updates across all of 1200 series
01	02/10/2023	Vaughn	Initial draft document to upgrade to SMIv2 and incorporate security improvements as recommended by NTCIP 9014.

NOTICES

Copyright Notice

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

PDF File License Agreement

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

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

Data Dictionary and MIB Distribution Permission

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

You may make or distribute unlimited copies, in part or full, of the DD or MIB, including copies for commercial distribution, provided that:

a) each copy you make or distribute contains the citation “From NTCIP 0000 [insert the standard number]. Used by permission of AASHTO / ITE / NEMA.”;
b) the copies are not edited from the published version of this document.

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

AASHTO / ITE / NEMA and their suppliers do not warrant the performance or results you may obtain by using these materials. AASHTO / ITE / NEMA and their suppliers make no warranties, express or implied, as to noninfringement of third party rights, merchantability, or fitness for any particular purpose. In no event will AASHTO / ITE / NEMA or their suppliers be liable to you or any third party for any claim or for any consequential, incidental or special damages, including any lost profits or lost savings, arising from your reproduction or use of these materials, even if an AASHTO / ITE / NEMA representative has been advised of the possibility of such damages.

Some states or jurisdictions do not allow the exclusion or limitation of incidental, consequential, or special damages, or the exclusion of implied warranties, so the above limitations may not apply to a given user.

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

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

PRL and RTM Distribution Permission

To the extent that these materials are distributed by AASHTO / ITE / NEMA in the form of a Profile Requirements List ("PRL") or a Requirements Traceability Matrix ("RTM"), AASHTO / ITE / NEMA extend the following permission:

a) you may make or distribute unlimited copies, including derivative works of the PRL (then known as a Profile Implementation Conformance Statement ("PICS")) or the RTM, provided that each copy you make or distribute contains the citation "Based on NTCIP 0000 [insert the standard number] PRL or RTM. Used by permission. Original text © AASHTO / ITE / NEMA.";
b) you may only modify the PRL or the RTM by adding: 1) text in the Project Requirements column, which is the only column that may be modified to show a product's implementation or the project-specific requirements; and/or 2) additional table columns or table rows that are clearly labeled as ADDITIONAL for project-unique or vendor-unique features; and
c) if the PRL or RTM excerpt is made from an unapproved draft, add to the citation "PRL (or RTM) excerpted from a draft standard containing preliminary information that is subject to change."

This limited permission does not include reuse in works offered by other standards developing organizations or publishers, and does not include reuse in works-for-hire, compendiums, or electronic storage devices that are not associated with procurement documents, or commercial hardware, or commercial software products intended for field installation.

A PICS is a PRL which is completed to indicate the features that are supported in an implementation. Visit www.ntcip.org for information on electronic copies of the MIBs, PRLs, and RTMs.

TRF Distribution Permission

A Testing Requirements Form ("TRF") may be a Testing Requirements Traceability Table and/or Test Procedures. To the extent that these materials are distributed by AASHTO / ITE / NEMA in the form of a TRF, AASHTO / ITE / NEMA extend the following permission:

a) you may make and/or distribute unlimited electronic or hard copies, including derivative works of the TRF, provided that each copy you make and/or distribute contains the citation "Based on NTCIP 0000 [insert the standard number] TRF. Used by permission. Original text © AASHTO / ITE / NEMA.";
b) you may not modify the logical flow of any test procedure, without clearly noting and marking any

such modification; and

c) if the TRF excerpt is made from an unapproved draft, add to the citation “TRF excerpted from a draft standard containing preliminary information that is subject to change.”

Content and Liability Disclaimer

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

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

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

In publishing and making this document available, AASHTO, ITE, and NEMA are not undertaking to render professional or other services for or on behalf of any person or entity, nor are AASHTO, ITE, and NEMA undertaking to perform any duty owed by any person or entity to someone else. Anyone using this document should rely on his or her own independent judgment or, as appropriate, seek the advice of a competent professional in determining the exercise of reasonable care in any given circumstances. Information and other standards on the topic covered by this publication may be available from other sources, which the user may wish to consult for additional views or information not covered by this publication.

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

Trademark Notice

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

< This page intentionally left blank. >

ACKNOWLEDGEMENTS

This document was prepared by the NTCIP Dynamic Message Sign Working Group (DMS WG), a subdivision of the Joint Committee on the NTCIP. The Joint Committee on the NTCIP 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 development organizations (SDOs) and provides guidance for NTCIP development.

When this document was prepared, the following individuals were active members of the NTCIP DMS WG:

Roberto Acevedo
Mark Aragon
Steve Bostrom
Russell Brookshire
Patrick Chan
Matt Fomich
Terry Haukom
Milan Patel
Robert Rausch (co-chair)
John Thai (co-chair)
Troy Torgersen
Ken Vaughn
Derek Vollmer

Other individuals providing input include:

Ashraf Ahmed
Russel Allen
Kingsley Azubike
Kevin Balke
Serge Beaudry
Raymund Berlo
Ralph Boaz
Zackery Bounyavong
Brenda Boyce
Bashir Choudry
Brian Coogan
Brian Doherty
Jason Fardini
Ed Fok
Miker Forbis
Nagachand Ganapaneni
Justin Hatch
Craig Hinners
Mat Jasnosz
Sangheon Kim
Dale Kirmer
Keith Koeppen
Denver Kruse
AJ Lahiri
James Barnhart
Stephane Lapierre
Robert Lopes
Christopher Lyons
Mike Mercer
Mohammad Iraki
Alex Mousadi
Satya Muthuswamy
Siva Narla
Dennis Nguyen
Rose Parisi
Keith Patton
Raymond Poon
Leandro Prusch
Peter Ragsdale
Joerg "Nu" Rosenbohm
Brian Scharles
Jay Schultz
Jacob Shain
Kellen Shain
Doug Spencer
Gregg Thompson
James Whalen
Robert White
Brad Wilkus

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

AECOM
Apptricity
AtkinsRealis
Booz Allen Hamilton

Caltrans
City of Anaheim
Consensus Systems Technologies
Daktronics
Florda DOT
Georgia DOT
Indiana DOT
ITS Korea
Kapsch
KNB Engineering
KLD Engineering
Ledstar
Minnesota DOT
New York State DOT
Nevada DOT

Skyline Products
Noblis
NoTraffic
Oregon DOT
Parsons
Pillar Consulting
Q-Free
Southwest Research Institute
TAPCO
TransCore ITS
Trevilon
Utah DOT
Ver-Mac
Vaisala
Washington State DOT

The U.S. Department of Transportation Joint Program Office provided funding assistance for the development of this document.

FOREWORD

This document is an NTCIP Device Data Dictionary Standard. Device Data Dictionary Standards provide formal definitions of data elements for use within NTCIP systems. The data is defined using the format specified in NTCIP 8004, which is based on the formats defined in RFC 2578. This data is typically exchanged using the Simple Network Management Protocol (SNMP) as defined in NTCIP 2301.

This document identifies and defines how a management station may wish to interface with a field device to control and monitor dynamic message signs (DMS). This document defines requirements that are applicable to all NTCIP DMS and it also contains optional and conditional sections that are applicable to specific environments for which they are intended.

Approvals

NTCIP Data Dictionary Standards are separately balloted and approved by AASHTO, ITE, and NEMA after recommendation by the Joint Committee on the NTCIP. Each organization has approved this document as the following standard type:

AASHTO—Standard Specification;
ITE–Software Standard;
NEMA—Standard;

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

User Comment Instructions

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

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

A User Comment should be submitted to this address:

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

A User Comment should be submitted in the following form:

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

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

History

Version	**Date**	**Description (latest on top)**
V04	TBD	Updated to SMIv2 with related security updates, added support for multiple languages (e.g., DisplayString changed to SnmpAdminString and added utf8Style characterSet), revised RTM to reference NTCIP 1201 v04 and ISO 20684 series, and revised test procedures to reflect changes.
v03	March 2011	Added Annex C (test procedures) along with minor corrections and clarifications
v02	March 2007	Developed to reflect additional lessons learned, to incorporate better documentation (e.g., Sections 2-4, which provide the concept of operations, requirements, and dialogs) and to add new features requested by the ITS community (e.g., colors, graphics, and 3-tiered diagnostics).
v01A	July 2001	Technical corrections for objects and MULTI tags
v01	December 1999	Original version approved and published by AASHTO and ITE to become a full joint-NTCIP standard. (no change in content)
NEMA TS 3.6	December 1997	Original version approved and published by NEMA.

INTRODUCTION

This document provides definitions of data elements for use with dynamic message signs. The data is defined using the SMIv2 object-type format as defined in RFC 2578 and would typically be exchanged using SNMPv3 per the rules of NTCIP 2301. The content of one object, the dmsMessageMultiString object, uses a complex syntax called the Mark-Up Language for Transportation Information (MULTI) format, which is defined in Section 6 of this document.

This document contains two normative and one informative annexes.

The following keywords apply to this document: AASHTO, ITE, NEMA, NTCIP, DMS, VMS, CMS, data, data dictionary, object, message sign, message board, sign, MULTI.

This document uses only metric units.

CONTENTS

FIGURES

TABLES

Section 1
General [Informative]

1.1 Scope

This document specifies the logical interface between Dynamic Message Signs (DMS) and the host systems that control them (commonly referred to as “central” systems). This document describes the supported DMS functionality in terms of user needs and requirements; however, the nature of the interface is determined in part by the operational nature of the devices being controlled, and therefore this document touches on such operational issues on occasion.

This document assumes a model of DMS operation in which DMS controllers possess intelligence, and the data used for message display and sign configuration is resident at the DMS controller. In particular, data elements such as fonts, graphics, message text, time-based schedules, and so forth may reside at the DMS controller, and the controller renders messages on the sign face based on this data (This model is typical of existing DMS applications, and may be contrasted with an alternate model in which, for example, the DMS controller only knows how to display static bitmaps, and all message layout and composition is performed by the central system.). We refer to the DMS controller's status, control, and configuration data as the “controller database”; this document specifies interfaces whereby this data can be manipulated by the central system. There are no imperative commands such as “Display a message” or “Report status”; the central system controls the behavior of the DMS purely through queries of and changes to the controller database using a suite of communication protocols appropriate for the underlying communications infrastructure. These communications protocols are defined in the NTCIP 23xx series (Application Layer protocols), NTCIP 22xx series (Transport Layer protocols), and NTCIP 21xx series (Subnetwork Layer protocols).

1.2 References

The following documents are referenced by this document. At the time of publication, the editions indicated were valid.

1.2.1 Normative References

Normative references contain provisions that, through reference in this text, constitute provisions of this document. 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.

ISO/TS 26048-1: -	*Intelligent transport systems — Field equipment SNMP data interface — Part 1: Global objects*
ITU-T X.696 (a.k.a., ISO/IEC 8825-7)	*Information technology — ASN.1 encoding rules: Specification of Octet Encoding Rules (OER), published August 2015*
ISO/IEC/IEEE 24765:2017	*Systems and software engineering — Vocabulary,* 2017, available at https://standards.iso.org/ittf/PubliclyAvailableStandards/index.html
RFC 854	*Telnet Protocol Specifications*
RFC 2578	*Structure of Management Information Version 2 (SMIv2)*, April 1999
RFC 2579	*Textual Conventions for SMIv2,* April 1999
RFC 2580	*Conformance Statements for SMIv2*, April 1999
RFC 3411	*An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*, December 2002
RFC 3419	*Textual Conventions for Transport Addresses*, December 2002
AASHTO / ITE / NEMA	*Structure and Identification of Management Information (SMI)*

NTCIP 8004 v03	published

1.2.2 Other References

Other references are included to provide a more complete understanding of this document and its relationship to other documents.

AASHTO / ITE / NEMA NTCIP 2202 v02	*Internet (TCP/IP and UDP/IP) Transport Profile*, published
AASHTO / ITE / NEMA NTCIP 2301 v02	*Simple Transportation Management Framework (STMF) Application Profile (AP) (AP-STMF)*, published
AASHTO / ITE / NEMA NTCIP 8007 v02	*Testing and Conformity Assessment Documentation within NTCIP Standards Publications*, published
AASHTO / ITE / NEMA NTCIP 9001 v04	*The NTCIP Guide* published July 2009
ANSI X3.4-1986	*Information Systems - Coded Character Sets - 7-Bit American National Standard Code For Information Interchange (7-Bit ASCII), 1986*
Architecture Reference for Cooperative and Intelligent Transportation (ARC-IT), Version 9.2	*National ITS Architecture, FHWA, 2023*
ISO/IEC 10646:2020	*Information technology — Universal coded character set (UCS)*
ISO 15784-2:2024	*Intelligent transport systems — Data exchange involving roadside equipment communication — Part 2: Centre to field device communications using SNMP*
ITS Standards Training Courses	*https://www.its.dot.gov/pcb/ITSCourses/Default.aspx#training*
OMG Unified Modeling Language Specification, Version 1.5	*OMG Unified Modeling Language Specification, Object Management Group, 2003*
Unicode 15.1.0	*The Unicode Standard Version 15 — Core Specification. 2023 September 12*

1.2.3 Contact Information

1.2.3.1 Architecture Reference for Cooperative and Intelligent Transportation (ARC-IT)

The latest version of the National ITS Architecture, as defined by ARC-IT, may be viewed on-line at http://arc-it.net

1.2.3.2 IAB Documents and RFCs

For Internet Architecture Board (IAB) documents, contact:

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

1.2.3.3 ISO Documents

For ISO documents, contact:

International Organization for Standards (ISO)
www.iso.org
www.iso.org/obp
https://standards.iso.org

The main ISO website (www.iso.org) can be used to search for and purchase standards among other things, including using for free.

The normative refernces within NTCIP only refer to the materials available for free on the ISO Standards Maintenance Portal (https://standards.iso.org) and the online browsing platform (www.iso.org/obp), which provides a searchable database of terms and their definitions.

1.2.3.4 ITU-T Documents

For ITU-T documents, contact:

International Telecommunications Union,
Telecommunication Standardization Sector (ITU-T)
www.itu.int/en/ITU-T

1.2.3.5 NTCIP Documents

Copies of NTCIP documents may be obtained from:

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

1.2.3.6 Object Management Group Documents

Copies of OMG standards may be obtained electronically from the Object Management Group at http://www.omg.org

1.3 General Statements

1.3.1 SNMPv3

The information defined in this document can theoretically be exchanged over any communications stack, but is designed to be exchanged using SNMPv3. Within the context of this document SNMPv3 refers to a complete communications stack, as described in NTCIP 2301 v03 and ISO 15784-2:2024, which (among other requirements) require support for Transport Layer Security (TLS) version 1.3 and the TLS Transport Model (TLSTM). TLSTM defines how the SNMP engine uses the TLS certificate to manage SNMP data access.

1.3.2 Compatibility with Versions Prior to NTCIP 1203 v04

Prior to v04, NTCIP 1203 was intended for use with SNMPv1, which has known security vulnerabilities. Starting with v04, NTCIP 1203 is intended for use with SNMPv3, which addresses the known security vulnerabilities with SNMPv1 with minimal redesign but is not directly compatible with prior implementations using SNMPv1.

Because the migration to SNMPv3 broke direct backward compatibility, the NTCIP community agreed to address a number of other known technical issues with NTCIP data to further address security, ambiguity, code reusability, longevity and other issues. Most of these issues were related with NTCIP 1201 and NTCIP 1103, which prior versions of this document normatively referenced. These references have been replaced with references to ISO 26048-1. In addition, v04 deprecated and replaced some objects from prior versions of NTCIP 1203.

This document does not define rules for exchanging any data that was deprecated in NTCIP 1203 v04 or earlier. Implementations that need to interoperate with equipment that conforms to versions of this document prior to NTCIP 1203 v04, should refer to those prior versions for specific requirements, especially in relation to data from NTCIP 1103 and NTCIP 1201.

Most of the objects directly defined in prior versions of this document were upgraded without significant compatibility issues. Deprecated objects are retained within the Management Information Base (MIB) in order to provide an unambiguous definition of how they should be presented if they are exchanged using SNMPv3 (e.g., this might be done when communicating to a proxy agent that provides an SNMPv3 interface for a native SNMPv1 device).

1.3.3 Use of SMIv2

SNMPv3 is designed to exchange data defined using the Structure of Management Information (SMI) Version 2 (v2) as defined in RFC 2578. Versions of this document prior to NTCIP 1203 v04 defined data using SMIv1, as defined in RFC 1212. The data definitions also conform with NTCIP 8004, which defines additional rules (e.g., the subclauses used within the DESCRIPTION clause. This document updates the MIB to conform to SMIv2 and the current version of NTCIP 8004 while retaining the object identifiers of previously declared object types. In some cases, minor revisions have been made to objects; when these have any technical impact, the issue is identified in the Informative subclause of the object type description and in the object refinement table in Annex A, when necessary.

1.3.4 Presentation of MIB and References

This document contains a MIB as well as references to elements defined within the MIB. This document uses the `Courier New font` to distinguish this text from normal English text.

1.3.5 Superseding and Supplanting Objects

Many of the object types defined in this document identify superseding or supplanting object types by using a "<Superseded by>" or "<Supplanted by>" clause. Supplanting object type(s) provide similar functionality while superseding objects provide a new interface for the same device capability.

For example, `eventConfigMode` (defined in NTCIP 1103 v03) is supplanted with `fdCondTriggerMode` (defined in ISO 26048-1). Although both objects provide similar functionality (i.e., identifying a type of condition to trigger an event), setting an instance of `eventConfigMode` does not affect any instance of `fdCondTriggerMode` in an implementation that supports both objects because the associated event logs are defined separately.

By contrast, `globalTime` (defined in NTCIP 1201 v03) is superseded by `fdClockUtcDate` and `fdClockUtcTime`. In other words, if a device supports both `globalTime` and `fdClockUtcTime`, setting `globalTime` will affect the time reported by `fdClockUtcTime`. The core device capability is the same (e.g., keeping time), but there is a new interface that provides greater range and resolution.

1.4 Terms

The following glossary exhibits many DMS-related terms and abbreviations used in the ITS industry in an attempt to support the standardization of DMS-related terms. For terms not defined here, English words are used in accordance with their definitions in the latest edition of Webster's New Collegiate Dictionary. Electrical and electronic terms not defined in this section or in Webster's New Collegiate Dictionary are used in accordance with their definitions in ISO/IEC/IEEE 24765:2017.

activate	The action of placing a message in the current buffer and performing the logic of running the message. Contrast with 'Display', which manipulates the sign display to make the current message visible to the driving public.
activate message	The command to direct the sign controller to display the message on the sign face.

activation priority	A numeric value between 1 and 255 that the controller compares to the Run-time Priority of the current message. If the Activation Priority is greater than or equal to the Run-time Priority of the current message, the controller can replace the message. If the Activation Priority of the new Message is less than Run-time priority of the current message the controller rejects the activation of the new message.
alternating message	A message that contains more than one page of information/text.
ambient light level	The amount of light surrounding the sign location.
ASCII	American Standard Code for Information Interchange, a 7-bit wide code used to represent a character set as defined by ANSI X3.4.
attribute	Shorthand notation for Message Attribute. Defines how a Message is displayed. See Message Attribute.
axial intensity	The brightness of light on the axis horizontally and vertically perpendicular to the sign face.
backup lamp	In a two lamp system, the secondary lamp that is used when the Primary Lamp has failed. Also, it may be turned on with the primary/normal lamp to create an over-bright illumination of the message.
beacon	A device that directs light in one direction and flashes (Similar to a one-section traffic intersection signal head). The device is intended to increase a driver's attention to a message. The color is undefined (also known as Strobe or Strobe Light).
bitmap	A digital representation of an image having bit reference pixels.
bitmapped object	An object that is represented as a series of bits, each having a boolean value. Bitmapped objects can have a SYNTAX of integer, octet string, or BITS. The ordering of bits is defined by the encoding rules for the SYNTAX.
blank message	A message that is devoid of informational content (blank) and the sign face is clear (all pixels off, or shutters closed depending on the display technology).
blank sign	A command or condition caused by a user command, error or fault condition, or default state in which a sign is not displaying a message, and depending on the display technology of the sign, has turned off lamps, LED drivers, etc.
blank-out sign	A type of DMS that has the capability to show a blank message or one fixed message.
border	The blank area between the outer most edge of the sign face and the outermost edge of the sign housing.
brightness	See Luminance.
brightness control	A term that defines how the light intensity of a sign is determined/set. Automatic control uses local detection of ambient light to determine the brightness level of the sign, whereas manual control defines the brightness level by a control command.
brightness level	The intensity of the light used to form a message or that would be used to form a message if one is not currently displayed. Usually selected in one of several ways. Some examples: NONE ON / OFF DAY / NIGHT/ OVERBRIGHT x of y levels a percent of maximum brightness output level
bulb matrix	See Lamp Matrix
cabinet	An enclosure that protects the device's controller from the elements.

candela (cd)	An SI unit of measure for luminance abbreviated cd.
central computer	A computer system that operates as a control source for one or more signs in the signage system. A computer/server that is host to its signs, also referred to as the host or central computer. The signage system may be controlled by central computers installed in more than one location. Or, it may be a remotely located central computer capable of managing the operation of one or more signs. Abbreviation is CC.
central control computer	See Central Computer.
central control mode	A state whereby control of the sign from the Central Computer. Preferred term for remote control mode.
central override Mode	A state whereby commands from Local Control Panel are ignored.
central system (sign management software)	The software that operates on the Central Computer controlling/monitoring signs.
changeable memory	A generic term for a type of memory that allows a user to modify the content. The content of the memory is not lost when power is turned off. See also 'Permanent Memory', 'Non-Volatile Memory' and 'Volatile Memory'.
changeable message sign	A sign that is capable of displaying one of two or more predefined messages, or a blank message. Abbreviated CMS. The capabilities associated with a CMS are: - drum sign with several faces, or pixel matrix - several predefined message - downloading of new messages, graphics or fonts not possible - uploading of messages and graphic definition possible - blank message possible - all messages are defined - may support more than a monochrome color scheme (each drum face may have a different color scheme, each face may have multi-color text) - error report capabilities similar to VMS - exercising of pixels
changeable messages	A library of messages stored in non-volatile, memory/storage devices. See also Permanent Messages and Volatile Messages.
character	One symbol from a specific alphabet, font or character set.
character font	See Font.
character group	See Character Module.
character height	The vertical pitch times the number of pixels in the column of pixels.
character matrix sign	A DMS sign that uses character matrixes with a fixed amount of blank space (no pixels present) between character matrixes to achieve the inter-character spacing. There is also blank space (no pixels present) between lines of characters to achieve the inter-line spacing.
character module, N	Component required to display N characters. This includes, but is not limited to, a subset of the following items based on the display technology of the sign: lamps, fiber, shutter, color filter, LED's, and frame to hold all of the above parts together as one unit.
character size	See Character Height.
character spacing	The spacing, in pixels, between two characters in line matrix or full matrix signs. The fixed amount of space between two characters on a character matrix sign.
character width	The horizontal pitch times the number of pixels in the row of pixels.

characters per line	The number of characters that can be displayed on one line. Used in character oriented signs. Line matrix and full matrix signs are described as n columns (pixels) wide.
checksum	A data error-detection scheme. The result of an algorithm performed on a block of data.
climate-contol	The ability to control the temperature and other factors affecting the environment in which the DMS electronics operates.
color	The chromaticity specified in terms of the CIE 1931 Colorimetric System. A visually perceived characteristic of light, specified at a particular wavelength in nanometers. Color is one attribute used to display a message. Depending on the display technology of the sign, the color used to display a message may be fixed or selectable.
column	A vertical line of pixels.
communication failure	The condition when a central computer cannot communicate with the sign controller due to errors or malfunctions.
communication interface	The communication port(s) on the controller used to communicate with other device(s).
compatible	The ability of two or more systems or components to exchange information (IEEE Std. 610.12-1990: IEEE Standard Glossary of Software Engineering Terminology).
cone of vision	The geometric figure (cone) used to define the area in which a message on a sign can be legibly viewed. It is measured in degrees. It is twice the angle from the axis of the pixel to the 50% brightness point on an LED display. The cone of vision is also known as the “viewing angle”.
configuration	The setting of the parameters within the controller to operate the sign with a defined set of ranges, parameters and functions.
configure	To change one or more settings in the device.
consistent	The ability of two or more systems or components to exchange information and use the supported information that has been exchanged and gracefully reject any unsupported information according to defined rules.
contrast ratio	The amount of measured light emanating from the message divided by the amount of measured light reflected from the background.
control mode	Defines the current method by which the sign controller receives instructions.
controller	See Sign Controller.
controller address	See Sign Address.
controller failure	The condition caused when the DMS Controller does not properly perform its intended functions.
controller reset	A function that restarts the controller from an initialization process. This may be activated via time-outs of an event (watchdog, power loss), local reset button, or software command.
current	a.) Reflecting the conditions at the present time (or at the time at which the data is time stamped) as determined by the controller. b.) The amount of electric charge flowing past a specified circuit point per unit time
cyclical redundancy check	A data error-detection scheme. A polynomial algorithm is performed on a block of data. There are different algorithms involving a different number of bits and bytes in the calculation such as CRC-16 and CRC-32. *(see also Frame Checking Sequence)*

default message	Under normal operating conditions, this term specifies the neutral message. Under default conditions, communications failure, power loss, power recovery and communications time-out, the default message is the message displayed as defined by the corresponding objects (see Section 5). These may or may not be the same as the neutral message.
default state	A defined mode of operation assumed when no other instructions have been received.
deprecated	This term is defined in NTCIP 8004 v02.
depth	The distance between the front and back of a sign or other enclosure. It can be measured as both inside and outside dimension.
determine	To read information from the device.
diagnostics	A set of routines operated in the controller used to verify the proper operation of the DMS components.
display	To reveal a message to the traveling public once it has been activated. Also see related terms: sign face (a DMS component), activate message (a command), and message (the image)
display activation time	The length of time required to display a page of text on the sign once the complete command has been received by the controller.
display module	See Character Module.
display technology	The means used to present a message, e.g., shuttered fiber, LED, flip disk, lamp matrix, combination of the two, etc.
display times	The time parameters within a message attribute.
DMS controller	See Sign Controller.
DMS housing	The enclosure that environmentally protects the components of the Dynamic Message Sign.
DMS manufacturer	The company that maintains a factory and staff that develops, engineers, and manufacturers the complete DMS sign assembly and DMS Controller from raw materials and components.
dot	One pixel in a display matrix.
download	To transfer information from the central computer into the referenced field device.
drum	The multifaceted cylinder, with associated lighting, motor/brake drive unit and position sensing switches that rotates to display one face to the motorist.
drum sign	A type of CMS using one or more drums to display a message.
dynamic message sign	Any sign system that can change the message presented to the viewer such as VMS, CMS and BOS. It includes the following major components: sign face, sign housing, controller, and, if present, the controller cabinet. Abbreviated DMS.
electrically erasable programmable read only memory (EEPROM)	. A variation of an EPROM chip in that instead of erasing the memory by placing it under UV light, portions of the chip can be erased electrically, and thus does not need to be removed from the circuit, provided the circuit supports erasing the chip.
electromagnetic shutter	A device that can be positioned via a pulse of electricity, and stay in the desired position due to an internal magnet.
environmental controls	Equipment to control the temperature and/or humidity within an enclosure, typically the sign housing and/or controller cabinet. This can include fans, heaters, thermostats, humidistats, override timers, motorized louvers, filters, ducting.

erasable programmable read only memory (EPROM)	Erasable Programmable Read Only Memory. A variation of a PROM chip where the contents can be changed by erasing the chip with a UV light eraser and then programming the chip again.
external device	A component that is not normally considered part of a DMS, but is connected to the DMS by some interface.
external illumination	A light source shining on the face of the sign so that its message may be read by the motorist.
external input	The communication interface with an external device.
extinguishable message sign (EMS)	See Blank-Out Sign.
feature	A service provided by / behavior of the device.
fiber optic	A slender thread-like strand of material used to carry light.
fiber optic bundle	Many fiber optic strands combined into one larger group. A fiber optic bundle terminates at one end on the sign face, the other end terminates at the light source.
fiber optic harness	A number of fiber optic bundles grouped together with one common end. The common end is inserted in the lamp module.
fiber optic sign	A light emitting sign whose pixels are made of ends of fiber optic bundles.
fiber optic/flip disk hybrid	A reflective flip-disk type of sign that employs a fiber optic display technology in addition to the reflective flip disk.
firmware	The logical programming stored in a controller's memory to operate the controller.
flash EPROM	A type of EEPROM with rapid programming capability.
flasher	A device that causes beacons to flash.
flashing	A message attribute causing all or parts of a message to turn on and off.
flashing beacons	See Beacon.
flashing display	A one page message that alternates between on and off.
flip disk	A two-state display technology using an electro-mechanically actuated disk for each pixel position. One side of the disk displays the ON state of the pixel and another side represents the pixel's OFF state.
flip LED	A hybrid display technology that combines flip-disk and LED technology.
font	A type style for a set of characters (letters, numbers, punctuation marks, and symbols).
forced air cooling	A device used to reduce the temperature within an enclosure or housing by moving air.
forced air ventilation	A device used to force out the air inside the enclosure or housing and introduce new air from the outside.
frame	See Page.
frame checking sequence (FCS)	Defines the value to be used within data packet frames for error detection. Implementations claiming conformance with this standard shall use the default International Telegraph and Telephone Consultative Committee (CCITT) 16-bit FCS as defined in ISO/IEC 13239:2002. Note: In this document, object definitions representing a CRC value rely on the ISO/IEC 13239:2002 definition; the FCS is determined most significant byte first, but transmitted least significant octet first.
front	The side of the sign containing the visible message.
front access	Access to the internal components of the sign accomplished via access panels or access doors located on the front of the sign.

full matrix	A type of VMS with the entire display area containing pixels with the same horizontal pitch and the same vertical pitch without fixed lines or characters. A full matrix sign s characterized by its ability to address and change each pixel independently.
full standardized range	The range of values identified and fully specified within a standard. Values left for proprietary use (e.g., the value 'other' in enumerated lists) are not a part of the Full Standardized Range since the meaning of the value is not 'fully specified'.
graphic	An image that is stored within the controller's memory and can be inserted into a message.
graphical user interface	The presentation of information to the user on a screen in graphic format.
host computer	See Central Computer.
housing	The enclosure of the sign containing the display elements.
illumination power	The energy source for message illumination.
intensity	The brightness of light emanating from the display, expressed in candela per unit area.
interchangeability	A condition which exists when two or more items possess such functional and physical characteristics as to be equivalent in performance and durability, and are capable of being exchanged one for the other without alteration of the items themselves, or adjoining items, except for adjustment, and without selection for fit and performance. (National Telecommunications and Information Administration, U.S. Department of Commerce)
interoperable	The ability of two or more systems or components to exchange information and use the information that has been exchanged (IEEE Std. 610.12-1990: IEEE Standard Glossary of Software Engineering Terminology).
interface	An interface is a named set of operations that characterize the behavior of an element (Unified Modeling Language Specification)
inter-line spacing	The amount of vertical space between two lines. The distance from the bottom of the bottom pixel on a line to the top of the top pixel on the line immediately below. On full matrix signs, it is measured as the number of pixel rows between lines of characters.
internal illumination	A light source within the sign housing that shines through the front of the sign, so that its message may be read by the motorist.
internal lighting	The lighting used for maintenance inside an enclosure or housing, independent of message illumination.
interoperability	The ability of two or more systems or components to exchange information and use the information that has been exchanged (IEEE Std. 610.12-1990: IEEE Standard Glossary of Software Engineering Terminology).
lamp	A light source used to illuminate the utilized pixels other than on a pixel-by-pixel basis. Fiber optics technology uses lamps to illuminate bundles of pixels.
lamp control Module	The device used to control the power going to the lamps.
lamp driver Module	An electronic board that directly supplies or disconnects power to the lamps to turn them on or off.
lamp driver System	See Lamp Control Module.
lamp matrix	A type of display technology where an incandescent light source is used for each pixel.
lamp status	The feedback data which indicates the operational status and condition of the lamp circuit.

Commented [MMS1]: Under Luminance it is "candela per square foot" or "candela per square meter".

LAN	Local Area Network – An intelligent control network that facilitates communication between devices that sense, monitor, communicate and control.
lane-use control sign	Overhead sign having displays that permit or prohibit the use of a lane or that indicate impending prohibitions of use [excerpted from MUTCD, clause 4E-8]. A sign that contains multiple symbols to indicate the permissive use of the lane in the direction of travel. Abbreviated LCS.
LED driver module	An electronic board that contains the control and memory elements to provide the signals to switch the LED pixel state, and which detects the operation of each individual pixel that it controls.
LED sign	A sign with pixels made from LED's.
LED/flip-disk hybrid	A type of VMS display technology that forms pixels with a combination of LED and flip disk technology. The LED is used for night viewing and the flip disk is used for daytime viewing.
legend	Unchangeable text on a sign face.
legibility distance	The 85 percentile distance at which people with 20/20 corrected vision can read the display.
light emitting diode	A type of display technology using a semi-conductor device that emits a point of light in a controllable manner. The characteristics of the point of light are determined by the type of LED used, e.g. color, cone of vision, luminance, etc. Abbreviated LED.
light output level	See Brightness Level.
line	A horizontal row of character modules (character or line matrix signs) or number of rows of pixels (full matrix signs) used to display text.
line matrix	A type of VMS sign that has no hardware defined blank spaces (no pixels) between characters. The entire line contains columns of pixels with a constant horizontal pitch across the entire line.
local control mode	One of several possible control modes to control a DMS. Local control mode is the primary control mode from the local control point (this could be a Local Control Panel or a locally connected device such as a laptop or a Personal Digital Assistant (PDA)).
local control panel	A system of switches or a keyboard located at the DMS that allows a person on-site control of the DMS, as opposed to control from a remote location via external communication.
locally activated messages	Stored messages, which are activated from the local control panel.
lumen	The unit of luminous flux emitted in a solid angle of one steradian by a uniform point source that has an intensity of one candela.
luminance	The intensity of light per unit area at its source. Usually measured in candela per unit area.
lux	A measurement of light. A unit of luminance produced on a surface area of one square meter by a luminous flux of one lumen uniformly distributed over the surface. (1 lux = 1 lumen per sq. meter)
magnetic memory	Memory based on magnetic power to keep an object in a desired position without the use of continuous electrical power.
maintenance computer	See Portable Maintenance Computer.
maintenance portable computer	See Portable Maintenance Computer.
manage	To monitor, command, and/or control.

management information base	Set of object definitions that define the attributes, properties and controllable features of devices on a network, which can be remotely monitored, configured and controlled. The information is provided in a format called Abstract Syntax Notation. 1 (ASN.1), which is an international standard for defining objects.
management station	A computer or computer network that can interact with the device via the defined interface to realize the features of the device.
Mark Up Language for Transportation Information	Name of format of the textual part of a message. The format is defined in Section 6 of NTCIP Standard 1203 version 2 (Section 3 of NTCIP 1203: 1997). Abbreviated MULTI.
master computer	See Central Computer.
master computer software	See Central Computer.
master controller	Obsolete term for central computer.
master/slave	The master is the controlling entity on a data link. It can give permission to any slave on the same link to transmit data. A slave transmits data only in response to permission from the master and it returns control to the master after finishing a transmission.
matrix	An array of pixels that can display an image.
matrix sign	A DMS that uses an array of pixels to display a message or part of a message (e.g., a line or character). Matrix signs are typically VMS because the pixel array allows for a large variety of possible displays.
message	The information to be displayed to the traveler and how it is to be displayed.
message attribute	The characteristics that define how a message shall be displayed. This includes how many pages of text, the amount of time each page is displayed, any flashing of text, the flashing time characteristics, and color definition. Not all technologies / manufacturers support all display attributes. Specific support for these items is based on the type of display technology and manufacturer.
message command	A controller command to activate a message on the sign.
message display Time	See Message Duration.
message duration	The time from message activation to message deactivation.
module	One assembly of components, like several similar assemblies, that each fit together to make one larger single unit with a unique purpose.
multi-drop	A communications architecture where multiple devices share a common communications channel.
multi-message sign	See Changeable Message Sign.
multi-page message	A message that has more than one page of text / graphics.
neutral message	A predefined generic message that is displayed when the sign is not commanded to show time-sensitive information.
neutral state	When a sign is blank or displaying a neutral message.
non-volatile memory	A generic term for memory that does not lose its content when power is turned off. See also Changeable Memory, Permanent Memory and Volatile Memory.
normal lamp	See Primary Lamp.
NTCIP	National Transportation Communications for Intelligent Transportation System Protocol
NVT-ASCII	A specific subset of ASCII that is used on network virtual terminals and is limited to character codes 0, 7-13, and 32-126. The value 0 indicates a string termination. Defined by RFC 854.

object	A data structure used to monitor or control one feature, attribute or controllable aspect of a manageable device.
obsolete	This term is defined in NTCIP 8004. .
off-axis angle	The angle from the optical axis of the LED, at which, the luminous intensity is one-half that at the optical center.
operator	An individual who needs to interact / interface with the device via the central system software to control and/or monitor its operations.
optical center	The point on an LED or output end of a fiberoptic bundle where luminous intensity is at its maximum.
optical fiber	See Fiber Optic.
page	The information that can fit on a sign at one time, together with its message attributes.
permanent memory	A generic term for memory that cannot be changed without physically replacing hardware components. See Changeable Memory, Non-Volatile Memory, and Volatile Memory.
permanent messages	A library of stored messaged in read-only devices. See also Changeable Messages and Volatile Messages.
phase	See Page
photo sensor	A light measuring device used to quantify the ambient light conditions at the sign.
photocell	See Photo Sensor.
photoelectric cells	See Photo Sensor.
physical address	The Data Link identifier which differentiates a field device in a multi-drop or point-to-point communication circuit, to allow the central computer to communicate with a specific field device. Also see 'sign address'
pitch	The center-to-center distance between two adjacent pixels, that is measured either horizontally or vertically.
pixel	The smallest independently controllable visual element of a VMS.
pixel service	A generic term for a cyclic maintenance service that exercises mechanical pixels to prevent sticking. The service may or may not be enabled during the display of a particular message.
point-to-multi point	A communications architecture that supports communications between a central system and many devices. Also called multi-drop communication.
point-to-point	A communications architecture that supports dedicated communications exclusively between two devices.
portable maintenance computer	A portable computer running maintenance software. It can communicate with a sign controller, control activation of the sign, and perform diagnostics on the controller. Abbreviated PMC.
portable remote computer	A portable computer running as a remote computer.
primary lamp	In a two-lamp system, the lamp that is turned on first.
primary/secondary	See Master/Slave.
PROM	Programmable Read Only Memory. A semiconductor device, memory chip that can be programmed once with a specific data set via a specialized electronic instrument, PROM programmer. The data programmed into the chip cannot be altered once it has been programmed.
protocol	A specific set of handshaking rules, procedures and conventions defining the format, sequence and timing of data transmissions between devices that must be accepted and used to understand each other.

random access memory	Memory that can be independently accessed at any location in a sequential or non-sequential order. Depending on the technology, the content of memory may be lost when power is turned off. Abbreviated RAM.
recovery	The action(s) performed by a controller to restore normal operation after an interruption disrupts or terminates the controller's normal operation.
remaining message display time	The amount of time before the message currently being displayed is turned off.
remote computer	A computer that can access the central computer from a remote location.
remote computer software	The application software that runs on the remote computer enabling it to communicate with the central computer's software.
remote control mode	See Central Control Mode.
requirement	A requirement describes a condition or capability to which a system must conform; either derived directly from user needs, or stated in a contract, standard, specification, or other formally imposed document. A desired feature, property, or behavior of a system.
requirements traceability	The ability to follow or study the logical progression among the needs, requirements and design details in a step-by-step fashion.
reset	See Controller Reset.
resident software	The software located in the controller. See also firmware.
retired	Within the SYNTAX field of an 'OBJECT-TYPE' macro, a status term used to classify an enumerated value for those values found to be flawed, or not useful, or no longer relevant. The term is only used within object definitions with enumerated values. Retired values shall always be included in the lists of values.
return	When discussing device requirements for providing data when an external system requests it, the term 'return' shall be understood that the data is sent to the requester.
rotate	To move a shutter to its opposite state (open or closed). To move a drum to the next position.
rotational shutters	A type of shutter that spins in one direction on an axis perpendicular to the light blocking device.
rotor	The motor/brake drive unit and position sensing switches that rotates to display one face of a drum to the traveler.
run-time priority	A numeric value between 1 and 255 that the Controller uses to determine the importance of a message, 1 lowest and 255 highest. To activate a new message, the Activation Priority of the new message must be greater than or equal to the Run-time Priority of the current message. If the Run-time Priority of the current message is greater than the Activation Priority of the new Message, the controller rejects activation of the new message.
scenario	A preset plan which assigns specific displays or actions to a specific sign or device when a predefined condition is detected. Also known as sequence.
schedule	A mechanism by which an operator can define times in the future at which the controller performs actions.
secondary lamp	The lamp that is turned on to replace a failed primary/normal lamp (see Backup Lamp). Also turned on with the primary/normal lamp to create an over-bright illumination of the message.
semi-graphic character	A character font that contains graphic shapes that fit within a character matrix.
sequence	See Scenario.

shutdown power	A type of power that is often referred to as 'last breath power'. The exact number of minutes/seconds associated with this type of power are not defined, but it must be sufficient to allow the device's computer to save the already collected data and to safely boot down.
shutter	A non-reflective device that either completely occludes or completely allows light from a light emitting pixel. Note: A shutter and a flip disk are not to be intermingled or confused.
shutter driver module	An electronic board that supplies the low voltage pulses to move the shutters into their open or closed positions.
shutter power supply module	An electronic board that supplies and monitors power to the shutters.
shuttered fiber	A type of DMS display technology using shutters and fiber optic.
sign	The sign housing, all of its contents, and all items attached to the sign housing that are used as part of the sign (e.g. photo sensors, contrast shields, static message signing, beacons, etc.).
sign access	The approach direction or mechanism used to gain access to the internal components of the sign, e.g. front, rear, walk-in.
sign address	A unique value assigned to each device on a communication channel. Used to identify the device for which the data packet is intended. Also called controller address, drop address.
sign controller	A device used to control and monitor the operations of a sign. It can have a variety of control interfaces, such as a local control panel, a local portable maintenance computer, or a central computer. The equipment within the controller is not specified by this term.
sign erasure	The act of clearing a message from the sign face.
sign face	The portion of the sign that can be controlled by the user or a management station.
sign height	The vertical dimension of the sign face.
sign housing	The sign face enclosure.
sign off	The state in which the sign is not displaying a message and all message drivers (lamps, LED drivers, etc.) are turned off. This is different from a display that contains all spaces.
sign status	The feedback data returned from the sign controller that indicates the operational condition of the sign, or the sign's components.
sign subsystem	A primary component of the DMS that can be separately monitored.
sign width	The horizontal dimension of the sign face.
sign writing	The process of changing a sign from its previous state to displaying a message.
spacing	The blank area between 2 adjacent characters. This is a hardware defined fixed distance in character matrix signs. In a line matrix sign, the horizontal (inter-character) spacing is variable and controlled by the controller software and the pixel spacing of the sign. In a full matrix sign, both horizontal inter-character and vertical inter-line spacing is variable and controlled by the controller software and pixel spacing of the sign.
specification	The project-specific detailed requirements for a DMS to be purchased by an agency or a statement by a manufacturer defining the detailed features provided by the DMS. Within this document, 'specification' often refers to the text contained in the 'Additional Project Requirements' column of the PRL.

start-up state	Either a blank message, a default message or the last valid display before the start-up.
static display	A message that uses only one page of text.
static message panel	See Legend.
status	The current condition of a referenced function or device.
stored messages	All messages loaded in a sign controller's memory.
strobe	A form of a Beacon.
strobe light	See Strobe.
stroke width	The width or diameter of a pixel.
sub-feature	A service that is part of a larger service. A specialization of a more generic feature.
supersede	To replace with a new interface to the same functional capability. The values of the original object and its superseding object are interlinked but typically provide different representations.
supplant	To replace with a new but similar functional capability. The values of the original object and its superseding object are not linked and do not impact each other.
supplemental beacon	See Beacon.
temperature sensor	A device used to measure the temperature and report it to another device.
temporary memory	A sign controller's storage area, or memory, that contains a message or message library that can be manipulated while the controller is operating on-line. This feature enables a central computer to download and update a message or message library into the sign controller.
text (sign text)	The characters used to create a message, without any information on how the characters are displayed.
traffic management center (TMC)	The location of the central computer and equipment which allows operations staff to monitor and manage traffic through roadside field devices (e.g. vehicle detectors, VMS, etc.). Abbreviated TMC.
traffic operations center (TOC)	See Traffic Management Center. Abbreviated TOC.
traveler	A person that is using the publicly accessible transportation network.
upload	To transfer information from the referenced field device to the central computer, or an attached portable computer.
user	A person who uses the system that is developed.
user need	The business or operational problem (opportunity) that must be fulfilled to justify purchase or use. While this is termed a 'user need' within the NTCIP community, it reflects needs of all stakeholders.
unicode	A character numbering scheme that supports virtually all languages and all characters for modern languages within the first 65535 code points. Defined by Unicode 15.1.0.
UTF-8	a variable-length character encoding format for the Unicode character set that is compatible with ASCII for the first 127 code assignments. Defined in ISO/IEC 10646.
validate	To ensure that an item of interest is as intended. For example, to ensure that a graphic has been stored without any errors.
variable message sign	A type of DMS, which allows a user to create and download the message to be displayed into the temporary memory area of the sign controller. Abbreviated VMS.
ventilation	The process of replacing existing air with new air. Typically done to cool the enclosure (sign housing, controller cabinet).
viewing angle	See Cone of Vision.

visibility	The ability to view an object. The greatest distance at which the sign can be seen without the aid of any instruments. This term does not reflect Legibility.
volatile memory	A generic term for memory that allows a user to modify the content, however loses its content when power is turned off. See also Changeable Memory, Permanent Memory, and Non-Volatile Memory.
volatile messages	A library of messages stored in read-write memory devices that lose their data upon loss of power. See also Changeable Messages and Permanent Messages.
watchdog	Circuitry that monitors the controller software and firmware for a stall condition. While the DMS Controller is powered on, the software polls the watchdog and resets the timing circuitry. If the watchdog circuitry times out without being reset by the software, the watchdog counter is incremented and the controller hardware is reset to clear the potential stall condition.
X by Y character matrix	An array of pixels, X columns wide by Y rows high, used to display a single character. The pixels are based on the display technology of the sign, fiber optic, LED, bulb, flip disk, etc. A single character module having 5 columns and 7 rows of pixels could be called a “5 by 7 character module” or a “5x7 character module”.

1.5 Abbreviations

The abbreviations used in this document are defined as follows:

ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange, a 7-bit wide code used to represent a character set.
ASN.1	Abstract Syntax Notation One
BOS	Blank-Out Sign
CMS	Changeable Message Sign
CRC	Cyclical Redundancy Check
DMS	Dynamic Message Sign
DUT	Device Under Test
EEPROM	Electrically Erasable Programmable Read Only Memory
FCS	Frame Checking Sequence
HOV	High Occupancy Vehicle
IAB STD	Internet Activities Board Standard
IP	Internet Protocol
ISO	International Organization for Standardization
ITS	Intelligent Transportation Systems
LCS	Lane Use Control Sign
LED	Light Emitting Diode
LUS	Lane-Use Control Sign
MIB	Management Information Base
MULTI	Mark Up Language for Transportation Information
NVT-ASCII	Network Virtual Terminal American American Standard Code for Information Interchange
OID	OBJECT IDENTIFIER
OER	Octet Encoding Rules
PMPP	Point to Multi-Point Protocol
PRL	Profile Requirements List
PROM	Programmable Read Only Memory

RAM	Random Access Memory
RFC	Request for Comments
RPM	Revolutions Per Minute
RTM	Requirements Traceability Matrix
SNMP	Simple Network Management Protocol
STMF	Simple Transportation Management Framework
T2	Transportation Transport Profile
TCP	Transmission Control Protocol
TMC	Traffic Management Center
TMP	Transportation Management Protocol
TOC	Traffic Operations Center
UDP	User Datagram Protocol
UTF-8	Unicode Transformation Format – 8-bit
VMS	Variable Message Sign
WG	Working Group

Section 2
Concept of Operations [Normative]

This section defines user needs that this document addresses in subsequent sections. Accepted system engineering processes detail that requirements should only be developed to fulfill well-defined user needs. The first stage in this process is to identify the ways in which the system is likely to be used. For this document, this entails identifying the various ways in which transportation operations personnel may use DMS information to fulfill their duties.

This concept of operations (ConOps) provides the reader with:

a) A detailed description of the scope of this document;
b) An explanation of how a DMS device is expected to fit into the larger context of an ITS network;
c) A starting point in the procurement process; and
d) An understanding of the perspective of the designers of this document.

Section 2 is intended for all readers of the document, including:

a) Transportation operations managers
b) Transportation operations personnel
c) Transportation engineers
d) System integrators
e) Device manufacturers

For the first three categories of readers, this section provides a useful understanding of how DMS equipment can be used in their system. For this audience, this section serves as the starting point in the procurement process. Readers can become familiar with each feature and determine whether that feature is appropriate for their agency-specific implementation. If it is, then the agency procurement specification should require support for that feature and all of the mandatory requirements related to that feature.

For the last two categories of readers, this section is useful to gain a more thorough understanding as to why more detailed requirements (specified in later sections) exist.

2.1 Tutorial [informative]

A ConOps describes a proposed system from the users' perspective. Typically, a ConOps is used on a project to ensure that the system developers understand users' needs. Within NTCIP standards, it is used to document the intent of each feature that the NTCIP standard supports. It also serves as the starting point for users to select features that may be appropriate for their project.

The ConOps starts with a discussion of the current situation and issues that have led to the need to deploy systems covered by the scope of this document and to the development of this document itself. This discussion is presented in lay terms so that both potential users of the system and system developers can understand and appreciate the situation.

The ConOps then documents key aspects about the proposed system, including:

a) Reference physical architecture – The reference physical architecture defines the overall context of the proposed system and defines the specific interface addressed. The reference physical architecture may be supplemented with one or more examples that describe how the reference physical architecture may be realized in an actual deployment.

b) Architectural Needs – The architectural needs discuss issues and needs relative to the system architecture that have a direct impact on this document.
c) Features – The features identify and describe the various functions that users may want the field device to perform. These features are derived from the high level user needs identified in the problem statement but are refined and organized into a more manageable structure that forms the basis of the Protocol Requirements List (PRL) and Requirements Traceability Matrix (RTM) contained in Section 3.3.3 and Annex A, respectively.

Architectural needs and features are collectively called the user needs. In Section 3, user needs provide a basis to analyze the system and define the various functional requirements of a DMS. Each user need is required to trace to one or more functional requirements, and each functional requirement is required to be derived from at least one user need. This traceability is shown in the PRL as provided in Section 3.3.3.

This document is intended for use in a broad range of deployments and not every feature is required for every deployment. To accommodate this diversity, the PRL identifies each user need and requirement as being mandatory, optional, or conditional. The only items marked mandatory are those that relate to the most basic functionality of the device. To obtain a device that meets specific needs, the user identifies the optional and conditional needs appropriate for the specific project.

Each requirement is also presented in the RTM in Annex A, which defines how the requirement is fulfilled through the standardized dialogs and data element definitions provided in Section 4 and Section 5, respectively.

A conformant device may support other user needs, as long as they are conformant with the requirements of this document and its normative references (e.g., ISO 26048-1, NTCIP 8004). For example, a device may support data that has not been defined by this document; however, when exchanged via NTCIP 2301, the data shall be properly registered with a valid OBJECT IDENTIFIER under the Global ISO Naming Tree.

> Note: Off-the-shelf interoperability and interchangeability can only be obtained by using well documented user needs, along with their corresponding requirements and design, that are broadly supported by the industry as a whole. Designing a system that uses environments or features not defined in a standard or not typically deployed in combination with one another may inhibit the goals of interoperability and interchangeability, especially if the documentation of these user needs is not available for distribution to system integrators. This document allows implementations to support additional user needs and to support innovation, which is constantly needed within the industry; but users should be aware of the risks involved with using such environments or features.

The ConOps concludes by describing security needs and by providing a description of how this document relates to the National ITS Architecture.

2.2 Current Situation and Problem Statement [informative]

Transportation system managers use DMS in a variety of ways to improve transportation system operations. To perform their jobs, transportation system managers need to convey a variety of information to the traveling public. The information can be:

a) Advisory, such as information about transit vehicle arrival times, road closures, traffic congestion, estimated travel times, and weather warnings,
b) Regulatory, such as regulations about speed limits, mandatory detour information, or availability of high occupancy vehicle (HOV) lane access requirements.

Dynamic Message Signs (DMS) are one tool that can be used to convey this information to the traveling public. Based upon their perceived needs, transportation system managers decide what capabilities they may need within their DMS and where these DMS are deployed.

Depending on the needs and the budgets available, the DMS may be deployed in a network to provide coordinated operations or as stand-alone devices to provide information to travelers in areas where no integrated network capability exists.

DMS can be deployed in both stationary deployments, such as the roadside (overhead or side mounted) or on transit platforms, or portable on moveable vehicles. Additionally, different communications technologies such as dial-up (e.g., via land-lines for stationary signs or to portable signs placed on permanent pads where dial-up lines have been provided), serial (mostly provided to stationary signs), or Ethernet (e.g., via hardwire or even over wireless networks) are used to communicate with DMS.

2.3 Reference Physical Architecture [informative]

In order to ensure a consistent understanding of the terminology used by this document and the intended deployment environment, this document defines a reference architecture. The reference architecture is presented according to the rules defined within the Architecture Reference for Cooperative and Intelligent Transportation (ARC-IT) and defines the overall context of a deployment.

While the definition of this reference architecture follows ARC-IT conventions, the content is generally more detailed and uses terminology that is more consistent with the DMS industry. Section 2.8 provides a mapping between the reference architecture defined in this section and the elements contained in ARC-IT.

2.3.1 Physical View of the Reference Architecture

The physical view of the reference architecture is depicted in Figure 1 and includes:

a) Functional objects that identify high-level functions that are required within the system
b) Physical objects that identify where functions typically reside, and
c) Information flows that provide a very high-level indication of the information that needs to be exchanged between specific physical objects

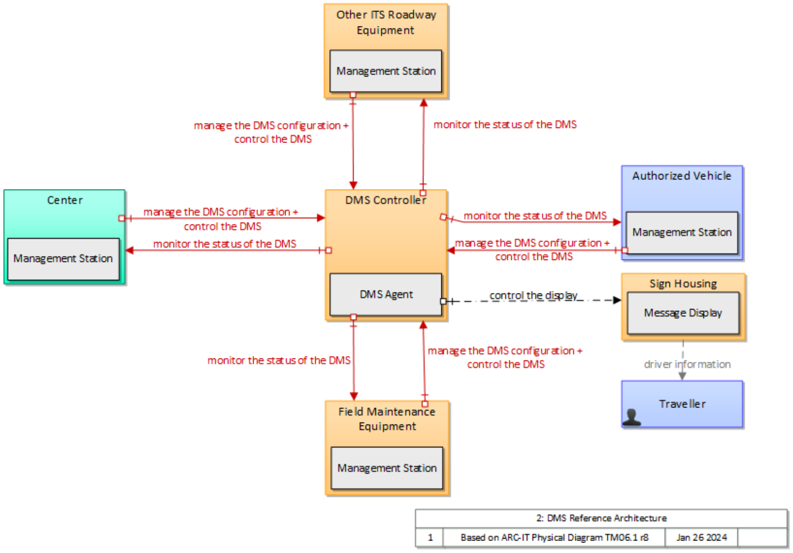
Other ITS Roadway Equipment
Management Station
manage the DMS configuration + control the DMS
monitor the status of the DMS
Authorized Vehicle
Center
DMS Controller
monitor the status of the DMS
Management Station
manage the DMS configuration + control the DMS
manage the DMS configuration + control the DMS
Management Station
monitor the status of the DMS
Sign Housing
DMS Agent
control the display
Message Display
driver information
monitor the status of the DMS
manage the DMS configuration + control the DMS
Traveller
Field Maintenance Equipment
Management Station
2: DMS Reference Architecture
1
Based on ARC-IT Physical Diagram TM06.1 r8
Jan 26 2024

Figure 1 Reference Architecture for a DMS

This document is only concerned with the interface between a management station and the DMS controller. To fulfill all of the end-user services, additional requirements may be necessary for the external computer.

2.3.1.1 Functional Objects

2.3.1.1.1 Management Station

A Management Station is a functional object that is responsible for allowing the end user (e.g., field technician, traffic engineer) to configure and manage one or more field devices, including the DMS. The Management Station can be located in a center (e.g., traffic management center, maintenance and construction management center), ITS roadway equipment (e.g., traffic signal controller), field maintenance equipment (e.g., technician's laptop), or an authorized vehicle (e.g., maintenance and construction vehicle).

The primary functions of the NTCIP Manager are to allow an end user to perform the following:

a) manage the DMS configuration,
b) control the DMS, and
c) monitor the status of the DMS.

Several management stations may communicate with the same DMS. The user should ensure that only one management station at a time controls the DMS (e.g., which message is currently displayed). Further, all control operations should be carefully coordinated with the management of the configuration to ensure the intended operation. All management stations may monitor the status simultaneously.

2.3.1.1.2 DMS Agent

The DMS Agent is a functional object that is responsible for controlling the display on the sign housing. This includes monitoring the health of the sign display and sign housing. It is also responsible for responding to all requests received from management stations, storing the current configuration and operating according to the configuration parameters and commands received. The DMS Agent is often deployed in a device dedicated to DMS functionality, but it can also be deployed as a module in an advanced transportation controller.

The primary functions of the DMS Agent are to perform the following:

a) convert NTCIP data into the appropriate commands to control the sign display,
b) manage the sign display to ensure that it operates according to the configuration and commands,
c) manage the resulting NTCIP data (e.g., maintaining logs per configuration), and
d) providing NTCIP data in response to authorized requests.

2.3.1.2 Physical Objects

DMS deployments typically include a traffic management center (TMC), a DMS controller, and a sign housing. The deployments are in an environment where the sign display can display its message to drivers. Deployments may also include other centers, other ITS roadway equipment (e.g., traffic signal controllers, weather stations, parking equipment), field maintenance equipment, and/or authorized vehicles. The following paragraphs describe the primary functions of the physical objects.

2.3.1.2.1 Center

A center is an entity that provides application, management, administrative, and support functions from a fixed location not in proximity to the road network.

2.3.1.2.2 Dynamic Message Sign Controller

A DMS controller is the controller that controls the sign display and communicates with the management station.

The DMS controller, its cabinet, and sign housing are jointly called the "DMS".

2.3.1.2.3 ITS Roadway Equipment

Within the scope of this document, “ITS Roadway Equipment” is another installation of ITS equipment along the roadway that interacts with the DMS controller. The ITS Roadway Equipment could be another DMS, but it is typically another device that has a need to display information. For example, a field-based application might monitor pavement sensors and command the DMS to display an ice warning when ice is detected.

2.3.1.2.4 Field Maintenance Equipment

“Field Maintenance Equipment” is the portable equipment used by field personnel to locally troubleshoot, initialize, reprogram, and test infrastructure equipment.

2.3.1.2.5 Authorized Vehicles

An “Authorized Vehicle” is any specialized service vehicle or equipment that has been explicitly granted access permissions to the DMS controller.

2.3.1.2.6 Sign Housing

The “Sign Housing” is the physical unit that includes the sign display that is designed to be read by travelers. The connection between the DMS controller and the sign housing is not addressed by this document.

2.3.1.2.7 Traveler

The “Traveler” represents any person who is part of the target audience of the message. It is expected that in most NTCIP deployments, the traveller will be a driver of a vehicle and thus, the traveler is shown in blue; however, it is recognized that deployments can target others, including those not in a vehicle.

2.3.1.3 Information Flows

The DMS reference architecture includes three information flows that are addressed by this document and two additional information flows that provide a broader context to show how the information reaches the traveller.

2.3.1.3.1 manage the DMS configuration

Information that manages the internal operation of the DMS controller and decisions that it makes. For example, this includes the messages available for display, schedules for messages, the fonts to be used when displaying a message, etc.

2.3.1.3.2 control the DMS

Commands used to directly control the DMS. This includes commands to display a specific message, control of auxiliary equipment (e.g., fans), reset commands, etc.

2.3.1.3.3 monitor the status of the DMS

Information that can be used to monitor the status and health of a DMS. This includes the current message displayed, diagnostic information, and other management parameters.

2.3.1.3.4 control the display

Electrical signals provided by to the sign display to be converted into a message on the sign display such that it is visible to travelers.

2.3.1.3.5 driver information

Regulatory, warning, guidance, and other information provided to the driver to support safe and efficient vehicle operation.

2.3.2 DMS Characteristics

A factor that complicated the development of this document and that complicates the work of a specification developer is the fact that there are a wide variety of DMS available in the marketplace. To promote interoperability among the different signs, this document provides a single protocol that is compatible with all of these varied DMS. However, the varied nature of these signs dictates that many of the features defined within this document are not applicable to all DMS. Thus, the user must categorize the DMS according to several key characteristics prior to determining which requirements are mandatory for a particular project and/or type of DMS. These characteristics include DMS Type, DMS Technology, and DMS Display Matrix Configuration.

> Note: A specification can allow for any of several types, technologies, or display matrix configurations by leaving the selection of these items as optional while noting that the support of the option is left to the manufacturer but that the manufacturer must choose at least one. For example, a specification could allow for either a line matrix or a full matrix sign by:
>
> a) selecting 'Yes' on line 2.3.2.3.2,
> b) selecting 'Yes' on lines 2.3.2.3.2.1 and 2.3.2.3.2.2 blank and also entering the required configurations, and
> c) selecting 'No' on line 2.3.2.3.2.3 in the PRL of Section 3.3.3.

2.3.2.1 DMS Types

There are many types of DMS and they can be characterized in many ways. One way is by the capabilities the DMS offers for handling messages. This characterization places a DMS into one of three major categories:

a) Blank-out Sign (BOS) – this type of DMS can only show one fixed message or nothing.
b) Changeable Message Sign (CMS) – this type of DMS can display one of two or more predefined messages, or be blank.
c) Variable Message Sign (VMS) – this type of DMS is one in which the message to be displayed can be created after the sign is installed in the field. It can also have predefined messages in its library of stored messages. By policy and/or system design, the management station may restrict the rights of selected operators to ensure that only authorized personnel can modify or create messages “on-the-fly”.

2.3.2.2 DMS Technologies

DMS can also be characterized by the technology that is used in the sign. The technologies used can include any combination of the following technologies:

a) Fiber Optic
b) Light emitting diode (LED)
c) Flip disk or shutter
d) Lamp matrix
e) Drum (rotating, multifaceted cylinder)

2.3.2.3 DMS Display Matrix Configuration

Finally, DMS can be characterized by the type of display layout employed by the sign, as follows:

a) No matrix (i.e., it is not a pixel matrix sign)
b) Matrix sign
    i. Full matrix
    ii. Line matrix
    iii. Character matrix

> Note: Typically, matrix signs are VMS and VMS are matrix signs, but this is not always true; for example, the term VMS would also include: 7-segment displays, electronic ink displays, etc.

2.3.2.4 DMS Display Support of Physical Beacons

The display of the DMS can also be supported or enhanced by the addition of physical beacons which are blinking lights attached to the DMS display. They may be activated as part of particular messages.

2.4 Architectural Needs

All the architectural needs are defined within the user needs of ISO 26048-1. Commonly used architectural needs within the scope of a DMS are listed in the PRL with references to the relevant ISO 26048-1 clauses and include the following:

a) **Authenticate users:** allows users to ensure that all communications with the device are with authenticated users.
b) **Control access to data:** allows administrators to configure the detailed access privileges that are granted to each authenticated user.
c) **Monitor failed access to the field device:** allows administrators to discover when unauthorized attempts are made to access the device.
d) **Manage the field device:** allows a user to remotely reset the controller and configure and monitor basic information about the field device, such as its:
    a. Identity,
    b. Location,
    c. Power source,
    d. Default language,
    e. Capabilities (e.g., which objects are supported, maximum message size),
    f. Status, and
    g. Configuration identifier.
e) **Monitor the field device:** allows a user to monitor information about the field device and its environment. This includes the ability to monitor:
    a. ambient air temperature (e.g., to display on the sign face),
    b. ambient relative humidity,
    c. ambient light,
    d. doors,
    e. enclosure air temperature,
    f. processor temperature,
    g. enclosure relative humidity,
    h. power supplies,
    i. mains power,
    j. battery power,
    k. generator power,
    l. solar power, and
    m. wind power.
f) **Monitor and control the field device peripherals:** allows a user to monitor and control support equipment, including:
    a. fans,
    b. heaters,
    c. dehumidifiers,
    d. air conditioners,
    e. thermostats,
    f. auxiliary bi-directional ports,
    g. auxiliary inputs, and
    h. auxiliary outputs.
g) **Receive notifications of triggers firing:** allows a user to define trigger conditions (e.g., an object value exceeding a defined value) and to automatically receive an SNMP notification when the trigger condition occurs.
h) **Manage device-specific notifications:** allows the user to configure all supported notifications to use notification channels, which prevent notifications from flooding the communications channel.
i) **Log system events:** allows the device to record information about internal events that can be of interest during forensic analysis, e.g., for debugging purposes.
j) **Log user-defined snapshots:** allows a user to define trigger conditions (e.g., an object value exceeding a defined value) and to automatically log information when the trigger condition occurs.

ISO 26048-1 also defines the following user needs that are not envisioned to be needed for most DMS deployments. These user needs are not included in the PRL to avoid their inadvertent selection, but they are not prohibited:

a) **Record a series of snapshots:** allows a user to define trigger conditions (e.g., an object value exceeding a defined value) and to automatically log a series of snapshots based on the trigger condition occurring. For example, when the cabinet door opens, record the cabinet temperature once a second for 60 seconds.
b) **Issue trigger-based commands:** allows a user to define trigger conditions (e.g., an object value exceeding a defined value) and to automatically send another device an SNMP set request when the trigger condition occurs. For example, when a gate closed message is displayed, command the attached gate to close (however, this example is discouraged as a gate closure should include verification mechanisms not offered by this feature to ensure safe operations).
c) **Efficient exchange of data (dynamic objects):** allows a management station to define dynamic objects that can be used for frequent requests to reduce communications overhead. For example, rather than monitoring 15 discreet status objects, the 15 objects can be placed into a dynamic object and retrieved as a single entity. This can be useful when a management station frequently exchanges a large number of integer values with the device.

The "Configure a complex device" user need is prohibited for use with message signs. It is designed for devices that have many inter-related parameters that must be set simultaneously for safety reasons and does not apply to DMS.

2.5 Features

This section identifies and describes the various standard features that may be offered by a DMS. Section 3 uses these features in the analysis of the system to define the various functional requirements of a DMS. A conformant DMS may offer additional features, as long as they are conformant with this document’s requirements.

The operation of a DMS can be categorized into three major areas:

a) Manage the DMS configuration
b) Control the DMS
c) Monitor the status of the DMS

2.5.1 Manage the DMS Configuration

The various sub-features for managing the DMS configuration include:

a) Determine DMS Capabilities
b) Determine Sign Display Capabilities
c) Manage Fonts
d) Manage Graphics
e) Manage Automatic Brightness
f) Configure Speed Limit
g) Manage the Display Location
h) Manage Physical Beacons

The subsequent sections detail these sub-features.

2.5.1.1 Determine DMS Capabilities

This feature allows the operator to determine basic information about the DMS, such as the type and technology of the DMS.
Note—More general information about the DMS is provided by the "Manage the field device" user need, as defined in ISO 26048-1 Clause 6.4.

2.5.1.2 Determine Sign Display Capabilities

This feature allows the operator to retrieve the necessary information to produce a rendering of a suggested or active message. This feature also allows the system to ensure that a message can be displayed on the DMS. The feature allows the operator to determine the detailed physical limitations of the DMS as well as details regarding the current fonts and any graphics that are stored.

2.5.1.3 Manage Fonts

This feature allows the operator to define and edit the appearance of the fonts used to display messages on the sign face. This helps an operator ensure that messages have a consistent appearance across many DMS in a large system despite the use of different manufacturers, etc. It allows the operator to manage the height and width of the font, and the color of the font. It allows the operator to edit or delete existing fonts, and to create new fonts in a controller. It also allows an operator to determine the existing configuration of fonts.

Each font supported by the DMS should support a common set of characters (e.g., ASCII codes) to improve interoperability, including letters numbers and various special characters that are frequently used on DMS.

2.5.1.4 Manage Graphics

This feature allows the operator to manage the graphics stored within a DMS. It allows the operator to define a graphic for later use, manage existing graphics, and determine the graphic storage capabilities of the DMS.

2.5.1.5 Manage Automatic Brightness

This feature allows the operator to configure when the sign may automatically switch from one brightness level to the next. This allows the operator to configure how the sign automatically responds to changing lighting conditions to compensate for sun shining in the traveler's vision or 'wash-out' conditions, such as during early morning and pre-sunset conditions.

2.5.1.6 Configure Speed Limit

This feature allows the operator to configure the speed limit applicable to the location of the DMS.

2.5.1.7 Reserved

This feature was formerly used for the "Configure Low Fuel Threshold" feature. This capability is now addressed by the ISO 26048-1, 8.8.14.2.4 (Field device generator fuel level monitored through SRSA) requirement of the ISO 26048-1, 6.5.4.4 (Monitor field device generator power) user need.

2.5.1.8 Manage the Display Location

This feature allows the operator to configure and determine the location (with accuracy) of the sign dispaly and the direction that it faces so that system operators can better understand the location and impact of the sign.

NOTE—An implementation can support the SNMP-read operation of these parameters without supporting the corresponding SNMP-write operation (e.g., an implementation can automatically base these values on data provided by a connected GNSS receiver).

NOTE—This data can be identical or mathematically determined based on the configured controller location if the DMS is designed for a fixed configuration of the sign dispaly and controller (e.g., a portable DMS might be configured such that the sign display position of the sign display is always immediately above the controller.)

NOTE—In versions prior to v04, this heading number was used for "Configure Low Fuel Threshold". That capability is now included within ISO 26048-1 Clause 6.5.8 "Monitor field device generator power".

2.5.1.9 Manage Physical Beacons

This feature allows the operator to configure flash timing sequences and colors for physical beacons.

NOTE—Virtual beacons (i.e., flashing pixels on the sign display) can be defined as a part of a message using graphics or characters. The flash sequences and colors for virtual beacons are defined within the MULTI string and are not affected by this feature.

2.5.2 Control the DMS

The various sub-features for controlling the DMS include:

a) Control a DMS from More than One Location
b) Control the Sign Face
c) Control the Brightness Outputs
d) Perform Preventative Maintenance

Subsequent sections detail these sub-features.

2.5.2.1 Control a DMS from More than One Location

This feature addresses the need for DMS to be controlled both remotely (e.g., from one or more central computers) and locally (e.g., from the controller directly or from a laptop computer connected to the controller). Whether the DMS is controlled remotely or locally, the features and capabilities are the same. This feature also addresses the need to prevent different sources of control from interfering with one another by attempting to control the DMS simultaneously.

2.5.2.2 Reserved

NOTE—This heading was formerly used for the "Remotely Reset the Sign Controller" user need. This capability is now addressed by the 8.5.2.5 (Control remote reset of the controller) requirement of 6.4 (Manage the field device).

2.5.2.3 Control the Sign Face

This feature addresses the need to place information on or remove information from the sign face to convey proper information to travelers. The feature includes the following sub-features:

a) Activate and Display a Message
b) Prioritize Messages
c) Define a Message
d) Blank the Sign
e) Schedule Messages for Display
f) Change Message Display based on an Internal Event
g) Change Message Display based on a Conditional Trigger

2.5.2.3.1 Activate and Display a Message

This feature allows an operator to activate a previously defined message to be displayed on the sign face. The message can be a blank message or come from a set of previously defined messages.

When activating the message the operator will need to specify the desired duration for the display and the relative priority for the proposed message to override the currently displayed message.

> Note: Activating a message that is stored in a central system library can be achieved by first downloading the message to the sign controller and then activating the message per this section. A user can also display a message defined "on-the-fly" by the same process.

2.5.2.3.2 Prioritize Messages

This feature allows an operator to prioritize particular messages. For example, a priority scheme will allow an operator to maintain an accident-related message, even if the same operator previously scheduled the display of a non-accident related message.

2.5.2.3.3 Define a Message

This feature enables the operator to create a message and to modify its format and content. The feature includes:

a) Uniquely identifying a message
b) Ensuring that a message is intact
c) Defining the exact contents of the message to be displayed on the sign face.
d) If supported, activating physical beacons when a message is displayed, to attract the traveler's attention

2.5.2.3.4 Blank a Sign

This feature enables the operator (or logic within the management station) to remove any messages displayed on a sign (causing the sign to appear blank).

2.5.2.3.5 Schedule Messages for Display

This feature enables the operator to configure the DMS to activate messages at a future time ("scheduling"). The operator can indicate a series of times at which an associated message will be activated either independently or as a part of a day plan schedule. The associated message can be any message stored in the sign controller, including a blank message.

2.5.2.3.6 Change Message Display Based on an Internal Event

This feature allows the operator to indicate which message should be displayed when certain pre-defined, non-scheduled events occur.

2.5.2.3.6.1 Change Message Display Based on Power Loss

This feature allows the operator to indicate which message should be displayed when the sign display loses power.

NOTE—This feature only applies to sign display technologies that can continue to display a message when no power is available (e.g., flip disk signs). Such signs can be equipped with capacitors or other technology to activate a message as power is being lost.

2.5.2.3.6.2 Change Message Display Based on Short Power Loss Recovery

This feature allows the operator to indicate which message should be displayed when power is restored to the sign display after a power loss. If a long power loss recovery message is also supported, this message is only displayed if the power loss is shorter than a configured value.

2.5.2.3.6.3 Change Message Display Based on Long Power Loss Recovery

This feature allows the operator to indicate which message should be displayed when power is restored to the sign display after a power loss exceeding a specified duration.

2.5.2.3.6.4 Change Message Display Based on Controller Reset

This feature allows the operator to indicate which message should be displayed after the sign controller is reset.

2.5.2.3.6.5 Change Message Display Based on Communication Loss

This feature allows the operator to indicate which message should be displayed when the sign controller has not received any communictions from a management station for a duration exceeding a configured value.

2.5.2.3.6.6 Change Message Display Based on End Message Duration

This feature allows the operator to indicate which message should be displayed when the previously running message reaches its end duration time.

2.5.2.3.7 Change Message Display Based on a Conditional Trigger

This feature allows the operator to indicate which message should be displayed when a user-defined event occurs, such as when an object exceeds or drops below a user-specified value.

2.5.2.4 Reserved

NOTE—This heading was formerly used for the "Control External Devices" user need. This capability is now addressed by the 6.5.5 (Manage field device auxiliary bi-directional ports), 6.5.6 (Manage field device auxiliary inputs), and 6.5.7 (Manage field device auxiliary outputs) user needs defined in ISO 26048-1.

2.5.2.5 Control the Brightness Output

This feature enables the operator to control the sign brightness either directly or through an automated algorithm, depending on the capabilities of the DMS. At a minimum, the operator must be able to control brightness of the sign display manually for light emitting signs. In addition, the operator should be able to control the brightness level through the use of light sensors (photocells) on the DMS, if available, that can detect the ambient light levels and adjust brightness levels in an appropriate fashion. This brightness control is needed to compensate for the external environment's effect on the visibility of the message, such as when the sun is shining in the eyes of travelers.

2.5.2.6 Perform Preventative Maintenance

This feature enables the operator to enable or disable the periodic exercise of pixels (activated either manually or via a schedule) to ensure that they are performing reliably.

2.5.3 Monitor the Status of the DMS

The various sub-features for monitoring the status of the DMS include:

a) Perform Diagnostics
b) Monitor the Current Message

Subsequent sections detail these sub-features.

2.5.3.1 Perform Diagnostics

This feature enables the operator to test the operational status of system components. It consists of the following sub-features:

a) Monitor Message Errors
b) Monitor the Sign Control Source
c) Monitor Attached Speed Detectors
d) Monitor Automatic Blanking of Sign
e) Monitor General DMS Status
f) Monitor Photocells
g) Monitor Lamp Status
h) Monitor Drum Status
i) Monitor Sign Shutdown Due to Temperature

2.5.3.1.1 Reserved

NOTE—This heading was formerly used for the "Determine Sign Error Conditions (High-Level Diagnostics)" user need. This capability is now incorporated into each diagnostic function.

2.5.3.1.2 Reserved

NOTE—This heading was formerly used for the "Monitor Sign Subsystem Failures (Mid-Level Diagnostics)" user need. This capability is now incorporated into each diagnostic function.

2.5.3.1.3 Reserved

NOTE—This heading was formerly used for the "Monitor Subsystem Failure Details (Low-Level Diagnostics)" user need. This capability is now incorporated into each diagnostic function.

2.5.3.1.4 Monitor Message Errors

This feature enables the operator to monitor the errors associated with defining or activating a particular message.

2.5.3.1.5 Reserved

NOTE—This heading was formerly used for the "Monitor Sign Environment" user need. This capability is now addressed by the ISO 26048-1, 6.5.2.2 (Monitor field device enclosure tempearture) and ISO 26048-1, 6.5.2.4 (Monitor field device enclosure humidity) user needs.

2.5.3.1.6 Monitor the Sign Control Source

This feature enables the operator to determine the physical location from or mechanism through which the DMS is being controlled. Possible control sources include:

a) Central computer
b) DMS time-based scheduler
c) An individual physically present at the DMS site
d) Internal logic (e.g., triggered message)

2.5.3.1.7 Monitor Attached Speed Detectors

This feature enables the operator to determine the current reading of any speed detectors attached to the DMS.

2.5.3.1.8 Reserved

This heading was formerly used for the "Monitor Door Status" user need. This capability is now addressed by the ISO 26048-1, 6.5.2.1 (Monitor field device doors) user need.

2.5.3.1.9 Reserved

NOTE—This heading was formerly used for the "Monitor Controller Software Operations " user need. This capability is now addressed by the ISO 26048-1, 8.5.2.11 (Monitor controller operational status) and 8.5.2.13 (Monitor watchdog failure count) requirements of the 6.4 (Manage the field device) user need.

2.5.3.1.10 Monitor Automatic Blanking of Sign

This feature enables the operator to monitor the automatic display of a blank message when diagnostics detect that too many pixels are non-operational, that the light outputs are faulty, or that the critical temperature has been reached. This prevents illegible messages from being displayed.

2.5.3.1.11 Reserved

NOTE—This heading was formerly used for the "Monitor Power Source" user need. This capability is now addressed by the ISO 26048-1, 8.8.2.1.9 (Monitor current power source) requirement of the 6.4 (Manage the field device) user need.

2.5.3.1.12 Reserved

NOTE—This heading was formerly used for the "Monitor Power Voltage" user need. This capability is now addressed by the 8.8.17.2.1 (Field device mains power voltage monitored through SRSA) requirement of 6.5.4.2 (Monitor field device mains power) and 8.8.10.2.1 (Field device battery voltage monitored through SRSA) requirement of ISO 26048-1, 6.5.4.3 (Monitor field device battery power).

2.5.3.1.13 Reserved

NOTE—This heading was formerly used for the "Monitor Fuel Level" user need. This capability is now addressed by the 8.8.14.2.4 (Field device generator fuel level monitored through SRSA) requirement of 6.5.4.4 (Monitor field device generator power). This feature is typically used in portable signs.

2.5.3.1.14 Reserved

NOTE—This heading was formerly used for the "Monitor Engine RPM" user need. This capability is now addressed by the 8.8.14.2.3 (Field device generator engine speed monitored through SRSA) requirement of 6.5.4.4 (Monitor field device generator power). This feature is typically used in portable signs.

2.5.3.1.15 Monitor General DMS Status

This feature enables the operator to monitor the DMS to determine which, if any, of the following conditions are active:

a) communications error
b) power error
c) attached device error
d) lamp error
e) pixel error
f) photocell error
g) message error
h) controller error
i) temperature warning
j) climate control system error
k) critical temperature error
l) drum sign rotor error
m) door open
n) humidity warning
o) manufacturer-specific error

2.5.3.1.16 Monitor Pixel Errors

This feature enables the operator to detect failed pixels within a matrix display.

2.5.3.1.17 Monitor Photocells

This feature enables the operator to monitor the status of individual photocells for devices that support automatic brightness.

2.5.3.1.18 Monitor Lamp Status

This feature enables the operator to monitor the status of lamps for either a lamp or fiber optic sign display.

2.5.3.1.19 Monitor Drum Status

This feature enables the operator to monitor the status of drums for a drum display.

2.5.3.1.20 Monitor Sign Shutdown Due to Temperature

This feature enables the operator to monitor sign shutdowns due to exceedingly high or low temperature.

2.5.3.2 Monitor the Current Message

This feature enables the operator to determine what message is currently displayed on the sign face.

2.5.4 Multi-Version Interoperability (MVI-Backward Compatibility)

All prior versions of this document failed to provide adequate security; therefore, this document does not identify any user needs to support backward compatibility.

2.6 Security

A management station needs to have confidence that communications with its devices are always secure and trustworthy. This requires confidentiality, integrity, and availability.

2.6.1 Addressing Vulnerabilities

This document standardizes the objects that can be used by a remote entity to monitor and/or control a device. Allowing remote operations without providing sufficient information security can have a negative effect on the communications network and/or the transportation system. In particular, the mis-configuration of "read-write" and/or "read-create" objects can result in the activation of unauthorized features or unauthorized changes to the device configuration; these changes can result in potentially hazardous situations.

Further, even failing to secure exchanges of "read-only" objects can have harmful effects. Providing unauthorized access to read-only objects can reveal information about the device that can be useful in conducting a cyber-attack against the device itself, other devices, and potentially the central system.

Likewise, failing to authenticate the source of the read-only data allows a cyber threat actor to potentially impersonate the device and provide false information to a traffic management center.

The object types defined by this document are intended to be transmitted over SNMPv3. SNMP versions prior to SNMPv3 do not include adequate security. Prior SNMP versions do not provide any control as to who on the network is allowed to access and read/change/create/delete objects, even if the network itself is secure.

2.6.2 Conformant Security Environment

The conformant security environment is highly recommended as it does not allow the device to support any unsecured protocol.

2.6.3 Consistent Security Environment

It is recognized that during a transition to a secure environment, agencies might need to procure equipment that conforms to their current unsecured environment while still wanting something that can be reconfigured to support their future secure environment. This is allowed as long as the device is able to disable the unsecured protocols. Nonetheless, the fact that consistent devices support unsecured protocols mean that the unsecured protocols can be enabled and the device is inherently less secure than a conformant device.

2.7 Operational Policies and Constraints

To provide adequate information security, deployments should:

a) Disable any SNMP versions prior to SNMPv3 as soon as possible,
b) Use the communications stack defined in NTCIP 2301,
c) Ensure that implementations are properly configured to only give read and/or, write access to objects based on the legitimate needs of each principal (i.e., user), and
d) Update security keys at intervals that inhibit brute-force attacks from being successful.

2.8 Relationship to the National ITS Architecture [Informative]

This document addresses 21 information flows (with a total of 33 information transfers or triples) included in the Architecture Reference for Cooperative and Intelligent Transportation (ARC-IT). These information transfers coorespond to the information flows in the DMS Referenence Architecture as indicated in Table 1.

Table 1 Relationship between DMS RA and ARC-IT

DMS RA Information Flow(s)	Source	ARC-IT Information Flow	Destination
Control and Monitor	IRE	dynamic sign coordination	Other IRE
Configure and Control	EMC	incident scene warning device control	IRE
Monitor	IRE	incident scene warning status	EMC
Configure and Control	TMC	lane management control	IRE
Control and Monitor	IRE	lane management coordination	TMC
Monitor	IRE	lane management information	TMC
Configure and Control	CV RSE	roadway dynamic signage data	IRE
Configure and Control	M&CMC	roadway dynamic signage data	IRE
Configure and Control	M&CV OBE	roadway dynamic signage data	IRE
Configure and Control	TMC	roadway dynamic signage data	IRE
Monitor	IRE	roadway dynamic signage status	CV RSE
Monitor	IRE	roadway dynamic signage status	M&CMC
Monitor	IRE	roadway dynamic signage status	M&CV OBE
Monitor	IRE	roadway dynamic signage status	TMC
Control and Monitor	IRE	roadway warning coordination	Other IRE

Configure and Control	TMC	roadway warning system control	IRE
Monitor	IRE	roadway warning system status	TMC
Configure and Control	EC	speed monitoring control	IRE
Configure and Control	M&CMC	speed monitoring control	IRE
Configure and Control	TMC	speed monitoring control	IRE
Monitor	IRE	speed monitoring information	EC
Monitor	IRE	speed monitoring information	M&CMC
Monitor	IRE	speed monitoring information	TMC
Configure and Control	TMC	stop sign gap assist control	IRE
Monitor	IRE	stop sign gap assist status	TMC
Configure and Control	TMC	variable speed limit control	IRE
Control and Monitor	IRE	variable speed limit coordination	IRE
Monitor	IRE	variable speed limit status	TMC
Configure and Control	BIS	vehicle signage local data	IRE
Configure and Control	Other IRE	vehicle signage local data	IRE
Configure and Control	PAE	vehicle signage local data	IRE
Configure and Control	M&CMC	work zone warning device control	IRE
Monitor	IRE	work zone warning status	M&CMC

Section 3
Functional Requirements [Normative]

This section defines the Functional Requirements based on the user needs identified in the ConOps, as defined in Section 2. This section is intended for all readers of the document, including:

a) Transportation operations managers
b) Transportation operations personnel
c) Transportation engineers
d) System integrators
e) Device manufacturers

For the first three categories of readers, this section is useful in understanding the detailed requirements that this docuemnt places on a DMS. For these readers, Sections 3.3.3 and 3.3.4 are particularly useful in preparing procurement specifications and in guiding the reader to the detailed text for both the user needs and requirements for any specific feature.

For the last two categories of readers, this section is likely to be useful in fully understanding what is required of equipment conformant with this document. The tables in Sections 3.3.3 and 3.3.4 are also useful to document the capabilities of implementations.

3.1 Tutorial [informative]

This section defines the formal requirements that are intended to fulfill the user needs identified in Section 2. This is achieved through a PRL that traces each user need to one or more requirements. The details of each requirement defined in this document are presented following the PRL. Functional requirements are presented in three broad categories as follows:

a) Architectural Requirements – These requirements define the required behavior of the system in exchanging data across the communications interface, including any restrictions to general architectural requirements, based upon the architectural needs identified in Section 2.
b) Data Exchange Requirements – These requirements define the required behavior of the system in exchanging data across the communications interface based upon the features identified in Section 2.
c) Supplemental Requirements – These requirements define additional requirements of the system that are derived from the architectural and/or data exchange requirements, but are not themselves architectural or data exchange requirements. A given supplemental requirement may relate to multiple architectural and/or data exchange requirements. Supplemental requirements frequently include range capabilities of the equipment (e.g., how many messages a VMS is required to support or what the message shall be on a blank-out sign).

3.2 Scope of the Interface [Informative]

<In the opinion of the responsible NTCIP working group, this section does not apply in the context of this document.>

3.3 Protocol Requirements List (PRL)

The PRL, provided in tables defined under Sections 3.3.3, and 3.3.4, map the user needs defined in Section 2 to the requirements defined in Section 3. The PRL can be used:

a) To indicate which requirements are to be implemented in a project-specific implementation.
b) As a checklist to reduce the risk of failure to conform to this document through oversight.
c) To provide a detailed indication of the capabilities of the implementation.
d) As a basis for initially checking the potential interoperability with another implementation.

3.3.1 Notation [Informative]

The following notations and symbols are used to indicate status and conditional status in the PRL within all NTCIP standards. Not all of these notations and symbols may be used within this document.

3.3.1.1 Conformance Symbols

The symbols in Table 2are used to indicate status.

Table 2 Conformance Symbols

Symbol	Status
M	Mandatory
M.#	Support of every item of the group labeled by the same numeral # is required, but only one is active at a time
O	Optional
O.# (range)	Part of an option group. Support of the number of items indicated by the '(range)' is required from all options labeled with the same numeral #
C	Conditional
N/A	Not-applicable (i.e. logically impossible in the scope of NTCIP 1203 v03)
X	Excluded or prohibited

The O.# (range) notation is used to show a set of selectable options (e.g., O.2 (1..*) would indicate that one or more of the option group 2 options must be implemented). Two-character combinations are used for dynamic requirements. In this case, the first character refers to the static (implementation) status, and the second refers to the dynamic (use); thus "MO" means "mandatory to be implemented, optional to be used."

3.3.1.2 Conditional Status Notation

The predicate notations in Table 3 may be used.

Table 3 Predicate Notations

Predicate	Notation
<predicate>:	This notation introduces a single item that is conditional on the <predicate>.
<predicate>::	This notation introduces a table or a group of tables, all of which are conditional on the <predicate>.
(predicate)	This notation introduces the first occurrence of the predicate. The feature associated with this notation is the base feature for all options that have this predicate in their conformance column.

The <predicate>: notation means that the status following it applies only when the PRL states that the feature or features identified by the predicate are supported. In the simplest case, <predicate> is the identifying tag of a single PRL item. The <predicate>:: notation may precede a table or group of tables in a section or subsection. When the group predicate is true then the associated section shall be completed. The symbol <predicate> also may be a Boolean expression composed of several indices. "AND", "OR", and "NOT" shall be used to indicate the Boolean logical operations.

The predicates used in this document map to the sections in Table 4.

Table 4 Predicate to NTCIP 1203 v03 Section Mapping

Predicate	Section
BOS	2.3.2.1.a
CMS	2.3.2.1.b
Door	2.5.3.1.8
Drum	2.3.2.2.e
Environment	2.5.3.1.5
Fiber	2.3.2.2.a
Flip/Shutter	2.3.2.2.c
Fonts	2.5.1.3
Graphics	2.5.1.4
Lamp	2.3.2.2.d
LED	2.3.2.2.b
Matrix	2.3.2.3.b
VMS	2.3.2.1.c
AutoBright	2.5.1.5
ClimateTest	3.5.3.1.1.3
DoM	3.6.6.2.13.7
DoW	3.6.6.2.13.6
Fields	3.6.6.2.13
Flash	3.6.6.2.10
LampTest	3.5.3.1.1.1
Month	3.6.6.2.13.8
PixelTest	3.5.3.1.1.2
Speed	3.5.3.1.9
Temp	3.6.6.2.13.4
Time	3.6.6.2.13.1 or 3.6.6.2.13.2 or 3.6.6.2.13.3
Year	3.6.6.2.13.9

3.3.1.3 Support Column Symbols

The support column can be used by a procurement specification to identify the required features for the given procurement specification or by an implementer to identify which features have been implemented. In either case, the user circles the appropriate answer (Yes, No, or N/A) in the support column.

Table 5 Support Requirement Column Entries

Entry	Identifier
Yes	Supported by the implementation.
No	Not supported by the implementation.
N/A	Not applicable

3.3.1.4 Data Interface versus Associated Hardware

This document defines the data interface that is to be supported by a DMS; requirements contained in this specification should not be interpreted to require any associated hardware. Hardware requirements should be defined in a separate specification.

EXAMPLE—A specification using the PRL contained within this document can select the "Manage field device fans" option. However, this option only requires that the controller supports a data interface for the fans supported by the device. If the hardare specification does not require any fans, the provided data interface can have minimal data support as there might not be any fans to manage.

3.3.2 Instructions for Completing the PRL

In the 'Support' column, each response shall be selected either from the indicated set of responses (for example: Yes / No / NA), or it shall reference additional items that are to be attached (for example, list of Permanent DMS Messages to be supported by an implementation).

If a conditional requirement is inapplicable, use the Not Applicable (NA) choice. If a mandatory requirement is not satisfied, exception information shall be supplied by entering a reference Xi, where i is a unique identifier, to an accompanying rationale for the non-conformance. When the status is expressed as a two-character combination (as defined in 3.3.3.1 above), the response shall address each element of the requirement; e.g., for the requirement "mo," the possible compliant responses are "yy" or "yn."

The reason that the PRL provides two tables is because the supplemental requirements may relate to multiple architectural and/or data exchange requirements (contained in the first table). This split reduces the amount of repetition that would otherwise increase the size of the first table.

> Note: A user might fill out the first table first before proceeding to the second table. However, it will likely be easier to complete the corresponding rows in the second table when considering specific items in the first table.
>
> Note: A specification can allow for flexibility in a deliverable by leaving the selection in the Support column blank for a given row. For example, a specification could require the supporting of graphics by selecting 'Yes' on Row 2.5.1.4, and leaving rows 3.5.1.4.1 thru 3.5.1.4.7 as well as 3.6.11 blank (if no additional project requirements needed to be stated).

3.3.2.1 Conformance Definition

To claim "Conformance" to this docuemnt, the vendor must minimally satisfy the mandatory requirements as identified in the PRL tables (3.3.3 and 3.3.4).

> Note: The reader and user of this doucment is advised that 'conformance' to this document should not be confused with 'compliance' to a specification. This document is as broad as possible to allow a very simple device such as a blank-out sign to be 'conformant'. A procurement specification needs to identify the requirements of a particular project and needs to require the support of those requirements. A specification writer is advised to match the requirements of a project with the corresponding standardized requirements defined in NTCIP 1203 v03 to achieve interoperability. This means that functions and requirements defined as 'optional' in this document might need to be selected in a specification (in effect made 'mandatory' for the project-specific specification).

A conformant device may offer additional features, as long as they are conformant with the requirements of this document and the standards it references (e.g., NTCIP 2301). For example, a device may support data that has not been defined by this document; however, when exchanged via the NTCIP 2301 protocols, the data shall be properly registered with a valid OBJECT IDENTIFIER under the Global ISO Naming Tree.

> Note: Off-the-shelf interoperability and interchangeability can only be obtained through well documented features broadly supported by the industry as a whole. Designing a system that uses features not defined in a standard or not typically deployed in combination with one another inhibits the goals of interoperability and interchangeability, especially if the documentation of these features is not available for distribution to system integrators. Standards allow the use of additional features to support innovation, which is constantly needed within the industry; but users should be aware of the risks involved with using such features.

3.3.2.2 Backward Compatibility and Support of Different Versions

In this document, the enhancement of security prevents interoperability with prior versions of this document.

3.3.3 Protocol Requirements List (PRL) Table

In addition to the conformance column and the support column, which were discussed in Section 3.3.1, the additional columns in the PRL table are the user needs columns, requirements columns and the additional specifications column.

3.3.3.1 User Needs Column

The PRL is based on the user needs defined in Section 2. The section number and user need name are indicated within this column.

3.3.3.2 Requirements Column

The PRL provides traceability by identifying the requirements defined in Section 3 that are associated with the user needs. The section number and functional requirement name are indicated within this column.

3.3.3.3 Additional Project Requirements Column

The "Additional Specifications" column may (and should) be used by a procurement specification to provide additional notes and requirements for the specified implementation. Default text may already exist in this column, which the user should complete to fully specify the equipment. However, additional text can be added to this field as needed to fully specify a feature.

† Designates that this requirement is composed of several more detailed requirements as defined in the second half of the PRL contained in Section 3.3.4.

User Need Section Number	User Need	FR Section Number	Functional Requirement	Conformance	Support / Project Requirement	Additional Project Requirements
ISO 26048-1, §6.1	Authenticate users			M	Yes	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.2	Control access to data			M	Yes	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.3	Monitor failed access to the field device			M	Yes	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.4	Manage the field device			M	Yes	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
		§3.6.14	Reporting Controller-to-Display Interface Errors		M	Yes
ISO 26048-1, §6.5.1.1	Monitor ambient air temperature			O	Yes / No	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.5.1.2	Monitor ambient relative humidity			O	Yes / No	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.5.1.3	Monitor ambient light			O	Yes / No	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.5.2.1	Monitor field device doors			M	Yes	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.5.2.2	Monitor field device enclosure temperature			M	Yes	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.5.2.3	Monitor field device processor temperature			O	Yes / No	

User Need Section Number	User Need	FR Section Number	Functional Requirement	Conformance	Support / Project Requirement	Additional Project Requirements
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.5.2.4	Monitor field device enclosure humidity			O	Yes / No	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.5.3.1	Monitor field device fans			O	Yes / No	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.5.3.2	Monitor field device heaters			O	Yes / No	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.5.3.3	Monitor field device dehumidifiers			O	Yes / No	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.5.3.4	Monitor field device air conditioners			O	Yes / No	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.5.3.5	Monitor field device thermostat			O	Yes / No	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.5.4.1	Monitor field device power supplies			O	Yes / No	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.5.4.2	Monitor field device mains power			O	Yes / No	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.5.4.3	Monitor field device battery power			O	Yes / No	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.5.4.4	Monitor field device generator power			O	Yes / No	

User Need Section Number	User Need	FR Section Number	Functional Requirement	Conformance	Support / Project Requirement	Additional Project Requirements
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.5.4.5	Monitor field device solar power			O	Yes / No	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.5.4.6	Monitor field device wind power			O	Yes / No	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.5.5.1	Monitor field device auxiliary bi-directional ports			O	Yes / No	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.5.5.2	Monitor field device auxiliary inputs			O	Yes / No	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.5.5.3	Monitor field device auxiliary outputs			O	Yes / No	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.6	Receive notification of triggers firing			O.1 (1..*)	Yes / No	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.7	Manage device specific notifications			deviceNotifications:M	Yes / NA	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.8	Log system events			M	Yes	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.9	Log user-defined data snapshots			O.1 (1..*)	Yes / No	
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
ISO 26048-1, §6.12	Configure a complex device			X	No	

User Need Section Number	User Need	FR Section Number	Functional Requirement	Conformance	Support / Project Requirement	Additional Project Requirements
		See ISO 26048-1 NTFT and FTRT tables (available at https://standards.iso.org/iso/ts/26048/-1/ed-1/en/)				
2.3.2	DMS Characteristics			M	Yes	
2.3.2.1	DMS Types			M	Yes	
2.3.2.1.a	BOS			O.1 (1)	Yes / No	
2.3.2.1.b	CMS			O.1 (1)	Yes / No	
2.3.2.1.c	VMS			O.1 (1)	Yes / No	
2.3.2.2	DMS Technologies			M	Yes	Note that certain combinations of the following technologies might not be supported by any product.
2.3.2.2.a	Fiber Optic			O	Yes / No	
2.3.2.2.b	Light emitting diode (LED)			O	Yes / No	
2.3.2.2.c	Flip disk or shutter			O	Yes / No	
2.3.2.2.d	Lamp matrix			O	Yes / No	
2.3.2.2.e	Drum (rotating, multifaceted cylinder)			O	Yes / No	
2.3.2.3	DMS Display Matrix Configuration			M	Yes	The DMS shall be ___ millimeters wide (0..65535) and ___ millimeters high (0..65535), inclusive of borders. The Sign's Border shall be at least ___ millimeters wide (0..65535) and ___ millimeters high (0..65535).
2.3.2.3.a	Non-matrix			O.2 (1)	Yes / No	
2.3.2.3.b	Matrix			O.2 (1)	Yes / No	The pitch between pixels shall be at least ___ millimeters (0..255).
2.3.2.3.b.i	Full matrix			O.3 (1)	Yes / No	The sign shall be ___ pixels wide (0..65535) and ___ pixels high (0..65535).
2.3.2.3.b.ii	Line matrix			O.3 (1)	Yes / No	The sign shall have ____ lines with each line being ___ pixels wide and ___ pixels high.
2.3.2.3.b.iii	Character matrix			O.3 (1)	Yes / No	The sign shall be ___ characters wide and ___ characters high with each character being ___ pixels wide (0..255), ___ pixels high (0..255).
2.3.2.4	DMS Display Support of Physical Beacons			M	Yes	The DMS shall support the following Beacon configuration:___________ Select one from the following (or define your own):

User Need Section Number	User Need	FR Section Number	Functional Requirement	Conformance	Support / Project Requirement	Additional Project Requirements
						• none • one Beacon • two Beacons with Sync-ed Flash • two Beacons with Opposing Flash • four Beacons with Sync-ed Flash • four Beacons with Alternate Row Flash • four Beacons with Alternate Column Flash • four Beacons with Alternate Diagonal Flash • four Beacons with No Sync-ed Flash • one Beacon Strobe • two Beacon Strobe • four Beacon Strobe
2.5	Features			M	Yes	
2.5.1	Manage the DMS Configuration			M	Yes	
2.5.1.1	Determine DMS Capabilities			M	Yes	
		§3.5.1.1.1	Determine Sign Type and Technology	M	Yes	
2.5.1.2	Determine Sign Display Capabilities			O	Yes / No	
		§3.5.1.2.1.1	Determine the Size of the Sign Face	M	Yes	
		§3.5.1.2.1.2	Determine the Size of the Sign Border	M	Yes	
		§3.5.1.9.1 (Beacons)	Determine Number of Beacons	M	Yes	
		§3.5.1.9.2	Determine Beacon Properties	Beacons:M	Yes / NA	
		§3.5.1.2.1.3	Determine Sign Access and Legend	M	Yes	
		§3.5.3.2.1	Monitor Information about the Currently Displayed Message	O	Yes / No	
		§3.5.1.2.2.1	Determine Sign Face	Matrix:M	Yes / NA	

User Need Section Number	User Need	FR Section Number	Functional Requirement	Conformance	Support / Project Requirement	Additional Project Requirements
			Size in Pixels			
		§3.5.1.2.2.2	Determine Character Size in Pixels	Matrix:M	Yes / NA	
		§3.5.1.2.2.3	Determine Pixel Spacing	Matrix:M	Yes / NA	
		§3.5.1.2.3.1	Determine Maximum Number of Pages	VMS:M	Yes / NA	The DMS shall support at least _____ (1..255) pages for a single message.
		§3.5.1.2.3.2	Determine Maximum Message Length	VMS:M	Yes / NA	The DMS shall support a Multi-String message of at least _____ (0..65535) bytes.
		§3.5.1.2.3.3	Determine Supported Color Schemes	VMS:M	Yes / NA	
		§3.5.1.2.3.4	Determine Message Display Capabilities	VMS:M	Yes / NA	
		§3.5.2.2.2.1	Determine Default Message Display Parameters	VMS:M	Yes / NA	
		§3.6.6	Supplemental Requirements for Message Definition	VMS:M	Yes / NA	
		§3.5.1.3.1	Determine Maximum Number of Fonts Supported	Fonts:M	Yes / NA	See PRL 3.6.1.1.
		§3.5.1.3.3	Determine Maximum Number of Characters per Font	Fonts:M	Yes / NA	
		§3.5.1.3.4	Retrieve a Font Definition	Fonts:M	Yes / NA	
		§3.5.1.4.1	Determine Number of Graphics	Graphics:M	Yes / NA	The DMS shall support at least _____ graphics.
		§3.5.1.4.4	Retrieve a Graphic Definition	Graphics:M	Yes / NA	
		§3.5.3.2.2	Monitor Dynamic Field Values	Fields:M	Yes / NA	

User Need Section Number	User Need	FR Section Number	Functional Requirement	Conformance	Support / Project Requirement	Additional Project Requirements
2.5.1.3	Manage Fonts			VMS:O	Yes / No / NA	
		§3.5.1.3.1	Determine Maximum Number of Fonts Supported	M	Yes	See PRL 3.6.1.1.
		§3.5.1.3.2	Determine Maximum Character Size	M	Yes	The DMS shall support at least ____ characters per font (1...65535).
		§3.5.1.3.3	Determine Maximum Number of Characters per Font	M	Yes	
		§3.5.1.3.4	Retrieve a Font Definition	M	Yes	
		§3.6.1	Supplemental Requirements for Fonts	M	Yes	If desired, the procurement officer should define the fonts or leave this up to the vendor. If officer defines the font(s), attach sheet(s) with definitions. Note: The Project Specifications may ask vendor to propose the fonts.
		§3.5.1.3.5	Configure a Font	O	Yes / No	
		§3.5.1.3.6	Delete a Font	O	Yes / No	
		§3.5.1.3.7	Validate a Font	O	Yes / No	
2.5.1.4	Manage Graphics			VMS:O	Yes / No / NA	
		§3.5.1.4.1	Determine Number of Graphics	M	Yes	The DMS shall support at least _____ graphics.
		§3.5.1.4.2	Determine Maximum Graphic Size	M	Yes	The DMS shall support a maximum graphic size of ___________ bytes.
		§3.5.1.4.3	Determine Available Graphics Memory	M	Yes	The DMS shall support a maximum graphic block size of ______________ bytes.
		§3.5.1.4.4	Retrieve a Graphic Definition	M	Yes	
		§3.6.11	Supplemental Requirements for Graphics	M	Yes	If desired, the procurement officer should define the graphics or leave this up to the vendor. If officer defines the graphic(s), attach sheet(s) with definitions. Note: The Project Specifications may ask vendor to

User Need Section Number	User Need	FR Section Number	Functional Requirement	Conformance	Support / Project Requirement	Additional Project Requirements
						propose the graphics.
		§3.5.1.4.5	Store a Graphic Definition	O	Yes / No	
		§3.5.1.4.6	Delete a Graphic	O	Yes / No	
		§3.5.1.4.7	Validate a Graphic	O	Yes / No	
2.5.1.5	Manage Automatic Brightness			O	Yes / No	
		§3.5.1.5.1	Determine Maximum Number of Light Sensor Levels	M	Yes	
		§3.5.2.3.1	Determine Number of Brightness Levels	M	Yes	
		§3.6.2	Supplemental Requirements for General Illumination Brightness	M	Yes	
		§3.5.1.5.2	Configure Light Output Algorithm	O	Yes / No	
		§3.5.1.5.3	Determine Current Light Output Algorithm	O	Yes / No	
		§3.6.3	Supplemental Requirements for Automatic Brightness Control	O	Yes / No	
2.5.1.6	Configure Speed Limit			O	Yes / No	
		§3.5.1.6	Configure Current Speed Limit	M	Yes	
		§3.5.1.7	Determine Current Speed Limit Configuration	M	Yes	
2.5.1.8	Manage the Display Location			O	Yes / No	
		§3.5.1.8.1	Determine the Sign	M	Yes	

User Need Section Number	User Need	FR Section Number	Functional Requirement	Conformance	Support / Project Requirement	Additional Project Requirements
			Display Location			
		§3.5.1.8.2	Configure the Sign Display Offset Location	M	Yes	
		§3.5.1.8.3	Determine Sign Display Offset Location	M	Yes	
2.5.1.9	Manage Physical Beacons			Beacon:M	Yes / NA	
		§3.5.1.9.1	Determine Number of Beacons	M	Yes	
		§3.5.1.9.2	Determine Beacon Properties	M	Yes	
		§3.5.1.9.3	Configure Beacon Operational Strategies	M	Yes	
		§3.5.1.9.4	Retrieve Beacon Operational Strategies	M	Yes	
		§3.5.1.9.5	Configure Beacon Operation within a Strategy	M	Yes	
		§3.5.1.9.6	Retrieve Beacon Operation within a Strategy	M	Yes	
2.5.2	Control the DMS			M	Yes	
2.5.2.1	Control a DMS from More than One Location			M	Yes	
		§3.5.2.1	Manage Control Source	M	Yes	
		§3.6.4	Supplemental Requirements for Control Modes	M	Yes	
2.5.2.3	Control the Sign Face			M	Yes	

User Need Section Number	User Need	FR Section Number	Functional Requirement	Conformance	Support / Project Requirement	Additional Project Requirements
2.5.2.3.1	Activate and Display a Message			M	Yes	
		§3.5.2.2.1	Activate a Message	M	Yes	
		§3.5.2.2.3.5	Retrieve Message	M	Yes	
		§3.5.2.2.7	Extend Message Activation	M	Yes	
		§3.6.5	Supplemental Requirements for Message Activation Request	M	Yes	
		§3.6.7	Supplemental Requirements for Locally Stored Messages	M	Yes	
		§3.5.2.2.6	Activate a Message with Status	Drum:M	Yes / NA	
2.5.2.3.2	Prioritize Messages			M	Yes	
		§3.5.2.2.1	Activate a Message	M	Yes	
		§3.6.5.4	Supplemental Requirements for Message Activation Priority	M	Yes	
		§3.6.6.4	Priority to Maintain a Message	M	Yes	
		§3.5.2.2.6	Activate a Message with Status	Drum:M	Yes / NA	
		§3.5.2.2.3.3	Define a Message	VMS:M	Yes / NA	
2.5.2.3.3	Define a Message			VMS:M	Yes / NA	
		§3.5.1.2.3.1	Determine Maximum Number of Pages	M	Yes	The DMS shall support at least _____ (1..255) pages for a single message.
		§3.5.1.2.3.2	Determine Maximum Message Length	M	Yes	The DMS shall support a Multi-String message of at least _____ (0..65535) bytes.
		§3.5.1.2.3.3	Determine Supported	M	Yes	

User Need Section Number	User Need	FR Section Number	Functional Requirement	Conformance	Support / Project Requirement	Additional Project Requirements
			Color Schemes			
		§3.5.1.2.3.4	Determine Message Display Capabilities	M	Yes	
		§3.5.2.2.2.1	Determine Default Message Display Parameters	M	Yes	
		§3.5.2.2.3.1	Determine Available Message Types	M	Yes	
		§3.5.2.2.3.2	Determine Available Message Space	M	Yes	
		§3.5.2.2.3.3	Define a Message	M	Yes	
		§3.5.2.2.3.4	Verify Message Contents	M	Yes	
		§3.5.2.2.3.5	Retrieve Message	M	Yes	
		§3.6.6	Supplemental Requirements for Message Definition	M	Yes	
		§3.6.7	Supplemental Requirements for Locally Stored Messages	M	Yes	
		§3.6.8	Supplemental Requirements for Color Scheme	M	Yes	
		§3.6.12	Supplemental Requirements for Page Justification	M	Yes	
		§3.6.13	Supplemental Requirements for Line Justification	M	Yes	
		§3.5.1.2.4	Delete All Messages of a Message Type with One Command	O	Yes / No	

User Need Section Number	User Need	FR Section Number	Functional Requirement	Conformance	Support / Project Requirement	Additional Project Requirements
		§3.5.2.2.2.2	Configure Default Background and Foreground Color	O	Yes / No	
		§3.5.2.2.2.3	Configure Default Flash-On and Flash-Off Times	O	Yes / No	The DMS shall support all flash on times from ____ tenths of a second (0..255) to ____ tenths of a second (0..255) in ____ tenths of a second increments. The DMS shall support all flash off times from ____ tenths of a second (0..255) to ____ tenths of a second (0..255) in ____ tenths of a second increments.
		§3.5.2.2.2.4	Configure Default Font	O	Yes / No	
		§3.5.2.2.2.5	Configure Default Line Justification	O	Yes / No	
		§3.5.2.2.2.6	Configure Default Page Justification	O	Yes / No	
		§3.5.2.2.2.7	Configure Default Page On-Time and Page Off-Time	O	Yes / No	The DMS shall support all page on times from ____ tenths of a second (1..255) to ____ tenths of a second (1..255) in ____ tenths of a second increments. The DMS shall support all page off times from ____ tenths of a second (0..255) to ____ tenths of a second (0..255) in ____ tenths of a second increments.
		§3.5.2.2.2.8	Configure Default Character Set	O	Yes / No	
		§3.5.1.3.1	Determine Maximum Number of Fonts Supported	Fonts:M	Yes / NA	See PRL 3.6.1.1.
		§3.5.1.3.3	Determine Maximum Number of Characters per Font	Fonts:M	Yes / NA	
		§3.5.1.3.4	Retrieve a Font Definition	Fonts:M	Yes / NA	
		§3.5.1.4.1	Determine Number of Graphics	Graphics:M	Yes / NA	The DMS shall support at least _____ graphics.

User Need Section Number	User Need	FR Section Number	Functional Requirement	Conformance	Support / Project Requirement	Additional Project Requirements
		§3.5.1.4.4	Retrieve a Graphic Definition	Graphics:M	Yes / NA	
		ISO 26048-1, §8.2.1	UTC clock	LocalTime:M	Yes / NA	
		ISO 26048-1, §8.2.2	Local clock	LocalTime:M	Yes / NA	
		ISO 26048-1, §8.2.3	Daylight saving time	LocalTime:M	Yes / NA	
		§3.6.1	Supplemental Requirements for Fonts	Fonts:M	Yes	If desired, the procurement officer should define the fonts or leave this up to the vendor. If officer defines the font(s), attach sheet(s) with definitions. Note: The Project Specifications may ask vendor to propose the fonts.
		§3.6.11	Supplemental Requirements for Graphics	Graphics:M	Yes	If desired, the procurement officer should define the graphics or leave this up to the vendor. If officer defines the graphic(s), attach sheet(s) with definitions. Note: The Project Specifications may ask vendor to propose the graphics.
2.5.2.3.4	Blank a Sign			M	Yes	
		§3.5.2.2.1	Activate a Message	M	Yes	
		§3.6.5	Supplemental Requirements for Message Activation Request	M	Yes	
		§3.5.2.2.6	Activate a Message with Status	Drum:M	Yes / NA	
2.5.2.3.5	Schedule Messages for Display			O	Yes / No	
		ISO 26048-1, §8.1	Action feature	M	Yes	
		ISO 26048-1, §8.2.1	UTC clock	M	Yes	
		ISO 26048-1, §8.2.2	Local clock	M	Yes	
		ISO 26048-1, §8.2.3	Daylight saving time	M	Yes	
		§3.5.2.2.1	Activate a Message	M	Yes	
		§3.5.2.2.4.1	Configure a DMS Message Action	M	Yes	
		§3.5.3.3.8	Retrieve a DMS	M	Yes	

User Need Section Number	User Need	FR Section Number	Functional Requirement	Conformance	Support / Project Requirement	Additional Project Requirements
			Message Action			
		§3.5.3.3.9	Determine Number of DMS Message Actions	M	Yes	
		§3.6.5	Supplemental Requirements for Message Activation Request	M	Yes	
		§3.6.10	Supplemental Requirements for Scheduling	M	Yes	
		ISO 26048-1, §8.6	Day plan feature	O.2(1..*)	Yes / No	
		ISO 26048-1, §8.14	Scheduled trigger feature	O.2(1..*)	Yes / No	
		§3.5.2.2.6	Activate a Message with Status	Drum:M	Yes / NA	
2.5.2.3.6	Change Message Display Based on an Internal Event			O	Yes / No	
2.5.2.3.6.1	Change Message Based on Power Loss			Flip/Shutter OR Drum:O.4(1..*)	Yes / No / NA	
		§3.5.2.2.5.1.3	Configure Message for Power Loss Event	M	Yes	
		§3.5.3.3.4	Monitor Power Loss Message	M	Yes	
		§3.6.5.1	Supplemental Requirements for Message Activation	M	Yes	
2.5.2.3.6.2	Change Message Based on Short Power Loss Recovery			O.4(1..*)	Yes / No	
		§3.5.2.2.5.1.1	Configure Message for Short Power Loss Recovery Event	M	Yes	Power recovery messages require support for changeable memory
		§3.5.3.3.2	Monitor Short Power	M	Yes	

User Need Section Number	User Need	FR Section Number	Functional Requirement	Conformance	Support / Project Requirement	Additional Project Requirements
			Recovery Message			
		§3.6.5.1	Supplemental Requirements for Message Activation	M	Yes	
2.5.2.3.6.3	Change Message Based on Long Power Loss Recovery			O.4(1..*)	Yes / No	
		§3.5.2.2.5.1.2	Configure Message for Long Power Loss Recovery Event	M	Yes	Power recovery messages require support for changeable memory
		§3.5.3.3.3	Monitor Long Power Recovery Message	M	Yes	
		§3.6.5.1	Supplemental Requirements for Message Activation	M	Yes	
2.5.2.3.6.4	Change Message Based on Controller Reset			O.4(1..*)	Yes / No	
		§3.5.2.2.5.1.4	Configure Message for Controller Reset Event	M	Yes	
		§3.5.3.3.5	Monitor Reset Message	M	Yes	
		§3.6.5.1	Supplemental Requirements for Message Activation	M	Yes	
2.5.2.3.6.5	Change Message Based on Communication Loss			O.4(1..*)	Yes / No	
		§3.5.2.2.5.1.5	Configure Message for Communications Loss Event	M	Yes	
		§3.5.3.3.6	Monitor Communications Loss Message	M	Yes	
		§3.6.5.1	Supplemental Requirements for Message Activation	M	Yes	

User Need Section Number	User Need	FR Section Number	Functional Requirement	Conformance	Support / Project Requirement	Additional Project Requirements
2.5.2.3.6.6	Change Message Based on End Message Duration			O.4(1..*)	Yes / No	
		§3.5.2.2.5.1.6	Configure Message for End Message Display Duration Event	M	Yes	
		§3.5.3.3.7	Monitor End Duration Message	M	Yes	
		§3.6.5.1	Supplemental Requirements for Message Activation	M	Yes	
2.5.2.3.7	Change Message Display Based on a Conditional Trigger			O	Yes / No	
		ISO 26048-1, §8.1	Action feature	M	Yes	
		ISO 26048-1, §8.4	Conditional trigger feature	M	Yes	
		§3.5.2.2.4.2	Configure a Callable Message	M	Yes	
		§3.5.3.3.1	Monitor a Callable Message	M	Yes	
		§3.5.3.3.10	Determine Number of Callable Messages	M	Yes	
		ISO 26048-1, §8.16	SNMP target feature	O	Yes / No	Select if conditional trigger needs to be able to evaluate data from external devices.
2.5.2.5	Control the Brightness Output			Lamp OR LED OR Fiber:M	Yes / NA	
		§3.5.2.3.1	Determine Number of Brightness Levels	M	Yes	
		§3.5.2.3.5	Switch Brightness Control Modes	O	Yes / No	
		§3.6.2	Supplemental Requirements for General Illumination	O	Yes / No	

User Need Section Number	User Need	FR Section Number	Functional Requirement	Conformance	Support / Project Requirement	Additional Project Requirements
			Brightness			
		§3.5.2.3.3	Manually Direct-Control Brightness	O.3(1..*)	Yes / No	
		§3.5.2.3.4	Manually Index-Control Brightness	O.3(1..*)	Yes / No	
		§3.5.2.3.2	Determine Current Composite Photocell Value	AutoBright:M	Yes / NA	
		§3.6.3	Supplemental Requirements for Automatic Brightness Control	AutoBright:M	Yes / NA	
2.5.2.6	Perform Preventative Maintenance			Fiber OR Flip/Shutter:O	Yes / No / NA	
		ISO 26048-1, §8.2.1	UTC clock	M	Yes	
		ISO 26048-1, §8.2.2	Local clock	M	Yes	
		§3.5.2.4.1	Manage the Exercise of Pixels	M	Yes	
		§3.5.2.4.2	Determine Pixel Service Settings	M	Yes	
		§3.6.6.6	Pixel Service Flag	M	Yes	
		ISO 26048-1, §8.2.3	Daylight saving time	O	Yes / No	
2.5.3	Monitor the Status of the DMS			M	Yes	
2.5.3.1	Perform Diagnostics			M	Yes	
2.5.3.1.4	Monitor Message Errors			M	Yes	
		§3.5.3.1.4	Monitor Message Activation Error Details	M	Yes	
2.5.3.1.6	Monitor the Sign Control Source			M	Yes	
		§3.5.3.1.5	Monitor the Sign's Control Source	M	Yes	

User Need Section Number	User Need	FR Section Number	Functional Requirement	Conformance	Support / Project Requirement	Additional Project Requirements
2.5.3.1.7	Monitor Attached Speed Detectors			O	Yes / No	
		ISO 26048-1, §8.18	Supplemental roadside sensors and actuators (SRSA) feature	M	Yes	
		§3.6.9.4	Supplemental Requirements for Speed Detectors	M	Yes	
2.5.3.1.10	Monitor Automatic Blanking of Sign			O	Yes / No	
		ISO 26048-1, §8.18	Supplemental roadside sensors and actuators (SRSA) feature	M	Yes	For at least one FET entry
		§3.5.3.1.1	Provide General DMS Error Status Information	M	Yes	
		§3.5.3.1.6	Determine Critical Temperature	M	Yes	
		§3.6.15	Critical Temperature Trigger	M	Yes	
		§3.6.16	Deactivate Action Group	M	Yes	
		§3.6.17	Reactivate Action Group	M	Yes	
		§3.5.3.2.1	Monitor Information about the Currently Displayed Message	O	Yes / No	
		§3.5.3.2.2	Monitor Dynamic Field Values	Fields:M	Yes / NA	
		§3.5.3.1.2.1	Activate Pixel Testing	Matrix:M	Yes / NA	
		§3.5.3.1.2.3	Monitor Pixel Error Map	Matrix:M	Yes / NA	

User Need Section Number	User Need	FR Section Number	Functional Requirement	Conformance	Support / Project Requirement	Additional Project Requirements
		§3.6.9.2	Supplemental Requirements for Lamp Status	Lamp OR Fiber: M	Yes / NA	
		§3.6.6	Supplemental Requirements for Message Definition	VMS:M	Yes / NA	
2.5.3.1.15	Monitor General DMS Status			M	Yes	
		§3.5.3.1.1	Provide General DMS Error Status Information	M	Yes	
2.5.3.1.16	Monitor Pixel Errors			Matrix:M	Yes / NA	
		§3.5.3.1.2.1	Activate Pixel Testing	M	Yes	
		§3.5.3.1.2.2	Monitor Pixel Error Count	M	Yes	
		§3.5.3.1.2.3	Monitor Pixel Error Map	M	Yes	
		§3.5.3.1.2.4	Monitor Pixel Error Details	M	Yes	
2.5.3.1.17	Monitor Photocells			AutoBright:M	Yes / NA	
		ISO 26048-1, §8.18	Supplemental roadside sensors and actuators (SRSA) feature	M	Yes	
		§3.6.9.1	Supplemental Requirements for Photocells	M	Yes	
2.5.3.1.18	Monitor Lamp Status			Lamp OR Fiber:M	Yes / NA	
		ISO 26048-1, §8.18	Supplemental roadside sensors and actuators (SRSA) feature	M	Yes	
		§3.5.3.1.3	Monitor Lamp Error	M	Yes	

User Need Section Number	User Need	FR Section Number	Functional Requirement	Conformance	Support / Project Requirement	Additional Project Requirements
			Details			
		§3.6.9.2	Supplemental Requirements for Lamp Status	M	Yes	
2.5.3.1.19	Monitor Drum Status			Drum:O	Yes / No / NA	
		ISO 26048-1, §8.18	Supplemental roadside sensors and actuators (SRSA) feature	M	Yes	
		§3.6.9.3	Supplemental Requirements for Sign Drum	M	Yes	
2.5.3.2	Monitor the Current Message			M	Yes	
		§3.5.3.2.1	Monitor Information about the Currently Displayed Message	O	Yes / No	
		§3.5.3.2.2	Monitor Dynamic Field Values	Fields:M	Yes / NA	
		§3.6.6	Supplemental Requirements for Message Definition	VMS:M	Yes / NA	
2.6	Security			M	Yes	
2.6.2	Conformant Security Environment			O.5 (1)	Yes / No	
		§3.7.1	Conformant Security	M	Yes	
2.6.3	Consistent Security Environment			O.5 (1)	Yes / No	
		§3.7.2	Consistent Security	M	Yes	

NOTE—The user needs for "Record a series of snapshots", "Issue trigger-based commands", and "Efficient excahnge of data", as defined in ISO 26048-1, 6.10, 6.11, and 6.13, are not envisioned to be needed for message sign deployments, but support for these user needs are not prohibited. They are omitted from the PRL to reduce the risk of procurements inadvertently selecting features that are expected to be uncommon and which could significantly increase procurement costs by specifying an unnecessary feature that is not widely used.

3.3.4 Protocol Requirements List – Supplemental Table

Protocol Requirements List (PRL) Suppllemetal Table						
Req ID	**Requirement**	**Req ID**	**Requirement**	**Conformance**	**Support**	**Additional Specifications**
	Supplemental Requirements					
3.6.1	Supplemental Requirements for Fonts					
		3.6.1.1	Support for a Number of Fonts	M	Yes	The DMS shall support at least ____ fonts (1..255). Note: The specification may optionally specify the fonts to be stored in the sign controller upon initial delivery by using an additional attached sheet to define the desired pixel-by-pixel bitmaps of each character of each font.
3.6.2	Supplemental Requirements for General Illumination Brightness					
		3.6.2.1	Support a Number of Brightness Levels	M	Yes	The DMS shall support at least _____ brightness levels (1..255).
3.6.3	Supplemental Requirements for Automatic Brightness Control					
		3.6.3.1	Automatically Control Brightness	M	Yes	
		3.6.3.2	Inhibit Flickering of Message Brightness	O	Yes / No	
		3.6.3.3	Support a Number of Light Sensor Levels	M	Yes	The DMS shall support at least _____ light sensor levels (0..65535).
3.6.3.4	Supplemental Requirements for Control Modes					
		3.6.4.1	Support Central Control Mode	M	Yes	
		3.6.4.2	Support Local Control Mode	M	Yes	
		3.6.4.3	Support Central Override Control Mode	O	Yes / No	
		3.6.4.4	Processing Requests from Multiple Sources	M	Yes	

Protocol Requirements List (PRL) Suppllemetal Table						
Req ID	**Requirement**	**Req ID**	**Requirement**	**Conformance**	**Support**	**Additional Specifications**
3.6.5	Supplemental Requirements for Message Activation Request					
		3.6.5.1	Supplemental Requirements for Internal Message Activation	M	Yes	
		3.6.5.1.1	Activate Any Message	M	Yes	
		3.6.5.1.2	Preserve Message Integrity	VMS:M	Yes / NA	
		3.6.5.1.3	Ensure Proper Message Content	M	Yes	
		3.6.5.2	Indicate Message Display Duration	M	Yes	
		3.6.5.3	Indicate Message Display Requester ID	M	Yes	
		3.6.5.4	Supplemental Requirements for Message Activation Priority	M	Yes	
3.6.6	Supplemental Requirements for Message Definition					
		3.6.6.1	Identify Message to Define	M	Yes	
		3.6.6.2	Define Message Content	M	Yes	
		3.6.6.2.1	Support Multi-Page Messages	O	Yes / No	The DMS shall support at least ___ pages (1..255) per message.
		3.6.6.2.2	Support Page Justification	O	Yes / No	
		3.6.6.2.2.1	Support for One Page Justification within a Message	O.7 (1)	Yes / No	
		3.6.6.2.2.2	Support for Multiple Page Justifications within a Message	O.7 (1)	Yes / No	
		3.6.6.2.3	Support Multiple Line Messages	O	Yes / No	The DMS shall support at least ___ lines (1..255) per page.
		3.6.6.2.4	Support Line Justification	O	Yes / No	
		3.6.6.2.4.1	Support for a Single Line	O.8 (1)	Yes / No	

Protocol Requirements List (PRL) Suppllemetal Table						
Req ID	**Requirement**	**Req ID**	**Requirement**	**Conformance**	**Support**	**Additional Specifications**
			Justification within a Message			
		3.6.6.2.4.2	Support Line Justification on a Page-by-Page Basis	O.8 (1)	Yes / No	
		3.6.6.2.4.3	Support Line Justification on a Line-by-Line Basis	O.8 (1)	Yes / No	
		3.6.6.2.5	Support Color	O	Yes / No	
		3.6.6.2.5.1	Support a Single Color Combination per Message	O.9 (1)	Yes / No	
		3.6.6.2.5.2	Support a Color Combination for each Page	O.9 (1)	Yes / No	
		3.6.6.2.5.3	Support a Color Combination for each Character within a Message	O.9 (1)	Yes / No	
		3.6.6.2.6	Support Font Commands	O	Yes / No	
		3.6.6.2.6.1	Support One Font within a Message	O.10 (1)	Yes / No	
		3.6.6.2.6.2	Support One Font per Page within a Message	O.10 (1)	Yes / No	
		3.6.6.2.6.3	Support Character-by-Character Selection of Fonts within a Message	O.10 (1)	Yes / No	
		3.6.6.2.7	Support Moving Text	O	Yes / No	
		3.6.6.2.8	Support Character Spacing	O	Yes / No	
		3.6.6.2.9	Support Customizable Page Display Times in a Message	O	Yes / No	
		3.6.6.2.10 (Flash)	Support Flashing	O	Yes / No	
		3.6.6.2.10.1	Support Character-by-Character Flashing	O.11 (1)	Yes / No	
		3.6.6.2.10.2	Support Line-by-Line Flashing	O.11 (1)	Yes / No	

Protocol Requirements List (PRL) Suppllemetal Table						
Req ID	**Requirement**	**Req ID**	**Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.6.6.2.10.3	Support Page-by-Page Flashing	O.11 (1)	Yes / No	
		3.6.6.2.11	Support Customizable Flashing Times within a Message	Flash:O	Yes / No / NA	
		3.6.6.2.12	Support Hexadecimal Character	O	Yes / No	
		3.6.6.2.13 (Fields)	Support Message Data Fields	O	Yes / No	
		3.6.6.2.13.1 (Time)	Support Current Time Field without AM/PM Field	Fields:O.12 (1..*)	Yes / No	
		3.6.6.2.13.2 (Time)	Support Current Time with AM/PM Field	Fields:O.12 (1..*)	Yes / No	
		3.6.6.2.13.3 (Time)	Support Current Time with am/pm Field	Fields:O.12 (1..*)	Yes / No	
		3.6.6.2.13.4 (Temp)	Support Current Temperature Field	Fields:O.12 (1..*)	Yes / No	
		3.6.6.2.13.5	Support Detected Vehicle Speed Field	Fields AND Speed:O.12 (1..*)	Yes / No / NA	
		3.6.6.2.13.6 (DoW)	Support Current Day of Week Field	Fields:O.12 (1..*)	Yes / No	
		3.6.6.2.13.7 (DoM)	Support Current Day of Month Field	Fields:O.12 (1..*)	Yes / No	
		3.6.6.2.13.8 (Month)	Support Current Month of Year Field	Fields:O.12 (1..*)	Yes / No	
		3.6.6.2.13.9 (Year)	Support Current Year Field	Fields:O.12 (1..*)	Yes / No	
		3.6.6.2.13.10	Support User-Definable Field	Fields:O.12 (1..*)	Yes / No	Note: For interoperability reasons, it is not recommended to use this field.
		3.6.6.2.13.11	Data Field Refresh Rate	Fields:M	Yes	The DMS shall update the fields at least every ____ seconds.

Protocol Requirements List (PRL) Suppllemetal Table						
Req ID	**Requirement**	**Req ID**	**Requirement**	**Conformance**	**Support**	**Additional Specifications**
		3.6.6.2.14	Support of Graphics	O	Yes / No	
		3.6.6.2.15	Specify Location of Message Display	O	Yes / No	
		3.6.6.2.16	Support of Text	M	Yes	
		3.6.6.2.16.1	Support of Textual Content	M	Yes	
		3.6.6.2.16.2	Support of Message Lengths Compatible with Sign Face	M	Yes	
		3.6.6.2.17	Support of Manufacturer Specific Message Definitions	O	Yes / No	The DMS shall support a manufacturer-specific tag ____________ [msx,y]. Note: For interoperability reasons, it is not recommended that this field be selected.
		3.6.6.3	Identify Message Owner	M	Yes	
		3.6.6.4	Priority to Maintain a Message	M	Yes	
		3.6.6.5	Physical Beacon Activation Flag	Beacons:M	Yes / NA	
		3.6.6.6	Pixel Service Flag	Fiber OR Flip/Shutter:M	Yes / NA	
		3.6.6.7	Message Status	M	Yes	
3.6.7	Supplemental Requirements for Locally Stored Messages					
		3.6.7.1	Support Permanent Messages	VMS:O;M	Yes / No / NA	The DMS shall support at least ___ different permanent messages. (0..65535) The Permanent Messages are: (attach separate sheet defining the message number and the content and layout of each permanent message)
		3.6.7.2	Support Changeable Messages	VMS:O.13 (1..*)	Yes / No / NA	The DMS shall support ____ changeable messages (0..65535) and ______ bytes of changeable memory (0..4294967295).
		3.6.7.3	Support Volatile Messages	VMS:O.13 (1..*)	Yes / No / NA	The DMS shall support ____ volatile messages (0..65535) and _______ bytes

Protocol Requirements List (PRL) Suppllemetal Table

Req ID	Requirement	Req ID	Requirement	Conformance	Support	Additional Specifications
						of volatile memory (0..4294967295). An equivalent number of changeable messages and memory may be / shall not be (select one) substituted for volatile messages per the requirements of NTCIP 1203 v02.
3.6.8	Supplemental Requirements for Color Scheme					
		3.6.8.1	Support 256 Shades Scheme	O.14 (1)	Yes / No	
		3.6.8.3	Support 24-Bit Color Scheme	O.14 (1)	Yes / No	
		3.6.8.4	Support Single Color	M	Yes	
3.6.9	Supplemental Requirements for Monitoring Subsystems					The primary power source shall be ______________ These tests shall be performed at least once every ____ seconds.
3.6.10	Supplemental Requirements for Scheduling					
		3.6.10.1	Support a Number of DMS Actions	M	Yes	The DMS shall support at least ____ DMS actions (0..255) for the schedule.
		3.6.10.2	Support the DMS Action for the Scheduler	M	Yes	
		3.6.10.3	Implement DMS Action When Called	M	Yes	
3.6.11	Supplemental Requirements for Graphics					
		3.6.11.1	Support for a Number of Graphics	M	Yes	The DMS shall support at least ___ graphics (0..255).
		3.6.11.2	Support for Graphic Memory	M	Yes	The DMS shall support at least ______ bytes (0..4294967295) of graphic memory.

Protocol Requirements List (PRL) Suppllemetal Table

Req ID	Requirement	Req ID	Requirement	Conformance	Support	Additional Specifications
3.6.12	Supplemental Requirements for Page Justification					
		3.6.12.1	Support top Page Justification	O.16 (1..*)	Yes / No	
		3.6.12.2	Support middle Page Justification	O.16 (1..*)	Yes / No	
		3.6.12.3	Support bottom Page Justification	O.16 (1..*)	Yes / No	
3.6.13	Supplemental Requirements for Line Justification					
		3.6.13.1	Support left Line Justification	O.17 (1..*)	Yes / No	
		3.6.13.2	Support center Line Justification	O.17 (1..*)	Yes / No	
		3.6.13.3	Support right Line Justification	O.17 (1..*)	Yes / No	
		3.5.13.4	Support full Line Justification	O.17 (1..*)	Yes / No	

3.3.5 MULTI Field Traceability Matrix

MULTI Field Traceability Matrix				
Requirement ID	Requirement	MULTI Tag ID	MULTI Tag Name	MULTI Tag
3.6.6.2.1	Support Multi-Page Messages			
		6.4.15	New Page	[np]
3.6.6.2.2	Support Page Justification			
		6.4.11	Justification - Page	[jpx]
		6.4.11	Top Justification	[jp2]
		6.4.11	Middle Justification	[jp3]
		6.4.11	Bottom Justification	[jp4]
3.6.6.2.2.1	Support for One Page Justification within a Message			
		6.4.11	Justification - Page	[jpx]
		6.4.11	Top Justification	[jp2]
		6.4.11	Middle Justification	[jp3]
		6.4.11	Bottom Justification	[jp4]
3.6.6.2.2.2	Support for Multiple Page Justifications within a Message			
		6.4.11	Justification - Page	[jpx]
		6.4.11	Top Justification	[jp2]
		6.4.11	Middle Justification	[jp3]
		6.4.11	Bottom Justification	[jp4]
3.6.6.2.3	Support Multiple Line Messages			
		6.4.14	New Line	[nlx]
3.6.6.2.4	Support Line Justification			
		6.4.10	Justification - Line	[jlx]
		6.4.10	Left Justification	[jl2]
		6.4.10	Center Justification	[jl3]
		6.4.10	Right Justification	[jl4]

MULTI Field Traceability Matrix				
Requirement ID	**Requirement**	**MULTI Tag ID**	**MULTI Tag Name**	**MULTI Tag**
		6.4.10	Full Justification	[jl5]
3.6.6.2.4.1	Support for a Single Line Justification within a Message			
		6.4.10	Justification - Line	[jlx]
		6.4.10	Left Justification	[jl2]
		6.4.10	Center Justification	[jl3]
		6.4.10	Right Justification	[jl4]
		6.4.10	Full Justification	[jl5]
3.6.6.2.4.2	Support Line Justification on a Page-by-Page Basis			
		6.4.10	Justification - Line	[jlx]
		6.4.10	Left Justification	[jl2]
		6.4.10	Center Justification	[jl3]
		6.4.10	Right Justification	[jl4]
		6.4.10	Full Justification	[jl5]
3.6.6.2.4.3	Support Line Justification on a Line-by-Line Basis			
		6.4.10	Justification - Line	[jlx]
		6.4.10	Left Justification	[jl2]
		6.4.10	Center Justification	[jl3]
		6.4.10	Right Justification	[jl4]
		6.4.10	Full Justification	[jl5]
3. 6.6.2.5	Support Color			
3.6.6.2.5.1	Support a Single Color Combination per Message			
		6.4.3	Color Foreground	[cfx] or [cfr,g,b]
		6.4.2	Page Background Color	[pbz] or [pbr,g,b]
3.6.6.2.5.2	Support a Color Combination for each Page			

MULTI Field Traceability Matrix				
Requirement ID	**Requirement**	**MULTI Tag ID**	**MULTI Tag Name**	**MULTI Tag**
		6.4.3	Color Foreground	[cfx] or [cfr,g,b]
		6.4.2	Page Background Color	[pbz] or [pbr,g,b]
3.6.6.2.5.3	Support a Color Combination for each Character within a Message			
		6.4.3	Color Foreground	[cfx] or [cfr,g,b]
		6.4.2	Page Background Color	[pbz] or [pbr,g,b]
3.6.6.2.5.4	Color for each Pixel within a Message			
		6.4.3	Color Foreground	[cfx] or [cfr,g,b]
		6.4.2	Page Background Color	[pbz] or [pbr,g,b]
		6.4.4	Color Rectangle	[crx,y,w,h,r,g,b] or [crx,y,w,h,z]
3.6.6.2.6	Support Font Commands			
		6.4.7	Font	[fox]
3.6.6.2.6.1	Support One Font within a Message			
		6.4.7	Font	[fox]
3.6.6.2.6.2	Support One Font per Page within a Message			
		6.4.7	Font	[fox]
3.6.6.2.6.3	Support Character by Character Selection of Fonts within a Message			
		6.4.7	Font	[fox]
3.6.6.2.7	Support Moving Text			
		6.4.13	Moving Text	[mvtdw,s,r,text]
3.6.6.2.8	Support Character Spacing			
		6.4.17	Spacing - Character	[scx]
3.6.6.2.9	Support Customizable Page Display Times in a Message			

MULTI Field Traceability Matrix				
Requirement ID	**Requirement**	**MULTI Tag ID**	**MULTI Tag Name**	**MULTI Tag**
		6.4.16	Page Time	[ptxoy]
3.6.6.2.11	Support Customizable Flashing Times within a Message			
		6.4.6	Flash Time	[fltxoy]
3.6.6.2.10	Support Flashing			
		6.4.6	Flash Time	[fltxoy]
3.6.6.2.10.1	Support Character-by-Character Flashing			
		6.4.5 6	Flash Time	[fltxoy]
3.6.6.2.10.2	Support Line-by-Line Flashing			
		6.4.5 6	Flash Time	[fltxoy]
3.6.6.2.10.3	Support Page-by-Page Flashing			
		6.4.5 6	Flash Time	[fltxoy]
3.6.6.2.12	Support Hexadecimal Character			
		6.4.8 9	Hexadecimal Character	[hcx]
3.6.6.2.13	Support Message Data Fields			
		6.4.3 5	Local Time 12 Hour	[f1,y]
		6.4.3 5	Local Time 24 Hour	[f2,y]
		6.4.3 5	Ambient Temperature Celsius	[f3,y]
		6.4.3 5	Ambient Temperature Fahrenheit	[f4,y]
		6.4.3 5	Speed km/h	[f5,y]
		6.4.3 5	Speed mph	[f6,y]
		6.4.3 5	Day of Week	[f7,y]
		6.4.3 5	Date of Month	[f8,y]
		6.4.3 5	Month of Year	[f9,y]
		6.4.3 5	Year 2 Digit	[f10,y]
		6.4.3 5	Year 4 Digit	[f11,y]
		6.4.3 5	Local time, 12 hour format with capital AM/PM indicator present	[f12,y]

MULTI Field Traceability Matrix				
Requirement ID	**Requirement**	**MULTI Tag ID**	**MULTI Tag Name**	**MULTI Tag**
		6.4.3 5	Local time, 12 hour format with lowercase am/pm indicator present	[f13,y]
3.6.6.2.13.1	Support Current Time Field without AM/PM Field			
		6.4.3 5	Local Time 12 Hour	[f1,y]
		6.4.3 5	Local Time 24 Hour	[f2,y]
3.6.6.2.13.4	Support Current Temperature Field			
		6.4.5	Ambient Temperature Celsius	[f3,y]
		6.4.5	Ambient Temperature Fahrenheit	[f4,y]
3.6.6.2.13.5	Support Detected Vehicle Speed Field			
		6.4.5	Speed km/h	[f5,y]
		6.4.5	Speed mph	[f6,y]
3.6.6.2.13.6	Support Current Day of Week Field			
		6.4.5	Day of Week	[f7,y]
3.6.6.2.13.7	Support Current Day of Month Field			
		6.4.5	Date of Month	[f8,y]
3.6.6.2.13.8	Support Current Month of Year Field			
		6.4.5	Month of Year	[f9,y]
3.6.6.2.13.9	Support Current Year Field			
		6.4.5	Year 2 Digit	[f10,y]
		6.4.5	Year 4 Digit	[f11,y]
3.6.6.2.13.2	Support Current Time with uppercase AM/PM Field			
		6.4.5	Local time, 12 hour format with capital AM/PM indicator present	[f12,y]
3.6.6.2.13.3	Support Current Time with lowercase am/pm			
		6.4.5	Local time, 12 hour format with lowercase am/pm indicator present	[f13,y]
3.6.6.2.13.10	Support User-Definable Field			

MULTI Field Traceability Matrix				
Requirement ID	**Requirement**	**MULTI Tag ID**	**MULTI Tag Name**	**MULTI Tag**
		6.4.5	User-Definable Field	[f50,y] to [f99,y]
3.6.6.2.13.11	Data Field Refresh Rate			
		6.4.5	Fields	[fx,y]
3.6.6.2.14	Support of Graphics			
		6.4.8	Graphic	[gn] or [gn,x,y] or [gn,x,y,cccc]
3.6.6.2.15	Specify Location of Message Display			
		6.4.18	Cursor Placement / XY LocationText Rectangle	[trx,y,w,h]
		6.4.2	Page Background Color	[pbz] or [pbr,g,b]
		6.4.3	Color Foreground	[cfx]
		6.4.4	Color Rectangle	[crx,y,w,h,r,g,b] or [crx,y,w,h,z]
3.6.8.2	Support Classic NTCIP Color Scheme			
		6.4.2	Page Background Color	[pbz] or [pbr,g,b]
		6.4.3	Color Foreground	[cfx] or [cfr,g,b]
		6.4.4	Color Rectangle	[crx,y,w,h,r,g,b] or [crx,y,w,h,z]
3.6.8.3	Support 24-Bit Color Scheme			
		6.4.2	Page Background Color	[pbz] or [pbr,g,b]
		6.4.3	Color Foreground	[cfx]
		6.4.4	Color Rectangle	[crx,y,w,h,r,g,b] or [crx,y,w,h,z]
3.6.8.4	Support Single Color			
		6.4.2	Page Background Color	[pbz] or [pbr,g,b]

MULTI Field Traceability Matrix				
Requirement ID	Requirement	MULTI Tag ID	MULTI Tag Name	MULTI Tag
		6.4.3	Color Foreground	[cfx]
3.6.12	Supplemental Requirements for Page Justification			
3.6.12.1	Support top Page Justification			
		6.4.11	Top Justification	[jp2]
3.6.12.2	Support middle Page Justification			
		6.4.11	Middle Justification	[jp3]
3.6.12.3	Support bottom Page Justification			
		6.4.11	Bottom Justification	[jp4]
3.6.13	Supplemental Requirements for Line Justification			
3.6.13.1	Support left Line Justification			
		6.4.10	Left Justification	[jl2]
3.6.13.2	Support center Line Justification			
		6.4.10	Center Justification	[jl3]
3.6.13.3	Support right Line Justification			
		6.4.10	Right Justification	[jl4]
3.6.13.4	Support full Line Justification			
		6.4.10	Full Justification	[jl5]

3.4 Common Requirements

Requirements for common NTCIP capabilities are provided in ISO 26048-1.
Note—This subsection is primarily present so that all other requirements can maintain numbering consistency with prior versions.

3.5 Data Exchange Requirements

The operation of a sign has been categorized into three major areas:

a) Manage the DMS configuration
b) Control the DMS
c) Monitor the status of the DMS

In the Concept of Operations (Section 2), each of these major areas has been broken down into sub-items. The Data Exchange Requirements also follow this structure.

3.5.1 Manage the DMS Configuration

Requirements for managing DMS configuration are provided in the following subsections.

3.5.1.1 Identify DMS

Requirements for identifying the DMS are provided in the following subsections.

3.5.1.1.1 Determine Sign Type and Technology

The DMS shall allow a management station to determine its type and technology.

3.5.1.2 Determine Message Display Capabilities

Requirements for determining the message display capabilities of the DMS are provided in the following subsections.

3.5.1.2.1 Determine Basic Message Display Capabilities

Requirements for determining the basic message display capabilities of the sign face are provided in the following subsections.

3.5.1.2.1.1 Determine the Size of the Sign Face

The DMS shall allow a management station to determine the height and width of the sign face.

3.5.1.2.1.2 Determine the Size of the Sign Border

The DMS shall allow a management station to determine the size of the horizontal and vertical border around the sign face.

3.5.1.2.1.3 Determine Sign Access and Legend

The DMS shall allow a management station to determine the access mechanism to the sign internal components and whether the DMS has a legend.

3.5.1.2.2 Determine Matrix Capabilities

Requirements for determining the detailed matrix capabilities of the sign are provided in the following subsections.

3.5.1.2.2.1 Determine Sign Face Size in Pixels

The DMS shall allow a management station to determine the height and width of the sign face in pixels.

3.5.1.2.2.2 Determine Character Size in Pixels

The DMS shall allow a management station to determine the height and width of a character in pixels.

3.5.1.2.2.3 Determine Pixel Spacing

The DMS shall allow a management station to determine the spacing of pixels (pitch).

3.5.1.2.3 Determine VMS Message Display Capabilities

Requirements for determining the detailed capabilities of the VMS are provided in the following subsections.

3.5.1.2.3.1 Determine Maximum Number of Pages

The DMS shall allow a management station to determine the maximum number of pages that can be included in a single message.

3.5.1.2.3.2 Determine Maximum Message Length

The DMS shall allow a management station to determine the maximum length for a downloadable message.

3.5.1.2.3.3 Determine Supported Color Schemes

The DMS shall allow a management station to determine whether the sign supports a color scheme other than the ' monochrome1bit' color scheme.

3.5.1.2.3.4 Determine Message Display Capabilities

The DMS shall allow a management station to determine the MULTI tags supported by the DMS.

3.5.1.2.4 Delete All Messages of a Message Type with One Command

The DMS shall allow a management station to delete all messages of a specific message type in one command. Messages types that can be deleted are either 'volatile' messages or 'changeable' messages.

3.5.1.3 Manage Fonts

Requirements for managing the font information are provided in the following subsections.

3.5.1.3.1 Determine Maximum Number of Fonts Supported

The DMS shall allow a management station to determine the maximum number of fonts that can be defined and the number that are defined within the sign controller.

3.5.1.3.2 Determine Maximum Character Size

The DMS shall allow a management station to determine the maximum size (in bytes) that the DMS allows for each character bitmap.

3.5.1.3.3 Determine Maximum Number of Characters per Font

The DMS shall allow a management station to determine the maximum number of characters that the DMS allows for an individual font.

3.5.1.3.4 Retrieve a Font Definition

The DMS shall allow a management station to upload the fonts defined in the sign controller.

3.5.1.3.5 Configure a Font

The DMS shall allow a management station to modify or create a font definition in the sign controller.

> Note: It is recognized that the message display on the sign could be unpredictable during the download of a font. Those specifying authorities or application developers who are sensitive to this issue can blank the display during a font download.

3.5.1.3.6 Delete a Font

The DMS shall allow a management station to delete a font definition in the sign controller.

3.5.1.3.7 Validate a Font

The DMS shall allow a management station to validate any font stored within the controller to ensure that the font specification is as expected and has not been corrupted during download or changed since last use.

3.5.1.4 Manage Graphics

Requirements for managing the storage of graphics in the DMS are provided in the following subsections.

3.5.1.4.1 Determine Number of Graphics

The DMS shall allow a management station to determine the number of graphics defined and the maximum number that can be defined within the sign controller.

3.5.1.4.2 Determine Maximum Graphic Size

The DMS shall allow the management station to identify the maximum size (in bytes) allowed for each graphic.

3.5.1.4.3 Determine Available Graphics Memory

The DMS shall allow the management station to identify the maximum memory available for graphics storage.

3.5.1.4.4 Retrieve a Graphic Definition

The DMS shall allow a management station to retrieve any of the graphics defined in the sign controller.

3.5.1.4.5 Store a Graphic Definition

The DMS shall allow a management station to store a graphic in the sign controller.

3.5.1.4.6 Delete a Graphic

The DMS shall allow a management station to delete a graphic in the sign controller.

3.5.1.4.7 Validate a Graphic

The DMS shall allow a management station to validate any graphic stored within the controller to ensure that the graphic is as expected and has not been corrupted during download or changed since last use.

3.5.1.5 Configure Brightness of Sign

Requirements for configuring the sign controller's internal algorithm to set sign brightness are provided in the following subsections.

3.5.1.5.1 Determine Maximum Number of Light Sensor Levels

The DMS shall allow a management station to determine the number of ambient light detection levels supported by the light sensors.

3.5.1.5.2 Configure Light Output Algorithm

The DMS shall allow a management station to configure the relationships between the detection of ambient light (light sensor input reading) and the brightness level of the sign (light output).

3.5.1.5.3 Determine Current Light Output Algorithm

The DMS shall allow a management station to determine the relationships between the detection of ambient light (light sensor input reading) and the brightness level of the sign (light output).

3.5.1.6 Configure Current Speed Limit

The DMS shall allow a management station to download a current speed limit to the sign controller.

3.5.1.7 Determine Current Speed Limit Configuration

The DMS shall allow a management station to determine the current speed limit as currently configured in the sign controller.

3.5.1.8 Manage Sign Display Location

Requirements for managing the sign location are provided in the following subsections.

3.5.1.8.1 Determine the Sign Display Location

The DMS shall allow a management station to determine the configured location of the sign display and its direction.

3.5.1.8.2 Configure the Sign Display Offset Location

The DMS shall allow a management station to configure the offset of the sign display location as measured from the field device location along with an indiction of the direction of the sign face.

3.5.1.8.3 Determine Sign Display Offset Location

The DMS shall allow a management station to determine the configured offset location of the sign display along with the indiction of the direction of the sign face.

3.5.1.9 Manage Physical Beacons

Requirements for managing the physical beacons are provided in the following subsections.

3.5.1.9.1 Determine Number of Beacons

The DMS shall allow a management station to determine the number of physical beacons supported by the DMS.

3.5.1.9.2 Determine Beacon Properties

The DMS shall allow a management station to determine the properties of the physical beacons, including the physical position, supported flash modes, and supported colors of each beacon.

3.5.1.9.3 Configure Beacon Operational Strategies

The DMS shall allow a management station to configure one or more operational strategies for the physical beacons, where each strategy defines the flash point, flash duration, flash cycle, and color for each beacon.

3.5.1.9.4 Retrieve Beacon Operational Strategies

The DMS shall allow a management station to retrieve the configuration of an operational strategy for the physical beacons, including the flash point, flash duration, flash cycle, and color for each beacon.

3.5.1.9.5 Configure Operation within a Strategy

The DMS shall allow a management station to configure one or more operational strategies for the physical beacons, where each strategy defines the flash point, flash duration, flash cycle, and color for each beacon.

3.5.1.9.6 Retrieve Beacon Operation within a Strategy

The DMS shall allow a management station to retrieve the configuration of an operational strategy for the physical beacons, including the flash point, flash duration, flash cycle, and color for each beacon.

3.5.2 Control the DMS

Requirements for controlling the DMS operation are provided in the following subsections.

3.5.2.1 Manage Control Source

A DMS shall allow the user to switch between the local and central control modes.

> Note: See the corresponding Dialog in Section 4 for further explanations.

3.5.2.2 Control the Sign Face

Requirements for controlling the sign face are provided in the following subsections.

3.5.2.2.1 Activate a Message

The DMS shall allow a management station to display a message on the sign face, including:

a) Any permanent message supported by the sign
b) Any previously defined message
c) A blank message of any run-time priority

d) A message based on the scheduling logic, if a scheduler is supported by the sign.

3.5.2.2.2 Manage Default Message Display Parameters

Requirements for managing default settings for certain message display parameters are provided in the following subsections.

3.5.2.2.2.1 Determine Default Message Display Parameters

The DMS shall allow a management station to determine the current settings for the following message display defaults:

a) Default background and foreground colors
b) Default font
c) Default flash-on and flash-off times
d) Default line justification
e) Default page justification
f) Default page-on and page-off times
g) Default character set

3.5.2.2.2.2 Configure Default Background and Foreground Color

The DMS shall allow a management station to configure the default background and default foreground colors for a message on the sign face to any color supported by the sign (see Supplemental Requirements for Color Scheme).

> Note: Reverse video in monochrome signs may be achieved by setting a color rectangle to the 'on' color and setting the foreground color to 'black'.

3.5.2.2.2.3 Configure Default Flash-On and Flash-Off Times

The DMS shall allow a management station to configure the default on-time and default off-time for flashing text or graphics. The specification will identify the range of values that the DMS shall support. If the specification does not indicate the ranges for the flashing rates, the DMS shall at least support all on and off values ranging from 0.0 seconds to 10.0 seconds in 0.5 second increments, inclusive.

3.5.2.2.2.4 Configure Default Font

The DMS shall allow a management station to select any font supported by the sign and configure it as the default font for displaying text.

3.5.2.2.2.5 Configure Default Line Justification

The DMS shall allow a management station to configure the default justification for a line to any type of justification supported by the DMS. The specification will identify the types of line justification that the DMS shall support. If the specification does not indicate the types of justification, the DMS shall support at least left justified.

3.5.2.2.2.6 Configure Default Page Justification

The DMS shall allow a management station to configure the default vertical justification for displaying a page of text on the sign face (e.g., at the top of the sign, in the middle, or at the bottom) to any type of justification supported by the DMS. The specification will identify the types of page justification that the DMS shall support. If the specification does not indicate the types of justification, the DMS shall support at least top justified.

3.5.2.2.2.7 Configure Default Page On-Time and Page Off-Time

The DMS shall allow a management station to configure the default time to display each page of a multi-page message and the default time to blank the sign face between the display of each page of the message. The specification will identify the range of values that the DMS shall support. If the specification does not indicate the ranges for the page times, the DMS shall at least support all page-on and page-off values ranging from 0.0 seconds to 10.0 seconds in 0.5 second increments, inclusive.

3.5.2.2.2.8 Configure Default Character Set
The DMS shall allow a management station to configure the default character set to be used when displaying a message (e.g., ASCII versus UTF-8) to any character set supported by the DMS.

3.5.2.2.3 Manage Message Library
Requirements for managing the contents of a message library are provided in the following subsections.

3.5.2.2.3.1 Determine Available Message Types
The DMS shall allow a management station to determine information about the different message storage memory types available within the sign controller. The different types are:

a) Permanent memory (content cannot be edited and will not be lost upon power failure)
b) Volatile memory (content is editable but will be lost upon power failure)
c) Changeable memory (content is editable but will not be lost upon power failure)

3.5.2.2.3.2 Determine Available Message Space
The DMS shall allow a management station to determine the number of messages that are currently stored and the remaining space within the controller's message library.

3.5.2.2.3.3 Define a Message
The DMS shall allow a management station to download a message for storage in the sign controller's message library.

3.5.2.2.3.4 Verify Message Contents
The DMS shall allow a management station to quickly verify that the contents of a message are as expected through the use of a relatively unique code.

3.5.2.2.3.5 Retrieve Message
The DMS shall allow a management station to upload any message definition from the sign controller.

3.5.2.2.4 Define Messages for Internal Activation
Requirements for managing the contents of a schedule to display one or more permanent or previously defined messages are provided in the following subsections.

3.5.2.2.4.1 Retrieve a DMS Message Action
The DMS shall allow a management station to retrieve a message that can be activated by an internal event through the ISO26048-1-Action feature.

NOTE—The ISO26048-1-Action feature can be called by conditional triggers (ISO26048-1-CondTrigger), day plan schedule (ISO26048-1-DayPlan), or scheduler (ISO26048-1-SchedTrigger).

NOTE—Prior to v04, this heading was limited to retrieving the DMS schedule and message(s) called.

3.5.2.2.4.2 Configure a DMS Message Action
The DMS shall allow a management station to define the message(s) to be displayed by the ISO26048-1-Action feature.

NOTE—The ISO26048-1-Action feature can be called by conditional triggers (ISO26048-1-CondTrigger), day plan schedule (ISO26048-1-DayPlan), or scheduler (ISO26048-1-SchedTrigger).

NOTE—Prior to v04, this heading was limited to configuring the DMS schedule and message(s) called.

3.5.2.2.5 Configure Event-Based Message Activation
Requirements for configuring the controller to activate a message (including blank or schedule) in response to certain internal events are provided in the following subsections.

3.5.2.2.5.1 Configure Messages Activated by Standardized Events

Requirements for configuring the message to be activated in response to various standardized internal events are provided in the following subsections.

3.5.2.2.5.1.1 Configure Message for Short Power Loss Recovery Event

The DMS shall allow a management station to define which message to display upon recovery from a short power loss.

3.5.2.2.5.1.2 Configure Message for Long Power Loss Recovery Event

The DMS shall allow a management station to define which message to display upon recovery from a long power loss.

3.5.2.2.5.1.3 Configure Message for Power Loss Event

The DMS shall allow a management station to define which message to display upon a loss of power.

> Note: This feature is not applicable to certain DMS technologies that require constant power to display messages such as pure LED, pure fiber optics, or bulb technologies.

3.5.2.2.5.1.4 Configure Message for Controller Reset Event

The DMS shall allow a management station to define which message to display upon the DMS controller being reset.

3.5.2.2.5.1.5 Configure Message for Communications Loss Event

The DMS shall allow a management station to define which message to display upon the detection of a loss of communications to the management station.

3.5.2.2.5.1.6 Configure Message for End Message Display Duration Event

The DMS shall allow a management station to define which message to display upon the expiration of the message display duration.

> Note: Every message is associated with a duration when it is activated, which may be infinite. If the duration expires, the message referenced by this configuration parameter defines the message to display next.

3.5.2.2.6 Activate a Message with Status

The DMS shall adhere to requirement 3.5.2.2.1 "Activate a Message". The DMS shall provide status of any message activation for slow activating message signs such as drum signs.

3.5.2.2.7 Extend Message Duration

The DMS shall allow a management station to extend the duration of the currently displayed message.

3.5.2.3 Control Sign Brightness

Requirements for controlling the brightness of the message on the sign face are provided in the following subsections.

3.5.2.3.1 Determine Number of Brightness Levels

The DMS shall allow a management station to determine the maximum number of (settable) brightness levels.

3.5.2.3.2 Determine Current Composite Photocell Value

The DMS shall allow a management station to determine the current composite photocell value as determined by the sign controller based on one or more photocell readings. The composite value is intended to scale to reflect how bright the sign needs to be, which can be affected by the amount of ambient light and the direction of the light (e.g., overhead sunlight versus glare from the sun being directly behind the sign).

3.5.2.3.3 Manually Direct-Control Brightness

The DMS shall allow a management station to manually control the light output of the display by selecting any of the brightness levels supported by the DMS.

3.5.2.3.4 Manually Index-Control Brightness

The DMS shall allow a management station to manually control the light output of the display by selecting any of the brightness levels defined within the brightness table.

> Note: The difference between these two manual modes ('manual direct-control' and 'manual index-control') is that a DMS might support 200 different brightness levels but only has three defined within the brightness table. For these three brightness levels, thresholds to switch from one level to another are defined in the brightness table; however, the DMS offers the possibility to define up to 200 brightness levels within the brightness table.

> Note: The previously available control mode 'manual' has been retired to address an ambiguity within NTCIP 1203:1997 and its amendment (2001). Instead the above two manual modes have been introduced to address this ambiguity. See Annex D for further information regarding this change from v1 to v2 of NTCIP 1203.

3.5.2.3.5 Switch Brightness Control Modes

The DMS shall allow a management station to switch between the defined brightness control modes.

> Note: See Section 3.6.2 for Supplemental Requirements related to brightness control modes.

3.5.2.4 Pixel Service

Requirements for pixel service are provided in the following subsections.

3.5.2.4.1 Manage the Exercise of Pixels

The DMS shall allow a management station to manage frequency and duration of the exercise of each pixel's physical actuation mechanism.

3.5.2.4.2 Determine Pixel Service Settings

The DMS shall allow a management station to determine the current settihngs for the frequency and duration of pixel service.

3.5.3 Monitor the Status of the DMS

Requirements for monitoring the status of the DMS are provided in the following subsections.

3.5.3.1 Perform Diagnostics

Requirements for performing diagnostic functions on the DMS are provided in the following subsections.

3.5.3.1.1 Provide General DMS Error Status Information

The DMS shall allow a management station to retrieve a high-level overview of the operational status of the DMS that includes an indication of the following error and warning conditions:

a) Communications Error
b) Power Error
c) Attached Device Error, if any attached devices are present
d) Lamp Error, if lamp technology is used
e) Pixel Error, if a pixel matrix is used
f) Light Sensor Error, if light sensors are present
g) Message Error
h) Controller Error
i) Temperature Warning, if temperature sensors are present in the sign housing or controller cabinet

j) Climate-Control System Error, if there is a climate control system
k) Critical Temperature Error, if temperature sensors are present in the sign housing or controller cabinet
l) Drum Sign Error, if drum technology is used
m) Open Door Warning, if door sensors are present
n) Humidity Warning, if humidity sensors are present
o) Any other errors that the manufacturer believes is important to report via a text-based object

3.5.3.1.2 Monitor Pixels

Requirements for monitoring pixels are provided in the following subsections.

3.5.3.1.2.1 Activate Pixel Testing

The DMS shall allow a management station to initiate a pixel test.

3.5.3.1.2.2 Monitor Pixel Error Count

The DMS shall allow a management station to determine the number of pixel errors that have been detected.

3.5.3.1.2.3 Monitor Pixel Error Map

The DMS shall allow a management station to determine the status of each pixel (not failed/stuck on/stuck off).

The DMS shall be accompanied with documentation that maps each individual bit to a specific pixel.

3.5.3.1.2.4 Monitor Pixel Error Details

The DMS shall allow a management station to determine the detailed information for any pixels that are not operational, including:

a) Horizontal location of the pixel
b) Vertical location of the pixel
c) The type of failure (stuck on/off, color error, electrical error, mechanical error, error affecting some/all strings of the pixel)

3.5.3.1.3 Monitor Lamp Error Details

The DMS shall allow a management station to obtain detailed information for any failed lamp, including:

a) Location of the topmost row of pixels served by the lamp
b) Location of the leftmost column of pixels served by the lamp
c) Location of the bottommost row of pixels served by the lamp
d) Location of the rightmost column of pixels served by the lamp

3.5.3.1.4 Monitor Message Activation Error Details

The DMS shall allow a management station to obtain detailed information regarding the success or failure of the last message activation, including details related to any message content errors. This information may be overwritten by other actions in the device, but there shall be a way to verify that the error details still apply to the last activation command.

3.5.3.1.5 Monitor the Sign's Control Source

The DMS shall allow a management station to determine the current control source for the DMS. See Supplemental Requirements for Control Modes for a description of the possible control modes.

3.5.3.1.6 Determine Critical Temperature Threshold

The DMS shall allow a management station to determine the manufacturer's critical enclosure temperature, which if exceeded, shall generate a critical temperature alarm and cause the sign to turn off. Likewise, the DMS shall allow a management station to determine the temperature at which the sign will turn back on after such a critical temperature alarm.

3.5.3.2 Monitor the Current Message

Requirements for monitoring the information about the currently displayed message and related parameters are provided in the following subsections.

3.5.3.2.1 Monitor Information about the Currently Displayed Message

The DMS shall allow a management station to monitor details about the current message, including:

a) The message content
b) The stored message number used to activate the current message
c) The message display time remaining
d) The process or management station that activated the message
e) The current brightness level of the message, if brightness is supported by the DMS
f) The status of the physical beacons, if present
g) The status of pixel service, if supported by the DMS

3.5.3.2.2 Monitor Dynamic Field Values

The DMS shall allow a management station to monitor the value(s) currently being displayed within the dynamic fields of the current message.

3.5.3.3 Monitor Status of DMS Control Functions

Requirements for monitoring the status of the various control functions are provided in the following subsections.

3.5.3.3.1 Monitor a Callable Message

The DMS shall allow a management station to determine which messages are currently configured to be displayed in response to a call from the ISO 26048-1 action table.

3.5.3.3.2 Monitor Short Power Recovery Message

The DMS shall allow a management station to determine which message is currently configured to be displayed in response to a power recovery event after a short power loss.

3.5.3.3.3 Monitor Long Power Recovery Message

The DMS shall allow a management station to determine which message is currently configured to be displayed in response to a power recovery event after a long power loss.

3.5.3.3.4 Monitor Power Loss Message

The DMS shall allow a management station to determine which message is currently configured to be displayed during a power loss.

3.5.3.3.5 Monitor Reset Message

The DMS shall allow a management station to determine which message is currently configured to be displayed in response to a software or hardware reset event.

3.5.3.3.6 Monitor Communications Loss Message

The DMS shall allow a management station to determine which message is currently configured to be displayed if communications with the management station are lost for a user-defined period of time. Detection of loss of communications shall be disabled when the DMS is in 'local' control mode.

3.5.3.3.7 Monitor End Duration Message

The DMS shall allow a management station to determine which message is currently configured to be displayed upon the termination of the current message duration.

3.5.3.3.8 Retrieve a DMS Message Action

The DMS shall allow a management station to determine which messages are currently configured to be displayed in response to a scheduled action.

3.5.3.3.9 Determine Number of DMS Message Actions

The DMS shall allow a management station to determine the number of scheduled messages that can be stored.

3.5.3.3.10 Determine Number of Callable Messages

The DMS shall allow a management station to determine the number of callable messages currently defined.

3.5.4 Providing for Multi-Version Interoperability

Any requirements for providing backwards compatibility with prior versions of NTCIP 1203 should be defined by using the PRL contained in NTCIP 1203 v03.

> Note: NTCIP 1203 versions prior to v04 were intended for deployment using SNMPv1, which does not provide adequate security. Support for SNMPv1 is not recommended.

3.6 Supplemental Non-Communications Requirements

Supplemental requirements for the DMS are provided in the following subsections. These requirements do not directly involve communications between the management station and the DMS, but, if the supplemental requirement is selected in the PRL, the DMS must perform the stated functionality to claim conformance to this document.

3.6.1 Supplemental Requirements for Fonts

Supplemental requirements for character set support are provided in the following subsections.

3.6.1.1 Support for a Number of Fonts

The DMS shall support the number of fonts as defined by the specification. If the specification does not define the number of fonts, the DMS shall support at least one font.

3.6.2 Supplemental Requirements for General Illumination Brightness

Supplemental requirements for general illumination brightness support are provided in the following subsections.

3.6.2.1 Support a Number of Brightness Levels

The DMS shall support the number of brightness levels as specified in the specification. If the specification does not define the number of brightness levels, the DMS shall support at least 1 brightness level.

3.6.3 Supplemental Requirements for Automatic Brightness Control

Supplemental requirements for automatically adjusting the brightness of a message are provided in the following subsections.

3.6.3.1 Automatically Control Brightness

The DMS shall automatically manage the light sensor-driven light output of the display when this mode is enabled.

3.6.3.2 Inhibit Flickering of Message Brightness

The DMS shall allow the Light Output Algorithm to include overlapping values, which shall enable the Light Output Algorithm to avoid flickering of the light output due to small changes in the measured ambient light conditions. If this feature is not supported, the DMS shall return a wrongValue error whenever the dmsIllumBrightnessValues object is set to a value that includes overlapping brightness ranges.

3.6.3.3 Support a Number of Light Sensor Levels

The DMS shall support the number of light sensor levels as specified in the specification. If the specification does not define the number of light sensor levels, the DMS shall support at least 3 light sensor levels.

3.6.3.4 Constraints on Manufacturer Value

All instances of fdSrsaPortMaxValue.FAL shall have the same value and shall reflect the maximum possible value of fdSrsaTypeValueStatsMfrValue.

The value of fdSrsaTypeValueStatsMfrValue shall indicate the overall level of ambient light as a value ranging from 0 (darkest) to the value of fdSrsaPortMaxValue.FAL (brightest), based on the readings of available photocells. The fdSrsaTypeValueStatsMfrValue object shall provide a virtual photocell level in that it may be algorithmically determined from one or more photocells and is the value used for calculations dealing with the brightness table. The algorithm used to determine the virtual level from the actual photocell readings is manufacturer specific to accommodate various hardware needs. .

3.6.4 Supplemental Requirements for Control Modes

Supplemental requirements for allowing different entities to control the DMS are provided in the following subsections.

3.6.4.1 Support Central Control Mode

A DMS shall allow an operator to control the sign from a remote location (e.g., from central).

3.6.4.2 Support Local Control Mode

The DMS shall allow an operator to control the sign through a local interface.

Note: A 'local' interface may include any of the following: a touch panel on the sign controller, a laptop connected directly to a 'local' port on the sign controller, or any other mounted or non-mounted panel that can be used to select a message for display.

3.6.4.3 Support Central Override Control Mode

The DMS shall allow the central system to override the local control mode.

> Note: An implementation may preclude the use of the "central override" mode, if it would pose a safety risk.

3.6.4.4 Processing Requests from Multiple Sources

The DMS shall only allow a single source to control the sign at any one time.

3.6.5 Supplemental Requirements for Message Activation Request

Supplemental requirements for activating a message for display on the sign face based on an external request are provided in the following subsections.

3.6.5.1 Supplemental Requirements for Message Activation

Supplemental requirements for activating a message for display on the sign face are provided in the following subsections.

3.6.5.1.1 Activate Any Message

The DMS shall allow the activation of any valid message that is stored in the sign controller.

3.6.5.1.2 Preserve Message Integrity

The DMS shall prohibit the display of a message that uses memory objects such as fonts or graphics that were altered after the message was composed and saved within the sign's local message library.

3.6.5.1.3 Ensure Proper Message Content

The DMS shall ensure that the contents of the message are the same as what the requester requests.

3.6.5.2 Indicate Message Display Duration

Each message activation shall be associated with a display duration for the sign controller to display the message. If the request is validated, the DMS shall display the associated message for the indicated duration.

3.6.5.3 Indicate Message Display Requester ID

Each message activation shall be associated with an indication of the entity that requested the display. The DMS shall store this information while the message is displayed.

3.6.5.4 Supplemental Requirements for Message Activation Priority

The DMS shall only activate the newly requested message if the activation priority is higher than the run-time priority of the currently displayed message.

3.6.6 Supplemental Requirements for Message Definition

Supplemental requirements for defining user-defined messages (e.g., volatile and changeable messages) are provided in the following subsections.

3.6.6.1 Identify Message to Define

Each message stored in the sign controller shall be associated with a unique identifier.

3.6.6.2 Define Message Content

Supplemental requirements for defining the message content are provided in the following subsections.

3.6.6.2.1 Support Multi-Page Messages

The DMS shall allow the message to contain the number of distinct page displays as defined by the specification. If the specification does not define the number of distinct page displays that must be supported, the DMS shall support at least one page per message.

3.6.6.2.2 Support Page Justification

The DMS shall allow the message content to specify all modes of vertical (page) justification supported by the sign (see Section 3.5.2.2.2.6). Supplemental requirements for supporting vertical justification of the message on the display are provided in the following subsections.

3.6.6.2.2.1 Support for One Page Justification within a Message

The DMS shall allow the message content to specify a single vertical (page) justification, which shall apply to all pages of the message.

3.6.6.2.2.2 Support for Multiple Page Justifications within a Message

The DMS shall allow the message content to specify vertical (page) justification on a page-by-page basis.

3.6.6.2.3 Support Multiple Line Messages

The DMS shall allow each page of the message to contain up to the number of lines as defined by the specification. If the specification does not define the number of lines that must be supported, the DMS shall support at least one line per page.

3.6.6.2.4 Support Line Justification

The DMS shall allow the message content to specify all modes of horizontal (line) justification supported by the sign (see Section 3.5.2.2.2.5). Supplemental requirements for horizontal (line) justification are provided in the following subsections.

3.6.6.2.4.1 Support for a Single Line Justification within a Message

The DMS shall allow the message content to specify a single line justification, which shall be used for each line within the message.

3.6.6.2.4.2 Support Line Justification on a Page-by-Page Basis

The DMS shall allow the message content to specify the line justification on a page-by-page basis.

3.6.6.2.4.3 Support Line Justification on a Line-by-Line Basis

The DMS shall allow the message content to specify the line justification on a line-by-line basis.

3.6.6.2.5 Support Color

The DMS shall allow the message content to specify any color supported by the sign (see Section 3.6.8). Supplemental requirements for foreground and background color commands within a message are provided in the following subsections.

3.6.6.2.5.1 Support a Single Color Combination per Message

The DMS shall allow the message content to specify a single foreground color and a single background color, both of which shall apply to the entire message.

3.6.6.2.5.2 Support a Color Combination for each Page

The DMS shall allow the message content to specify the foreground color and background color on a page-by-page basis.

3.6.6.2.5.3 Support a Color Combination for each Character within a Message

The DMS shall allow the message content to specify the foreground color and background color on a character-by-character basis.

3.6.6.2.5.4 Color Rectangle

The DMS shall allow the message content to specify an area of the sign to display a selected color.

3.6.6.2.6 Support Font Commands

The DMS shall allow the message content to specify any font supported by the sign (see Section 3.5.2.2.2.4). Supplemental requirements for supporting font commands within a message are provided in the following subsections.

Note: For an example of a font, see NEMA TS 4.

3.6.6.2.6.1 Support One Font within a Message

The DMS shall allow the message content to specify a single font, which shall apply to the entire message.

3.6.6.2.6.2 Support One Font per Page within a Message

A DMS shall allow the message content to specify the font on a page-by-page basis.

3.6.6.2.6.3 Support Character by Character Selection of Fonts within a Message

A DMS shall allow the message content to specify the font on a character-by-character basis.

3.6.6.2.7 Support Moving Text

The DMS shall allow the message content to include a 'window' that contains moving text at a defined speed and direction. If this function is supported, all of the configurable parameters of this function shall be fully supported.

3.6.6.2.8 Support Character Spacing

The DMS shall allow the message content to specify the spacing between characters in a text string or between text and a graphic on a character-by-character basis. If this function is supported, all of the configurable parameters of this function shall be fully supported.

3.6.6.2.9 Support Customizable Page Display Times in a Message

The DMS shall allow the message content to specify the time to display each page and the time to blank the sign face between each page when displaying a multi-page message. The allowed range for the display time and the blank time shall be identical to the range identified in the specification for Section 3.4.2.3.2.7.

3.6.6.2.10 Support Flashing

Supplemental requirements for flashing text are provided in the following subsections.

3.6.6.2.10.1 Support Character-by-Character Flashing

The DMS shall allow the message content to identify portions of text (and/or graphics) to be flashed on a character-by-character basis.

3.6.6.2.10.2 Support Line-by-Line Flashing

The DMS shall allow the message content to identify portions of text (and/or graphics) to be flashed on a line-by-line basis.

3.6.6.2.10.3 Support Page-by-Page Flashing

The DMS shall allow the message content to identify portions of text (and/or graphics) to be flashed on a page-by-page basis.

3.6.6.2.11 Support Customizable Flashing Times within a Message

The DMS shall allow the message content to specify the time to display and the time to blank each section of flashing text. The allowed range for the display time and the blank time shall be identical to the range identified in the specification for Section 3.5.2.2.2.3.

3.6.6.2.12 Support Hexadecimal Character

The DMS shall allow the message content to specify the display of character numbers greater than 255 (0xFF).

> Note: This allows the display of non-Latin-based characters using their standardized UNICODE values, assuming that support for these characters have also been specified by the Supplemental Requirements for Character Sets.

3.6.6.2.13 Support Message Data Fields

Supplemental requirements for defining a message that includes fields that display dynamic data are provided in the following subsections.

3.6.6.2.13.1 Support Current Time Field without AM/PM Field

The DMS shall allow the message content to include field(s) indicating the current time in either 12-hour or 24-hour format, selectable by the user. The 12-hour format shall not include any AM/PM indicator.

3.6.6.2.13.2 Support Current Time with Uppercase AM/PM Field

The DMS shall allow the message content to include field(s) indicating the current time with uppercase AM/PM indicated after the time value.

3.6.6.2.13.3 Support Current Time with Lowercase AM/PM Field

The DMS shall allow the message content to include field(s) indicating the current time with lowercase am/pm indicated after the time value.

3.6.6.2.13.4 Support Current Temperature Field

A DMS shall allow the message content to include field(s) indicating the current ambient air temperature in either Fahrenheit or Celsius, selectable by the user, and using either 2 or 3 character fields, also selectable by the user.

3.6.6.2.13.5 Support Detected Vehicle Speed Field

The DMS shall allow the message content to include field(s) indicating the current travel speed of the traffic in either miles-per-hour or kilometer-per-hour, selectable by the user, and using either 2 or 3 character fields, also selectable by the user.

3.6.6.2.13.6 Support Current Day of Week Field

The DMS shall allow the message content to include field(s) indicating the current day of the week in a 3-

character format such as SUN, MON, TUE, etc.

3.6.6.2.13.7 Support Current Day of Month Field

The DMS shall allow the message content to include field(s) indicating the current date of the month.

3.6.6.2.13.8 Support Current Month of Year Field

The DMS shall allow the message content to include field(s) indicating the current month of the year.

3.6.6.2.13.9 Support Current Year Field

The DMS shall allow the message content to include field(s) indicating the current year.

3.6.6.2.13.10 Support User-Definable Field

The DMS shall allow the message content to include field(s) indicating user-definable parameters.

> Note: For interoperability reasons, it is not recommended to require this function.

3.6.6.2.13.11 Data Field Refresh Rate

The DMS shall update each field at a refresh rate as defined in the specification. If the specification does not indicate the refresh rate, the DMS shall update the fields at least every 60 seconds.

> Note: An operator or user of a DMS may want to display information based on data received from a device that has a direct interface with the DMS Controller. This is accomplished via fields within the displayed message, where the fields within the message being displayed change based on the data (typically real-time) from the other device. The device could be a clock calendar, a weather station, a speed station, etc. Fields can be defined as time, date, year, day of week, temperature, or speed.

3.6.6.2.14 Support of Graphics

The DMS shall allow the message content to include zero or more graphic(s) at any location of the face of the display.

3.6.6.2.15 Specify Location of Message Display

A DMS shall allow the message content to specify the starting position of text and graphics on the sign face at a one-pixel resolution.

3.6.6.2.16 Support of Text

Supplemental requirements for including text characters in a message are provided in the following subsections.

3.6.6.2.16.1 Support of Textual Content

The DMS shall allow the message content to include any character supported by the DMS in any order, unless otherwise restricted by the specification.

3.6.6.2.16.2 Support of Message Lengths Compatible with Sign Face

The DMS shall allow the message to contain any number of characters per page for each page, up to the physical limits of the sign face.

3.6.6.2.17 Support of Manufacturer Specific Message Definitions

The DMS shall support manufacturer-specific tags.

> Note: For interoperability reasons, it is not recommended to require this function.

3.6.6.3 Identify Message Owner

Each message stored in the sign controller shall be associated with an owner name.

3.6.6.4 Priority to Maintain a Message

Each message stored in the sign controller shall be associated with a run-time priority.

3.6.6.5 Physical Beacon Activation Flag

Each message stored in a sign controller library shall indicate whether any existing attached physical beacons are to flash while this message is displayed.

3.6.6.6 Pixel Service Flag

Each message stored in a sign controller library shall indicate whether a pixel service can be executed while the message is displayed.

3.6.6.7 Message Status

Each message stored in the sign controller shall be associated with a status to indicate if it is valid for display, being modified, etc.

> Note: See Section 4.3 for state transition details.

3.6.7 Supplemental Requirements for Locally Stored Messages

Supplemental requirements for storing local messages are provided in the following subsections.

3.6.7.1 Support Permanent Messages

The DMS shall support the permanent message(s) as defined by the specification. If the procurement specification does not define the permanent messages, the DMS shall support at least one permanent message that can be used for testing the sign operation.

> Note: A procurement specification should specify the minimum number of permanent messages that the DMS is required to support and their details (e.g., identification number, MULTI string including MULTI tags, physical beacon status, etc.).

> Note: Refer to Section 1.4 for the definition of Permanent Messages.

3.6.7.2 Support Changeable Messages

The DMS shall support the number of changeable messages and amount of changeable memory as defined by the specification. If the specification does not define the number of changeable messages, the DMS shall support at least one changeable message. If the specification does not define the amount of changeable memory, the DMS shall support an amount of changeable memory that is at least the product of the number of messages multiplied by 100 bytes.

> Note: Refer to Section 1.4 for the definition of Changeable Messages.

3.6.7.3 Support Volatile Messages

The DMS shall support the number of volatile messages and amount of volatile memory as defined by the specification. If the specification does not define the number of volatile messages, the DMS shall support at least one volatile message. If the specification does not define the amount of volatile memory, the DMS shall support an amount of volatile memory that is at least the product of the number of volatile messages multiplied by 100 bytes.

Unless otherwise specified in a specification, the DMS may fulfill the requirements of this section by providing additional changeable messages and additional changeable memory. If the DMS implements this option, the total number of changeable messages supported by the DMS shall be at least the sum of the required changeable messages and the required volatile messages; likewise, the total changeable memory supported by the DMS shall be at least the sum of the required changeable memory and the required volatile memory.

> Note: Refer to Section 1.4 for the definition of Volatile Messages.

3.6.8 Supplemental Requirements for Color Scheme

Supplemental requirements for supporting color are provided in the following subsections.

3.6.8.1 Support 256 Shades Scheme

The DMS shall support the Monochrome 8 Bit color scheme where each pixel can be defined using a gray-scale palette with 256 shades ranging form 0 (off) to 255 (full intensity).

3.6.8.2 Support Classic NTCIP Color Scheme

The DMS shall support the Classic NTCIP color scheme (for single-intensity multi-color signs): The defined colors are:

a) black
b) red
c) yellow
d) green
e) cyan
f) blue
g) magenta
h) white
i) orange
j) amber

3.6.8.3 Support 24-Bit Color Scheme

The DMS shall support the Color 24 Bit color scheme where each pixel can be defined by three bytes, one for each red, green and blue.

3.6.8.4 Support Single Color

The sign face shall support black (or off) and at least one other color.

3.6.9 Supplemental Requirements for Monitoring DMS Subsystems

Supplemental requirements for monitoring DMS subsystems are provided in the following subsections.

3.6.9.1 Supplemental Requirements for Photocells

The photocell feature indicates the status of each photocell (i.e., illuminance sensor) that detect the current amount of light. Different photocells are often designed to face different directions and therefore can report very different values. The algorithm for transforming the individual sensor readings into the summary dmsIllumPhotocellLevelStatus object is manufacturer specific.

3.6.9.1.1 Photocell Documentation

The DMS shall be accompanied with documentation that clearly identifies the location of each photocell listed in the SRSA table.

3.6.9.1.2 Photocells Monitored through SRSA

For each photocell component of the field device, the field device shall provide an entry in the SRSA table where fdSRSAType equals "DIS".

3.6.9.2 Supplemental Requirements for Lamp Status

The lamp status feature indicates the status of the lamps that illuminate pixels on the sign face. Lamps are only used for some sign display technologies.

3.6.9.2.1 Lamp Documentation

The DMS shall be accompanied with documentation that clearly identifies the location of each lamp listed in the SRSA table.

3.6.9.2.2 Lamp Status Monitored through SRSA

For each lamp component of the field device, the field device shall provide an entry in the SRSA table where fdSRSAType equals "DLS".

NOTE—Per ISO 26048-1, lamp status rows are required to support the fdSrsaTypeTestTable, which provides for the manual activation of tests.

3.6.9.3 Supplemental Requirements for Sign Drum

The sign drum feature indicates the status of the sign drum (for drum signs only).

3.6.9.3.1 Sign Drum Documentation

The DMS shall be accompanied with documentation that clearly identifies the location of each drum listed in the SRSA table.

3.6.9.3.2 Sign Drums Monitored through SRSA

For each sign drum component of the field device, the field device shall provide an entry in the SRSA table where fdSRSAType equals "DDR".

3.6.9.4 Supplemental Requirements for Speed Detectors

The speed detector feature allows the field device to report the speed of passing vehicles.

3.6.9.4.1 Speed Detector Documentation

The DMS shall be accompanied with documentation that clearly identifies the location of each speed detector listed in the SRSA table.

3.6.9.4.2 Speed Detectors Monitored through SRSA

For each speed detector component of the field device, the field device shall provide an entry in the SRSA table where fdSRSAType equals "FSD".

3.6.10 Supplemental Requirements for Scheduling

Supplemental requirements for defining a time-based schedule are provided in the following subsections.

3.6.10.1 Support a Number of DMS Actions

The DMS shall support the number of DMS actions as defined in the specification. If the specification does not define the number of DMS actions, the DMS shall support at least two DMS actions.

> Note: A DMS action is defined as being a unique command to activate display a message that can be called programmatically (e.g., by the fdActionTable). For example, displaying changeable message number 1 would be one action, displaying changeable message number 2 would be a second action and blanking the sign would be a third action.

3.6.10.2 Support Callable DMS Actions

The DMS shall allow the action feature (ISO 26048-1) to be configured to call the DMS action table (dmsActionTable) to activate any message supported by the DMS and currently valid within the message table.

3.6.10.3 Implement DMS Action When Called

The Activate Message action shall change the state of the scheduled message buffer and shall only cause the display of the message if the current message is the Scheduler.

3.6.11 Supplemental Requirements for Graphics

Supplemental requirements for defining graphics are provided in the following subsections.

3.6.11.1 Support for a Number of Graphics

The DMS shall support the number of graphics as defined by the specification. If the specification does not define the number of graphics, the DMS shall support at least one graphic.

3.6.11.2 Support for Graphic Memory

The DMS shall support the number of bytes of graphic memory as defined in the specification. If the specification does not define the amount of graphic memory, the DMS shall support at least one kilobyte of graphic memory.

3.6.12 Supplemental Requirements for Page Justification

Supplemental requirements for page justification are provided in the following subsections.

3.6.12.1 Support Top Page Justification

The DMS shall support top page justification.

3.6.12.2 Support Middle Page Justification

The DMS shall support middle page justification.

3.6.12.3 Support Bottom Page Justification

The DMS shall support bottom page justification.

3.6.13 Supplemental Requirements for Line Justification

3.6.13.1 Support Left Line Justification

The DMS shall support left line justification.

3.6.13.2 Support Center Line Justification

The DMS shall support center line justification.

3.6.13.3 Support Right Line Justification

The DMS shall support right line justification.

3.6.13.4 Support Full Line Justification

The DMS shall support full line justification.

3.6.14 Reporting Controller-to-Display Interface Errors

The DMS shall report any controller-to-display interface error using the sensorActuatorError bit (i.e., Bit 5) within the fdControllerStatus object.

3.6.15 Critical Temperature Trigger

The DMS shall support the conditional trigger feature (ISO 26048-1, 8.4) with a permanent entry in the conditional trigger table for each context (i.e., sign display) supported by the field device as follows:

a) Owner = 255
b) Index = 1
c) Description = "Message display critical temperature threshold"
d) Mode = hysteresis (5)
e) SampleType = current
f) Value = <temperature at which the controller shall deactivate the sign display>
g) Value2 = <temperature at which the controller shall reactivate the sign display>
h) Object = <one of fdSrsaPortValue.FET.x, where x is a specific sensor that is monitored, fdSrsaTypeValueStatsMax.FET, fdSrsaTypeValueStatsMean.FET, or fdSrsaTypeValueStatsMfrValue.FET>
i) Wildcard = false (2)
j) ObjectTarget = ""
k) Context = <one for each entry supported by the device>
l) Frequency = <per manufacturer recommendation>
m) Duration = <per manufacturer recommendation>
n) Startup = <per manufacturer recommendation>
o) Startup2 = <per manufacturer recommendation>
p) ActionGroup = <the deactivate action group that satisfies requirement 3.6.16>
q) ActionGroup2 = < the reactivate action group that satisfies requirement 3.6.17>
r) StorageType = permanent (4)

3.6.16 Deactivate Action Group

The DMS shall support the action feature (ISO 26048-1, 8.1) with a permanent entry in the action group table that points to dmsCallableSignDisplayDeactivate. The action group shall allow a management station to create additional actions within the action group (e.g., to create a log entry when the trigger is fired).

3.6.17 Reactivate Action Group

The DMS shall support the action feature (ISO 26048-1, 8.1) with a permanent entry in the action group table that points to dmsCallableSignDisplayReactivate. The action group shall allow a management station to create additional actions within the action group (e.g., to create a log entry when the trigger is fired).

3.7 Security Requirements

3.7.1 Conformant Security

To claim "conformance" with this document, a DMS shall only support protocols that:

a) are sent over TLS 1.3 or a more secure transport layer as recognized by the Cybersecurity and Infrastructure Security Agency (CISA) and
b) provide access control to data and operations using authenticated security crdentials.

Implementations should support the communications stack defined in NTCIP 2301 v03 for the exchanged of data defined by this document.

NOTE—This excludes support for less secure protocols, such as SNMP versions prior to SNMPv3. In other words, conformant devices do not have the installed code to activate prior versions of SNMP.

3.7.2 Consistent Security

To claim "consistency" with this document, a DMS shall:

a) Support protocols providing equivalent security to SNMPv3 per the rules of NTCIP 2301 v03,
b) Support less secure protocols (e.g., prior versions of SNMP), and
c) Allow users to disable all protocols that are less secure than SNMPv3 according to NTCIP 2301 v03.

Implementations should support the communications stack defined in NTCIP 2301 v03 for the exchanged of data defined by this document.

NOTE—Consistent devices are defined to allow agencies to start procuring and deploying SNMPv3-ready devices immediately, before their management station has been updated to support SNMPv3.

Section 4
Dialogs [Normative]

This section is intended for product developers such as manufacturers and system integrators. Other parties might find this section and the following two sections (Object Definitions and MULTI Definitions) helpful to gain a full understanding of design details.

This section presents the standardized dialogs (i.e., sequence of data exchanges) that fulfill various requirements. As SNMP communications are largely driven by the management station, most of the requirements define how the device responds to various possible actions a management station might take.

The NTCIP standards effort is based on SNMP. This protocol offers a high degree of flexibility as to how the management station structures its requests. For example, with SNMP, the management station can do any of the following:

a) Send only those requests that are critical at the current time, whereas a standardized dialog typically sends requests relating to all associated data, regardless of whether it is critical for current purposes.
b) Combine a number of requests in a single packet, whereas a standardized dialog dictates the exact contents of each packet.
c) Separate a group of requests into multiple packets, whereas a standardized dialog dictates the exact contents of each packet.
d) Interweave requests from multiple dialogs, whereas a standardized dialog dictates the exact ordering of messages, which are not interrupted with other messages.

This flexibility can be a powerful tool allowing a management station to optimize the use of communication facilities, which is the primary reason that SNMP was chosen as the core NTCIP protocol. However, the flexibility also means that there are numerous allowable variations in the management process that a management station may choose to use.

Unfortunately, this flexibility presents a challenge to ensuring interoperability. While a conformant DMS is required to support any allowed sequence within this document, ensuring that a given DMS actually supports every possible combination would be impractical. Instead, most agencies will only require that the device be tested to a standard set of procedures, which would use standardized dialogs. To improve communications efficiency, management stations may use non-standard dialogs (e.g., a combination of `GetRequest-PDU`s and/or `SetRequest-PDU`s that is not defined as a standardized dialog, but which a conformant device is required to support (i.e., according to the state transitions defined in Section 4.3, the SYNTAX and MAX-ACCESS clauses of Section 5, and the object refinements defined in A.7. Because these more efficient dialogs may not be known until the acquisition of the management station, which may be years after the acquisition of the device, there is a potential for an interoperability problem to arise.

To overcome this complication, this section defines a lowest common denominator approach to communications between a management station and a DMS. It defines the standardized dialog for each Data Exchange Requirement defined in Section 3. Management stations may support other dialogs to fulfill these same requirements, as long as these dialogs are consistent with the rules defined in this document. Such a management station is termed a 'consistent management station'. A consistent management station interoperates with any 'conformant' device. However, since an agency can not be certain that a device is 100% conformant in every possible scenario (given practical constraints), interoperability problems could still arise.

A 'conformant management station' is required to offer a mode in which it will only use the standardized dialogs as defined in this section. With this limited definition, there is relatively little variability in what constitutes a conformant management station, and fully testing a management station for conformance is a relatively straightforward process that can be done within the practical constraints faced by most procuring agencies. A conformant management station provides an agency with a much greater chance of achieving a base level of interoperability with off-the-shelf devices that have been tested against this document.

The rules for the standardized dialogs follow:

a) The dialogs contain a number of `GetRequest-PDU`s, `GetNextRequest-PDU`s, `GetBulkRequest-PDU`s, and `SetRequest-PDU`s. These requests shall follow the rules defined in NTCIP 2301, which references ISO 15784-2 and RFC 3416. Each specified request shall be transmitted as a single message.
b) The contents of each request are identified by an object name. Each object name consists of an object type and an instance identifier. Formal definitions of each object type are provided in the MIB associated with the object type in the RTM, as defined in Annex A. The meaning of the instance identifier is provided by these same definitions coupled with standard SNMP rules (see RFC 2578).
c) Each message shall contain all of the objects as shown, unless otherwise indicated
d) A message shall not contain any other objects
e) The contents of each message sent by the management station may appear in any order

   Note: Ideally, the order of objects should match the order as shown in this document to provide for the highest probability of interoperability. However, it is recognized that many implementations may use off-the-shelf software, which may prevent the designation of an exact ordering of objects and as a result, this ordering is not a requirement of this document.

f) After sending a message, the management station shall not transmit any other data across the communications channel until the earlier of:
   a. The management station receiving a response from the device or
   b. The expiration of the response time.
g) If the response indicates an error occurred in the operation, the management station shall exit the process, unless specific error-handling rules are specified by the dialog.
h) Dialogs containing a sequence of only `GetRequest-PDU`s may request objects in any order.
i) The device shall process the `GetRequest-PDU`s, `GetNextRequest-PDU`s, `GetBulkRequest-PDU`s, and `SetRequest-PDU`s in accordance with all of the rules of NTCIP 2301, including updating the value in the database and initiating the transmission of the appropriate response (assuming that the device has permission to transmit) within the maximum response time specified.

NOTE—ISO 26048-1 requires a maximum response time of 100ms for standardized requests unless otherwise specified.

In addition, because a consistent management station can alter the order and contents of requests, this document defines rules for when certain data exchanges are allowed. Unless otherwise indicated, a conformant device shall allow an authorized user to retrieve (e.g., through a `GetRequest-PDU`) or alter (e.g., through a `SetRequest-PDU`) any object for which the user has appropriate access rights. However, the access to some data is associated with a state machine and Section 4.3 defines the various rules that apply to these state machines.

4.1 Tutorial [Informative]

The Requirements Traceability Matrix (RTM) presented in Annex A identifies the standardized dialog that can be used to achieve each of the data exchange requirements defined in Section 3.5. Some data exchange requirements reference one of the general SNMP dialogs defined in ISO 26048-1 along with a list of data elements.

This section defines the standardized dialogs that are specific to this document. Each of these dialogs is defined by a number of steps. The data elements referenced within the dialogs are defined in Section 5 and are included in the corresponding row of the RTM with an indication of where they are defined.

Dialogs may also be accompanied by an informative figure that provides a graphical depiction of the normative text. The figures conform to the Unified Modeling Language (UML) and depict the management station as an outside actor sending a series of messages to the device and the device returning responses. If there is any conflict between the figure and the text, the text takes precedence.

Section 4.2 defines how the system is designed to work for a given data exchange requirement. It defunes the standardized sequence of actions that a management station can follow to provide the specific service.

Section 4.3 defines specific state-machine mechanisms used within this document. It describes which states may be present, which transitions are or are not allowed.

4.2 Specified Dialogs

This section provides the standardized data exchange sequences that can be used by management stations to ensure interoperable implementations for the various data exchange requirements identified in Section 3.5.

4.2.1 Calculating the Checksum Value

This document requires the creation and usage of a checksum for several different functions including the graphic ID, font ID, message CRC as well as for the SYNTAX values used by several objects (e.g., MessageActivationCode and MessageCodeID). These checksums shall be calculated the same way in all instances.

The algorithm is based on the CRC-16 algorithm defined in ISO 13239:2002.

The following is provided as an example:

a) Let us assume that the content of the message text to be displayed is "[jp3]TEST [fl]Flashing[/fl]" (=MULTI String content), that the message is to be stored in volatile memory, in slot number 5, and that the sign does not support any physical beacons and no pixel service.
b) The resulting message ID Code is "04 00 05 95 F9" (see below for details).
c) Let us further assume that this message is to be displayed for 267 minutes with activation priority 55.
d) Using this and the above information, the resulting version 4 message activation code is "01 0B 37 04 00 05 95 F9"
Where:

01 0B	2-byte duration value of '267' in hex
37	1-byte priority value of '55' in hex
04	1-byte message type value of 'volatile (4)' in hex
00 05	2-byte message number value of '5' in hex
95 F9	2-byte checksum value for a MULTI-string value of '[jp3]TEST [fl]Flashing[/fl]' in hex

4.2.2 Managing the DMS Configuration

Standardized dialogs for managing the DMS configuration and that are more complex than simple GETs or SETs are defined in the following subsections.

4.2.2.1 Retrieving a Font Definition

The standardized dialog for a management station to retrieve a font shall be as follows:

a) (Precondition) The management station shall be aware of the number of fonts supported by the DMS, the character set supported by the DMS, and which font definition is being requested.
b) The management station shall GET the fontStatus.x and verify the value is 'inUse', 'readyForUse', 'permanent', or 'unmanaged'. If the response is anything else, the management station shall exit this process as the font is not valid.

c) The management station shall GET the following objects:
   1) fontNumber.x,
   2) fontName.x,
   3) fontHeight.x,
   4) fontCharSpacing.x,
   5) fontLineSpacing.x,
   6) fontVersionID.x,
   7) fontStatus.x.
d) For each character of the font, the management station shall GET the following objects:
   1) characterNumber.x.y
   2) characterWidth.x.y
   3) characterBitmap.x.y.

Where:
x = font index
y = character number

Note: Since the character table may be sparsely populated, it is impossible to know which character numbers are supported without custom designing the management station to device documentation, doing an exhaustive search until all characters are found (and receiving a noSuchInstance value for entries that did not exist), or using GET-NEXT operations. The recommended solution for management stations that have to support this feature is to use a series of GET-NEXT operations to poll the device for each row until all rows of the table are retrieved.

This dialog is being used in conjunction with the State Machine Diagram as defined in Section 4.3.1.

4.2.2.2 Configuring a Font

The standardized dialog for a management station to configure a font shall be as follows (see Figure 2):

a) (Precondition) The management station shall be aware of the number of fonts supported by the DMS, the characters supported by the DMS, the font to be configured, and the characters within the font to be configured.
b) The management station shall GET fontStatus.x. If its value is 'inUse' or 'permanent', the management station shall exit the process. The management station may then change the message and restart this process from the beginning.
c) The management station shall SET fontStatus.x to 'modifyReq' to put the selected font in the 'modifying' state.
d) The management station shall GET fontStatus.x. If its value is 'modifying', it is now safe to modify the font data. If its value is not 'modifying', exit the process. (See Section 4.3.1 for a complete state chart diagram for font status).
e) The management station shall SET fontHeight.x to the new value desired to ensure the font is deleted if height changes.
f) The management station shall SET the following data to the desired values:
   1) fontNumber.x,
   2) fontName.x,
   3) fontCharSpacing.x, and
   4) fontLineSpacing.x.
g) The management station shall SET the following data to the desired values for the subject font and the subject character (Repeat for each character to be modified):
   1) characterWidth.x.y and
   2) characterBitmap.x.y.
h) The management station shall SET fontStatus.x to 'readyForUseReq' to allow messages using the font to be displayed successfully.

   Where:

x = font index
y = character number

Note: NTCIP 1203:1997 did not include a fontStatus object. Thus, management stations should be designed to gracefully recover if Step b) results in a noSuchObject value by skipping Steps c), d), and h).

Note: The DMS WG recognizes that the message display on the sign could be unpredictable during the download of a font. Those specifying authorities or application developers who are sensitive to this issue can blank the display during a font download.

This dialog is being used in conjunction with the State Machine Diagram as defined in Section 4.3.1.

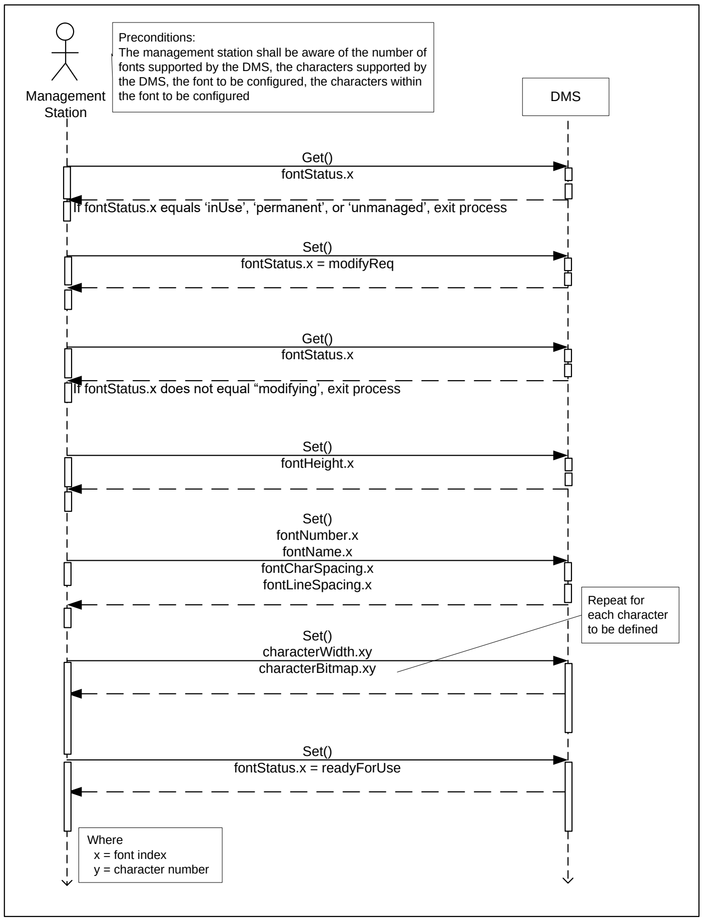
Preconditions:
The management station shall be aware of the number of fonts supported by the DMS, the characters supported by the DMS, the font to be configured, the characters within the font to be configured
Management Station
DMS
Get()
fontStatus.x
If fontStatus.x equals ‘inUse’, ‘permanent’, or ‘unmanaged’, exit process
Set()
fontStatus.x = modifyReq
Get()
fontStatus.x
If fontStatus.x does not equal “modifying”, exit process
Set()
fontHeight.x
Set()
fontNumber.x
fontName.x
fontCharSpacing.x
fontLineSpacing.x
Repeat for each character to be defined
Set()
characterWidth.xy
characterBitmap.xy
Set()
fontStatus.x = readyForUse
Where
x = font index
y = character number

Figure 2 Configuring a Font

4.2.2.3 Deleting a Font

The standardized dialog for a management station to delete a font shall be as follows:

a) (Precondition) The management station will know which font is to be deleted and should ensure that the DMS supports this font.
b) The management station shall GET dmsFontStatus.x. If the value is equal to 'inUse' or 'permanent', the management station shall exit this process as fonts in these states cannot be deleted.
c) The management station shall SET dmsFontStatus.x to 'notUsedReq'.
d) The management station shall GET dmsFontStatus.x and ensure this value is equal to 'notUsed'.

Where:
x = dmsFontIndex

This dialog is being used in conjunction with the State Machine Diagram as defined in Section 4.3.1.

4.2.2.4 Validating a Font

The standardized dialog for a management station to validate a font (i.e., ensure that the font configuration is as expected) shall be as follows:

a) (Precondition) The management station shall be aware of which font it wants to validate and shall ensure that the device supports this font.
b) (Precondition) The management station shall be aware of the expected CRC value for the subject font. The expected CRC value may be based on a previously retrieved value when the font was in a known state, or may be determined directly by calculating the CRC based on the expected font configuration.
c) The management station shall GET fontStatus.x.
d) The management station shall ensure that fontStatus.x equals 'permanent', 'readyForUse', 'unmanaged', or 'inUse'. If fontStatus.x is any other value, the value for fontVersionID.x may not reflect the currently stored data and the management station shall exit this process.
e) The management station shall GET fontVersionID.x.
f) The management station shall compare the expected value for fontVersionID.x to the newly retrieved value. If the values match, the font configuration is (within a very high degree of probability) as expected. If the values are different, the font has changed.

Where:
x = font index

Note: If the fontVersionID values are different, the management station may want to delete the font or download the font again.

Note: NTCIP 1203:1997 did not include a fontStatus object. Thus, management stations should be designed to gracefully recover if Step c) results in a noSuchObject value by skipping Step d).

This dialog is being used in conjunction with the State Machine Diagram as defined in Section 4.3.1.

4.2.2.5 Retrieving a Graphic Definition

The standardized dialog for a management station to retrieve a graphic shall be as follows:

a) (Precondition) The management station shall ensure that the sign supports the graphic to be retrieved.
b) (Precondition) The management station shall ensure that it has the value of dmsGraphicBlockSize so that it can decode the bitmap blocks properly.
c) The management station shall GET dmsGraphicStatus.x. If the value is 'notUsed', 'modifying', or 'calculatingID', exit the process as the graphic is not defined.
d) The management station shall GET the following objects:
    1) dmsGraphicNumber.x

   2) dmsGraphicName.x
   3) dmsGraphicHeight.x
   4) dmsGraphicWidth.x
   5) dmsGraphicType.x
   6) dmsGraphicTransparentEnabled.x,
   7) dmsGraphicTransparentColor.x
e) The management station shall GET the following objects:
   1) dmsGraphicBlockBitmap.x.y (as needed to retrieve entire graphic)

   Note: Repeat Step d) for number of bitmap blocks that is contained within the index of this entry.

Where:
x = dmsGraphicIndex, and dmsGraphicBitmapIndex
y = dmsGraphicBlockNumber

This dialog is being used in conjunction with the State Machine Diagram as defined in Section 4.3.2.

4.2.2.6 Storing a Graphic Definition

The standardized dialog for a management station to download a graphic (see Figure 3) shall be as follows:

a) (Precondition) The management station shall ensure that the row of the graphic table to be changed is within the range of the maximum graphics the DMS can store.
b) (Precondition) The management station shall ensure that it has the value of dmsGraphicBlockSize so that it can encode the bitmap blocks properly.
c) The management station shall GET dmsGraphicStatus.x. If its value is 'inUse', 'permanent', or 'calculatingID', the management station shall exit the process. It may then change the message and then restart this process.
d) The management station shall SET dmsGraphicStatus.x to 'modifyReq' to put the selected graphic in the 'modifying' state.
e) The management station shall GET dmsGraphicStatus.x. If its value is 'modifying', it is now safe to modify the graphic data. If its value is anything other than 'modifying', exit the process. (See Section 4.3.2 for a complete state chart diagram for graphic status.)
f) The management station shall SET the following objects to the desired values:
   1) dmsGraphicNumber.x,
   2) dmsGraphicName.x,
   3) dmsGraphicHeight.x,
   4) dmsGraphicWidth.x,
   5) dmsGraphicType.x,
   6) dmsGraphicTransparentEnabled.x,
   7) dmsGraphicTransparentColor.x.
g) The management station shall SET dmsGraphicBlockBitmap.x.y (as required to store entire graphic).
h) The management station shall SET dmsGraphicStatus.x to 'readyForUseReq' to allow messages using the graphic to be displayed successfully.

Where:
x = dmsGraphicIndex and dmsGraphicBitmapIndex
y = dmsGraphicBlockNumber

This dialog is being used in conjunction with the State Machine Diagram as defined in Section 4.3.2.

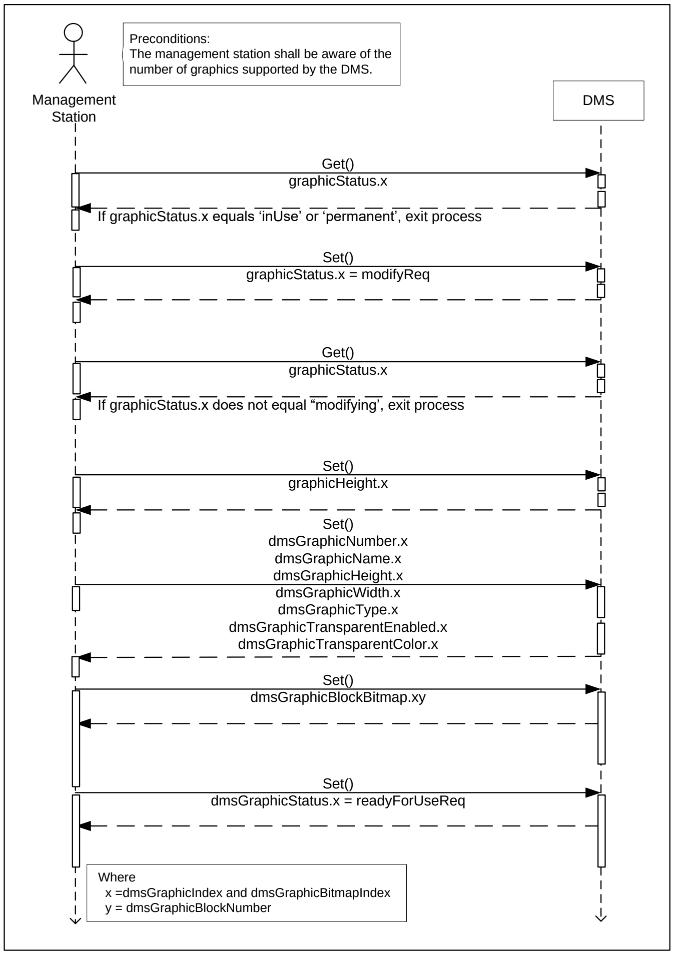
Preconditions:
The management station shall be aware of the number of graphics supported by the DMS.
Management Station
DMS
Get()
graphicStatus.x
If graphicStatus.x equals 'inUse' or 'permanent', exit process
Set()
graphicStatus.x = modifyReq
Get()
graphicStatus.x
If graphicStatus.x does not equal "modifying', exit process
Set()
graphicHeight.x
Set()
dmsGraphicNumber.x
dmsGraphicName.x
dmsGraphicHeight.x
dmsGraphicWidth.x
dmsGraphicType.x
dmsGraphicTransparentEnabled.x
dmsGraphicTransparentColor.x
Set()
dmsGraphicBlockBitmap.xy
Set()
dmsGraphicStatus.x = readyForUseReq
Where
x =dmsGraphicIndex and dmsGraphicBitmapIndex
y = dmsGraphicBlockNumber

Figure 3 Storing a Graphic

4.2.2.7 Deleting a Graphic

The standardized dialog for a management station to delete a graphic shall be as follows:

a) (Precondition) The management station will know which graphic is to be deleted and should ensure that the DMS supports this graphic.
b) The management station shall GET dmsGraphicStatus.x. If the value is equal to 'inUse' or 'permanent', the management station shall exit this process as graphics can not be deleted when they are in use or are permanent.
c) The management station shall SET dmsGraphicStatus.x to 'notUsedReq'.
d) The management station shall GET dmsGraphicStatus.x and ensure this value is equal to 'notUsed'.

Where:
x = dmsGraphicIndex

This dialog is being used in conjunction with the State Machine Diagram as defined in Section 4.3.2.

4.2.2.8 Validating a Graphic

The standardized dialog for a management station to validate a graphic (i.e., ensure that the graphic is as expected) shall be as follows:

a) (Precondition) The management station shall be aware of which graphic it wants to validate and shall ensure that the device supports this graphic.
b) (Precondition) The management station shall be aware of the expected CRC value for the graphic. The expected CRC value may be based on a previously retrieved value when the graphic was in a known state, or may be determined directly by calculating the CRC based on the expected graphic.
c) The management station shall GET dmsGraphicStatus.x and dmsGraphicID.x
d) The management station shall ensure that dmsGraphicStatus.x is equal to 'permanent', 'readyForUse', or 'inUse'. If dmsGraphicStatus.x indicates any other value, the dmsGraphicID.x value may not be valid for the stored information and the management station should exit this process.
e) The management station shall ensure that the retrieved value for dmsGraphicID.x is equal to the expected value. If the values match, the graphic information is (within a very high degree of probability) as expected. If the values are different, the graphic information has changed.
f) (Postcondition) If the values are different, the management station may wish to delete the graphic or download the graphic again.

Where:
x = dmsGraphicIndex

This dialog is being used in conjunction with the State Machine Diagram as defined in Section 4.3.2.

4.2.2.9 Configuring Light Output Algorithm

The standardized dialog for a management station to configure the brightness values (see Figure 4) shall be as follows:

a) (Precondition) The management station shall be aware of the maximum number of photocell levels supported by the device and the desired photocell curve to be stored in the device, which must not contain photocell levels above that supported by the device.
b) The management station shall SET dmsIllumBrightnessValues.0 to the desired value (see Section 5.8.7 for example values).
c) If the response indicates a wrongValue, inconsistentValue, or genErr error, the management station shall GET dmsIllumBrightnessValuesError.0 to determine the cause of the error.

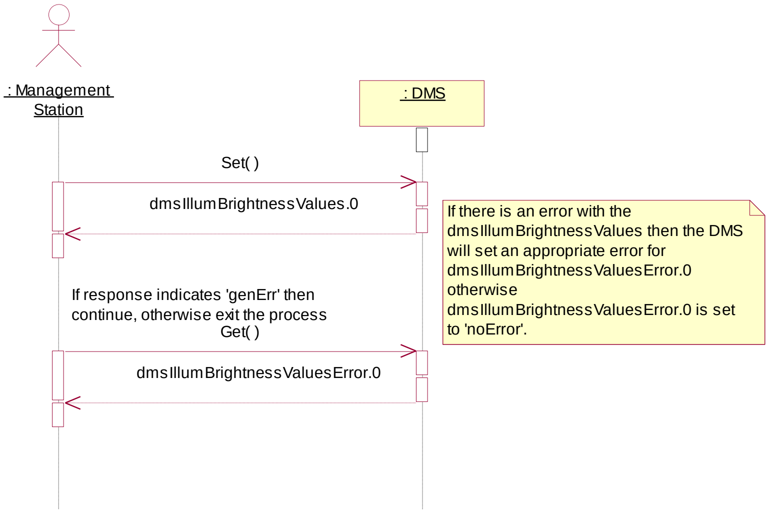
: Management Station
: DMS
Set( )
dmsIllumBrightnessValues.0
If there is an error with the dmsIllumBrightnessValues then the DMS will set an appropriate error for dmsIllumBrightnessValuesError.0 otherwise dmsIllumBrightnessValuesError.0 is set to 'noError'.
If response indicates 'genErr' then continue, otherwise exit the process
Get( )
dmsIllumBrightnessValuesError.0

Figure 4 Configuring Light Output Algorithm

4.2.3 Controlling the DMS

Standardized dialogs for controlling the DMS that are more complex than simple GETs or SETs are defined in the following subsections.

4.2.3.1 Activating a Message

The standardized dialog for a management station to activate a message on the sign display shall be as follows:

a) (Precondition) The management station shall ensure that the desired message is supported by the DMS. This may entail downloading the desired message contents to the DMS. (See Section 4.2.3.2)
b) The management station shall SET dmsActivateMessageV4.0 to the desired value. This will cause the controller to perform a consistency check on the message. (See Section 4.3.5 for a description of this consistency check.)
Note: dmsActivateMessageV4.0 is a structure that contains the following information: message type (permanent, changeable, blank, etc.), message number, duration, activation priority, and a CRC of the message contents.
c) If the message passes the consistency check, the controller shall activate the message and respond with 'noError'. The management station shall GET shortErrorStatus.0 to ensure that there are no errors preventing the display of the message (e.g. a 'criticalTemperature' alarm) after its activation. If the 'other' bit of shortErrorStatus is set, the management station shall GET dmsOtherShortErrorDescription.0 to determine the cause of the error.The management station may then exit the process.
d) if the message fails the consistency check, the controller shall not activate the message and shall respond with an appropriate error value. The management station shall GET dmsActivateMsgError.0 and dmsActivateErrorMsgCodeV4.0 to determine the type of error.
e) If dmsActivateMsgError equals 'syntaxMULTI' then the management station shall GET the following data to determine the error details:
   1) dmsMultiSyntaxError.0
   2) dmsMultiSyntaxErrorPosition.0
f) If dmsActivateMessageError equals “syntaxMULTI(8)” and dmsMultiSyntaxError equals “other(1)”

then the management station shall GET dmsMultiOtherErrorDescription.0 to determine the vendor specific error.

This process is depicted in Figure 5. This dialog is being used in conjunction with the following State Machine Diagrams as defined in Sections 4.3.4 and 4.3.5.

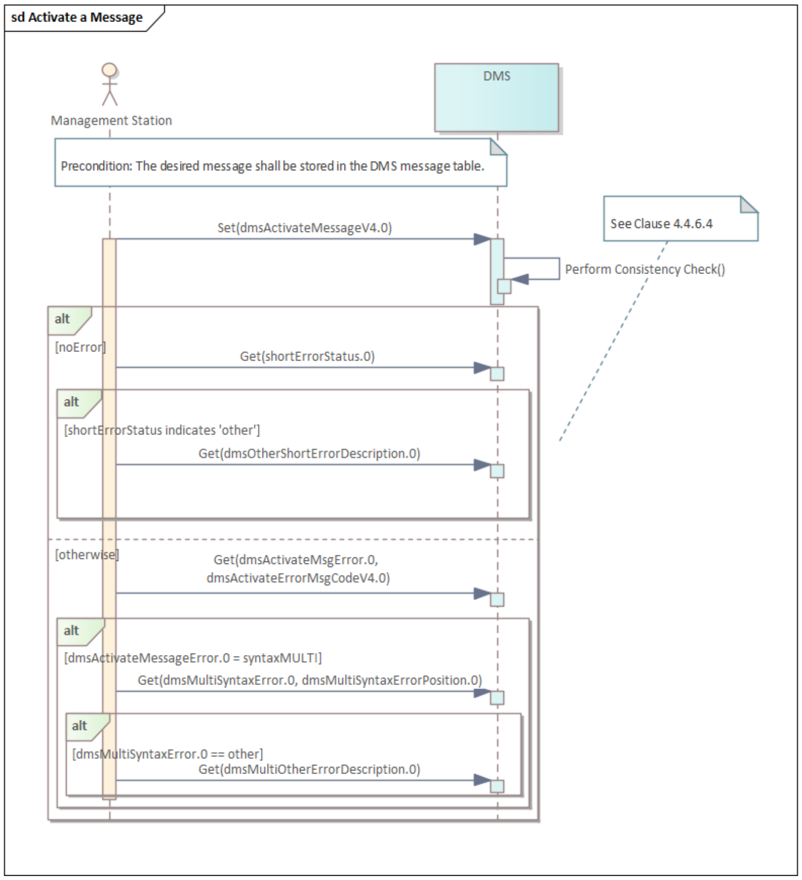
sd Activate a Message
Management Station
DMS
Precondition: The desired message shall be stored in the DMS message table.
See Clause 4.4.6.4
Set(dmsActivateMessageV4.0)
Perform Consistency Check()
alt
[noError]
Get(shortErrorStatus.0)
alt
[shortErrorStatus indicates 'other']
Get(dmsOtherShortErrorDescription.0)
[otherwise]
Get(dmsActivateMsgError.0, dmsActivateErrorMsgCodeV4.0)
alt
[dmsActivateMessageError.0 = syntaxMULTI]
Get(dmsMultiSyntaxError.0, dmsMultiSyntaxErrorPosition.0)
alt
[dmsMultiSyntaxError.0 == other]
Get(dmsMultiOtherErrorDescription.0)

Figure 5 Activating a Message

4.2.3.2 Defining a Message

According to the NTCIP paradigm, no message can be displayed unless it is defined in the message table within the sign controllers' memory. The standardized dialog for a management station to download a message to the DMS (see Figure 6) shall be as follows:

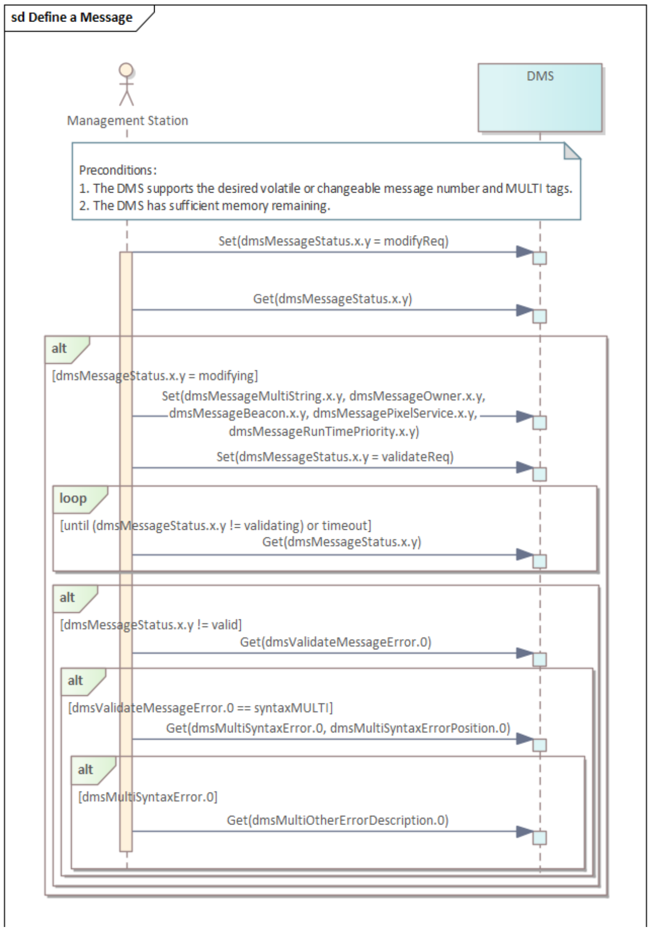
sd Define a Message
Management Station
DMS
Preconditions:
1. The DMS supports the desired volatile or changeable message number and MULTI tags.
2. The DMS has sufficient memory remaining.
Set(dmsMessageStatus.x.y = modifyReq)
Get(dmsMessageStatus.x.y)
alt
[dmsMessageStatus.x.y = modifying]
Set(dmsMessageMultiString.x.y, dmsMessageOwner.x.y, dmsMessageBeacon.x.y, dmsMessagePixelService.x.y, dmsMessageRunTimePriority.x.y)
Set(dmsMessageStatus.x.y = validateReq)
loop
[until (dmsMessageStatus.x.y != validating) or timeout]
Get(dmsMessageStatus.x.y)
alt
[dmsMessageStatus.x.y != valid]
Get(dmsValidateMessageError.0)
alt
[dmsValidateMessageError.0 == syntaxMULTI]
Get(dmsMultiSyntaxError.0, dmsMultiSyntaxErrorPosition.0)
alt
[dmsMultiSyntaxError.0]
Get(dmsMultiOtherErrorDescription.0)

Figure 6 Defining a Message

a) (Precondition) The management station shall ensure that the DMS supports the desired volatile or changeable message number and the tags within the message. The management station should not attempt this procedure for any other message type.
b) (Precondition) The management station shall ensure that there is sufficient storage space remaining for the message to be downloaded.
c) The management station shall SET dmsMessageStatus.x.y to 'modifyReq'.
d) The management station shall GET dmsMessageStatus.x.y.
e) If the value is not 'modifying', exit the process. In this case, the management station may SET dmsMessageStatus.x.y to 'notUsedReq' and attempt to restart this process from the beginning. (See Section 4.3.4 for a complete description of the Message Table State Machine.)
f) The management station shall SET the following data to the desired values:
    1) dmsMessageMultiString.x.y

2) dmsMessageOwner.x.y
3) dmsMessageBeacon.x.y
4) dmsMessagePixelService.x.y
5) dmsMessageRunTimePriority.x.y

Note: Support for dmsMessageBeacon and dmsMessagePixelService could be interprted as optional prior to NTCIP 1203 v04. Implementations that use a proxy agent to communicate to these implementations can overcome this limitation by removing one or both of these objects from the request. Starting with v04, all implementations are required to support these objects, but can constrain the values to zero (0).

g) The management station shall SET dmsMessageStatus.x.y to 'validateReq'. This will cause the controller to initiate a consistency check on the message. (See Section 4.3.5 for a description of this consistency check.)
h) The management station shall repeatedly GET dmsMessageStatus.x.y until the value is not 'validating' or a time-out has been reached.
i) If the value is 'valid', exit the process. Otherwise, the management station shall GET dmsValidateMessageError.0 to determine the reason the message was not validated.
j) If the value is 'syntaxMULTI', the management station shall GET the following data to determine the error details:
   1) dmsMultiSyntaxError.0
   2) dmsMultiSyntaxErrorPosition.0
k) If the value is 'other', the management station shall GET the following data to determine the error details:
   1) dmsMultiOtherErrorDescription.0

Where:
x = message type
y = message number

Note: If, at the end of this process, the value of dmsMessageStatus.x.y is 'valid', the message can be activated.

This dialog is being used in conjunction with the State Machine Diagram as defined in Sections 4.3.4.

4.2.3.3 Retrieving a Message

The standardized dialog for a management station to upload a message from the DMS shall be as follows:

a) (Precondition) The management station shall ensure that the DMS supports the desired message type and number.
b) The management station shall GET the following data:
   1) dmsMessageMultiString.x.y
   2) dmsMessageOwner.x.y
   3) dmsMessageBeacon.x.y
   4) dmsMessagePixelService.x.y
   5) dmsMessageRunTimePriority.x.y
   6) dmsMessageStatus.x.y

Note: Support for dmsMessageBeacon and dmsMessagePixelService could be interprted as optional prior to NTCIP 1203 v04. Implementations that use a proxy agent to communicate to these implementations can overcome this limitation by removing one or both of these objects from the request. Starting with v04, all implementations are required to support these objects, but can constrain the values to zero (0).

Where:
x = message type
y = message number

> Note: The purpose of the dmsMsgSourceMode object is to determine who (person or mechanism) put up a message, while the dmsMsgTableSource object identifies the actual message displayed.

4.2.3.4 Manually Controlling Sign Brightness

The standardized dialog for a management station to manually control the brightness of the sign face shall be as follows:

a) (Precondition)The management station shall be aware of the number of brightness levels supported by the DMS and is shall be aware of the manual control mode that the DMS supports (either direct or indexed).
b) The management station shall SET dmsIllumControl.0 to 'manualDirect' or 'manualIndexed' depending what is desired and supported by the DMS.
Note: Per the definition of dmsIllumControl, this step causes the value stored in dmsIllumManLevel to change to the current brightness level.
c) The management station shall SET dmsIllumManLevel.0 to the desired level ranging from 0 (off) to
   i. maximum number of brightness levels supported by the DMS (if 'manualDirect' is the dmsIllumControl mode) or
   ii. the maximum number of brightness levels defined in the dmsBrightnessValues talbe (if 'manualIndexed' is the dmsIllumControl mode).

   Note: The DMS may implement the new brightness level over a period of time to prevent a visible flicker effect.

4.2.3.5 Manage the Exercise of Pixels

The standardized dialog for a management station to exercise the pixels on a sign face shall be as follows:

a) The management station shall SET vmsPixelServiceTime.0 to the time when the first pixel service is to occur during each day.
b) The management station shall SET the vmsPixelServiceDurationV4.0 to the length of time the pixel service is to last every time it is initiated. If this value is zero, the pixel service is disabled.
c) The management station shall SET vmsPixelServiceFrequency.0 to the time between pixel services. Set this value to zero for continuous exercising.
d) (Postcondition) The management station shall activate a message that is defined to allow pixel service (e.g., the dmsMessagePixelService object for the message has been set to a value of 1). (See Section 4.2.3.2)

4.2.3.6 Activating a Message with Status

The standardized dialog for a management station to activate a message on the sign display shall be as follows:

a) (Precondition) The management station shall follow the steps 1 through 2 within the section 4.2.3.1 "Activating a Message".
b) If the response indicates 'noError', the message may or may not be activated
c) The management station shall GET dmsActivateMessageState.0.
d) If the response from step 3 indicates 'slowActivating(4)', the DMS is in the process of activating the message. Goto step 3.
e) If the response from step 3 indicates 'slowActivatedError(3)', the management station shall GET shortErrorStatus.0 to determine the source of the error.
f) If the response from step 3 indicates 'fastActivationSign(1)', the message is activated continue with step 8.
g) If the response from step 3 indicates 'slowActivatedOK(2)', the message is activated continue with step 8.
h) The message has been activated and the management station shall GET shortErrorStatus.0 to ensure that there are no errors preventing the display of the message (e.g. a 'criticalTemperature'

alarm). If the 'other' bit of shortErrorStatus is set, the management station shall GET dmsOtherShortErrorDescr.0 to determine the cause of the error. The management station may then exit the process.

i) If the response from Step 2 indicates an error, the message was not activated. The management station shall GET dmsActivateMsgError.0 and dmsActivateErrorMsgCodeV4.0 to determine the type of error.
j) If dmsActivateMsgError equals 'syntaxMULTI' then the management station shall GET the following data to determine the error details:
   1) dmsMultiSyntaxError.0
   2) dmsMultiSyntaxErrorPosition.0
k) If dmsActivateMessageError equals “syntaxMULTI(8)” and dmsMultiSyntaxError equals “other(1)” then the management station shall GET dmsMultiOtherErrorDescription.0 to determine the vendor specific error.

   Note: The information is only applicable to the manual activation command, if the Source is 'externalActivation' and the dmsActivateErrorMsgCodeV4 was identical to the activation code sent to the device.

4.2.4 Monitoring the Status of the DMS

Standardized dialogs for monitoring the DMS configuration that are more complex than simple GETs or SETs are defined in the following subsections.

4.2.4.1 Activating Pixel Testing

The standardized dialog for a management station to command the DMS to activate pixel testing shall be as follows:

a) The management station shall SET pixelTestActivation.0 to 'test'.
b) The management station shall repeatedly GET pixelTestActivation.0 until it either returns the value of 'noTest' or a maximum time-out is reached. If the time-out is reached, the DMS is apparently locked and the management station shall exit the process.
c) (PostCondition) The following objects will have been updated during the pixel test to reflect current conditions. The management station may GET any of these objects as appropriate.
   1) pixelFailureTableNumRows
   2) any object within the pixelFailureTable

4.2.4.2 Monitoring Pixel Error Details

The standardized dialog for a management station to monitor details about any pixel errors shall be as follows:

a) (Precondition) The management station shall execute pixel testing. (See Section 4.2.4.1)
b) (Precondition) The management station shall be aware of which pixels are currently failed on or failed off.
c) For the desired failed pixel, the management station shall GET the following data:
   1) pixelFailureXLocation.x.y
   2) pixelFailureYLocation.x.y
   3) pixelFailureStatus.x.y

   Where:
   x = failure detection type
   y = pixel failure index

4.2.4.3 Monitoring Message Activation Error Details

The standardized dialog for a management station to monitor details about any message activation errors shall be as follows:

a) The management station shall GET the following data:

1) dmsActivateMsgError.0
2) dmsActivateErrorMsgCodeV4.0
3) dmsMultiSyntaxError.0
4) dmsMultiSyntaxErrorPosition.0

b) If dmsActivateMessageError equals “syntaxMULTI(8)” and dmsMultiSyntaxError equals “other(1)” then the management station shall GET dmsMultiOtherErrorDescription.0 to determine the vendor specific error.
c) If the dmsActivateMsgError.0 has a value of anything other than 'syntaxMULTI', the full description of the error is given by the value, the remaining data shall be ignored, and the management station shall exit the process.

4.2.4.4 Monitoring the Current Message

The standardized dialog for a management station to monitor the current message shall be as follows:

a) The management station shall GET dmsMsgTableSource.0 to determine the message number, message type, and message CRC of the currently displayed message.
b) The management station shall GET dmsMessageTimeRemaining.0.
c) The management station shall GET dmsMsgRequesterAddressType, dmsMsgRequesterAddress, and dmsMsgRequesterSecurityName to determine the source address of the controller that activated the currently displayed message and the associated security name.
d) The management station shall GET dmsMsgSourceMode.0 to determine the source from which the message was generated (e.g., default message, communications port, scheduler, etc.).
e) The management station shall GET the following data:
   1) dmsMessageMultiString.5.1
   2) dmsMessageOwner.5.1
   3) dmsMessageBeacon.5.1
   4) dmsMesagePixelService.5.1.
   5) dmsMessageRunTimePriority.5.1

   Note: Instance "5.1" is the currentBuffer row of the Message Table.
   Note: The response to this request may include noSuchObject values, when communicating to a proxy to an implementation prior to NTCIP 1203 v04 as these devices were not required to support the beacon or pixel service objects. If the values are missing, the CRC will be calculated with default values of zero (0).
f) The management station shall GET the following data:
   1) dmsIllumBrightLevelStatus.0
   2) dmsIllumLightOutputStatus.0

4.2.4.5 Monitoring Dynamic Field Values

The standardized dialog for a management station to monitor the value of dynamic fields within a message shall be as follows:

a) The management station shall GET statMultiFieldRows.0 to determine the number of dynamic fields used within the current message.
b) For each dynamic field, the management station shall GET the following data:
   1) statMultiFieldCode.x
   2) statMultiCurrentFieldValue.x

   Note: If statMultiFieldRows.0 equals zero (0), step 2 should be skipped.

   Where:
   x = MULTI Field Index

4.3 State Transition Diagrams

State-Transition diagrams are included for those objects that have states or manage states. The State Transition Diagrams include state-transition tables (listing of the possible state transitions), legitimate transitions, and any illegitimate transitions.

“State-transition diagrams describe all of the states that an object can have, the events under which an object changes state (transitions), the conditions that must be fulfilled before the transition will occur (guards), and the activities undertaken during the life of an object (actions).” (Reference: State-Transition Diagrams: Testing UML Models, Part 4 by Lee Copeland)

The following subsections define the states for various object classes that may be supported by the device.

4.3.1 Font State Machine Definition

The DMS shall allow a management station to manage each font through the dmsFontStatus object. The allowed transitions and explanations associated with this diagram are provided within the table below.

4.3.1.1 General Description of the Font State Machine

When the device is not in the 'unmanaged (11)' state and a user desires to modify anything in a font, the font must be in the 'modifying' state otherwise a notWritable error shall be returned. A 'modifyReq (7)' must be issued to put the font into the 'modifying (2)' state. A 'modifyRequest (7)' can only be issued from the following states: 'modifying (2)'; 'readyForUse (4)'; 'notUsed (1)'; or 'unmanaged (11)'. A inconsistentValue will be returned, if a ''modifyReq (7)' request is attempted from any other state.

The following operations are exclusive to the 'modifying (2)' state:

a) Characters may be set in the dmsCharacterTable.
b) The font's parameters may be changed.
c) Setting the dmsFontStatus object to a value of 'readyForUseReq (8)' switches the state to a value of 'calculatingID (3)' as well as causing the font's CRC to be calculated. After that, the value of the dmsFontStatus shall be set to 'readyForUse (4)'.
d) The font state (value of the dmsFontStatus object) can be changed to 'unmanaged (9)' by issueing a request to the dmsFontStatus object using the value 'unManagedReq (10)'.

Font Status can never be changed from a value of 'permanent (6)' to any other state. If attempted, a notWritable error is returned.

At any time it shall be possible to set the dmsFontStatus object to a value of 'notUsedRequest (9)' except when the dmsFontStatus object has a value of 'permanent (6)' or 'inUse (5)'. A value of 'inUse (5)' indicates that the font is used within the currently displayed message on the sign. The first exception case is covered in the previous paragraph, while in the latter two exception cases, an inconsistentValue errorwill be returned .

The dmsFontStatus object can only be commanded to 'unmanagedReq (10)', when the current value of this object is either 'modifying (2)' or 'notUsed (1)' or 'unmanaged (11)', otherwise an inconsistentValue error will be returned.

A managing device shall only be allowed to activate a message if the dmsFontStatus object has a value of 'unmanaged (11)', 'permanent (6)', or 'readyForUse (4)', otherwise an inconsistentValue errorshall be returned and.the dmsActivateMsgError object shall be changed to a value of 'syntaxMULTI' and the dmsMultiSyntaxError object shall be changed to a value of 'fontNotDefined'.

4.3.1.2 Possible Font State Machine Transitions

The following table shows which transitions are allowed by this version of the standard. The table shows the possible transitions, any errors that should be returned, if certain non-allowed transitions are attempted, as well as other information.

Current State	Command or Event	Result
1–notUsed	7–modifyReq	2–modifying
	8–readyForUseReq	inconsistentValue

Current State	Command or Event	Result
	9–notUsedReq	1–notUsed
	10–unmanagedReq	11–unmanaged
	Set any data of the font (within the dmsFontTable or the dmsCharacterTable)	notWritable

	Activate a message using font	activateMsgError = syntaxMULTI
		dmsMultiSyntaxError = fontNotDefined

2–modifying	7–modifyReq	2–modifying
	8–readyForUseReq	3–calculatingID
	9–notUsedReq	1–notUsed
	10–unmanagedReq	11–unmanaged
	Set any data of the font (within the dmsFontTable or the dmsCharacterTable)	process command
	Set dmsFontHeight to a different value	Set characterWidth=0 Set characterBitmap = zero length

	Activate a message using font	activateMsgError = syntaxMULTI
		dmsMultiSyntaxError = fontNotDefined

3–calculatingID	7–modifyReq	inconsistentValue
	8–readyForUseReq	inconsistentValue
	9–notUsedReq	1–notUsed
	10–unmanagedReq	inconsistentValue
	calculation is finished	4–readyForUse
	Set any data of the font (within the dmsFontTable or the dmsCharacterTable)	notWritable

	Activate a message using font	activateMsgError = syntaxMULTI
		dmsMultiSyntaxError = fontNotDefined

4–readyForUse	7–modifyReq	2–modifying
	8–readyForUseReq	4–readyForUse
	9–notUsedReq	1–notUsed
	10–unmanagedReq	inconsistentValue
	Set any data of the font (within the dmsFontTable or the dmsCharacterTable)	notWritable
	Activate a message using font	5-inUse
5–inUse	7–modifyReq	inconsistentValue
	8–readyForUseReq	inconsistentValue
	9–notUsedReq	inconsistentValue
	10–unmanagedReq	inconsistentValue
	Set any data of the font (within the dmsFontTable or the dmsCharacterTable)	notWritable
	Another messsage with the same font is activated	5–inUse
	message is deactivated (“Activate a message not using font”)	4–readyForUse
6–permanent	7–modifyReq	notWritable
	8–readyForUseReq	notWritable
	9–notUsedReq	notWritable
	10–unmanagedReq	notWritable

Current State	Command or Event	Result
	Set any data of the font (within the dmsFontTable or the dmsCharacterTable)	notWritable
	Another messsage with the same font is activated	6–permanent
	message is deactivated (“Activate a message not using font”)	6–permanent
11–unmanaged	7–modifyReq	2–modifying
	8–readyForUseReq	inconsistentValue
	9–notUsedReq	1–notUsed
	10–unmanagedReq	11–unmanaged
	Set any data of the font (within the dmsFontTable or the dmsCharacterTable)	Manufacturer specific or NTCIP 1203 v1
	Activate a message using font	Manufacturer specific or NTCIP 1203 v1

4.3.1.3 Not Used State

When in the notUsed state, the following rules shall apply to the subject font:

a) The DMS shall allow the management station to SET the subject fontStatus object to 'notUsedReq', 'modifyReq', or 'unmanagedReq'.
b) The DMS shall return a inconsistentValue error for a request to SET the subject fontStatus object to any other value.
c) The DMS shall return a notWritable error for any request to SET any other settable font information for the subject font.
d) The DMS shall reject any attempt (internal event or external request) to activate a message using the subject font and shall change the dmsActivateMsgError object to a value of 'syntaxMULTI' and change the dmsMultiSyntaxError object to a value of 'fontNotDefined'.

4.3.1.4 Modifying State

When in the modifying state, the following rules shall apply to the subject font:

a) The DMS shall allow the management station to SET the subject fontStatus object to 'notUsedReq', 'modifyReq', 'readyForUseReq', or 'unmanagedReq'.
b) If the management station SETs the fontStatus to 'readyForUseReq', the DMS shall automatically update the value of fontVersionID prior to setting the state to 'readyForUse' and set the value of the the subject fontStatus object to 'calculatingID'.
c) If the management station SETs the fontStatus to 'readyForUseReq', if the corresponding fontNumber is not unique among all fonts with fontStatus set to 'permanent', 'readyForUse', 'inUse', or 'unmanaged', a inconsistentValue error will be returned and the fontStatus will change to 'notInUse'.
d) The DMS shall return a inconsistentValue error for a request to SET the subject fontStatus object to any other value.
e) The DMS shall allow a management station to SET any other settable font information for the subject font.
f) The DMS shall reject any attempt (internal event or external request) to activate a message using the subject font and shall change the dmsActivateMsgError object to a value of 'syntaxMULTI' and change the dmsMultiSyntaxError object to a value of 'fontNotDefined'.
g) If the management station SETs the fontHeight to a different value, the DMS shall set all corresponding characterWidth objects to zero (0) and all corresponding characterBitmap objects to zero length.

4.3.1.5 Calculating ID State

When in the calculatingID state, the following rules shall apply to the subject font:

a) The DMS shall update the fontVersionID and then transition to the 'readyForUse' state.
b) The DMS shall allow the management station to SET the subject fontStatus object to 'notUsedReq'.
c) The DMS shall return a inconsistentValue error for a request to SET the subject fontStatus object to any other value.
d) The DMS shall return a notWritable error for any request to SET any other settable font information for the subject font.
e) The DMS shall reject any attempt (internal event or external request) to activate a message using the subject font and shall change the dmsActivateMsgError object to a value of 'syntaxMULTI' and change the dmsMultiSyntaxError object to a value of 'fontNotDefined'.

4.3.1.6 Ready for Use State

When in the readyForUse state, the following rules shall apply to the subject font:

a) The DMS shall allow the management station to SET the subject fontStatus object to 'notUsedReq', 'modifyReq', or 'readyForUseReq'.
b) The DMS shall return a inconsistentValue error for a request to SET the subject fontStatus object to any other value.
c) The DMS shall return a notWritable error for any request to SET any other settable font information for the subject font.
d) The DMS shall allow the font to be used in a message being activated.
e) Upon the successful activation of a message using the font, the subject fontStatus object shall change to 'inUse'.

4.3.1.7 In Use State

When in the inUse state, the following rules shall apply to the subject font:

a) The DMS shall return a inconsistentValue error for a request to SET the subject fontStatus object.
b) The DMS shall return a notWritable error for any request to SET any other settable font information for the subject font.
c) Upon the activation of another message using the subject font, the subject fontStatus object shall remain in the 'inUse' state.
d) Upon the successful activation of another message that does not use the subject font, the subject fontStatus object shall change to the 'readyForUse' state.

4.3.1.8 Permanent State

When in the permanent state, the following rules shall apply to the subject font:

a) The DMS shall return a inconsistentValue error for any request to SET the subject fontStatus object.
b) The DMS shall return a notWritable error for any request to SET any other settable font information for the subject font.
c) The DMS shall allow the font to be used in a message being activated.
d) Upon the successful activation of a message using the font, the fontStatus shall remain 'permanent'.
e) Upon successful activation of a message that does not use the font, the fontStatus shall remain 'permanent'.

4.3.1.9 Unmanaged State

The 'unmanaged' state has been developed to allow for backwards compatibiity allowing a management station to manage both Version 1 and Version 2 signs. When in the 'unmanaged' state, the following rules shall apply to the subject font:

a) The DMS shall allow the management station to SET the subject fontStatus object to 'notUsedReq', 'modifyReq', or 'unmanagedReq'.
b) The DMS shall return a inconsistentValue error for a request to SET the subject fontStatus object to

any other value.

c) When requesting to SET any other settable font information for the subject font, the DMS is assumed to be a Version 1 sign and to react accordingly, i.e., manufacturer-specific.
d) When requesting to activate a message using this font, the DMS is assumed to be a Version 1 sign and to react accordingly, i.e., manufacturer-specific.

4.3.1.10 Other Restrictions

The DMS shall return a notWritable to any SET request containing a fontStatus object for a subject font and any other settable font information for the subject font.

The DMS shall not impose any restrictions on the operations of a subject font based on the status of any other font supported by the DMS (e.g., Font 1 shall not be disabled because Font 2 is being modified).

The contents of the fontVersionID object shall only be considered valid when the fontStatus is readyForUse, inUse, permanent, or unmanaged (i.e., DMS is assumed to be a Version 1 sign).

> Note: Modifying a font that is associated with a permanent message should be performed with extreme caution to prevent undesirable results.

4.3.1.11 Backwards Compatibility

If a sign supports only Version 1, then the fontStatus object will not exist, and this will be equivalent to fontStatus being set to unmanaged (9). If a font is in the 'unmanaged' state, fonts will be modified exactly as before in Version 1, and it will not be possible for a font to have a 'permanent' state.

> Note: DMS conforming to NTCIP 1203:1997 or its Amendment 1 do not support the fontStatus state machine or SNMPv3. However, because a version 1 device can be connected through a proxy agent that converts between SNMPv1 and SNMPv3; an SNMPv3 management station can still be (indirectly) communicate with an SNMPv1 implementation. Further, the exact time at which the fontVersionID is calculated in such devices is not formally defined, but as the correct value was required upon any poll, most manufacturers updated the fontVersionID upon each change to either the font or the character table.

4.3.2 Graphic State Machine Definition

The DMS shall allow a management station to monitor the current state of each graphic through the dmsGraphicStatus object. Figure 7 depicts the state transition diagram for a graphic; the detailed rules associated with this diagram follow.

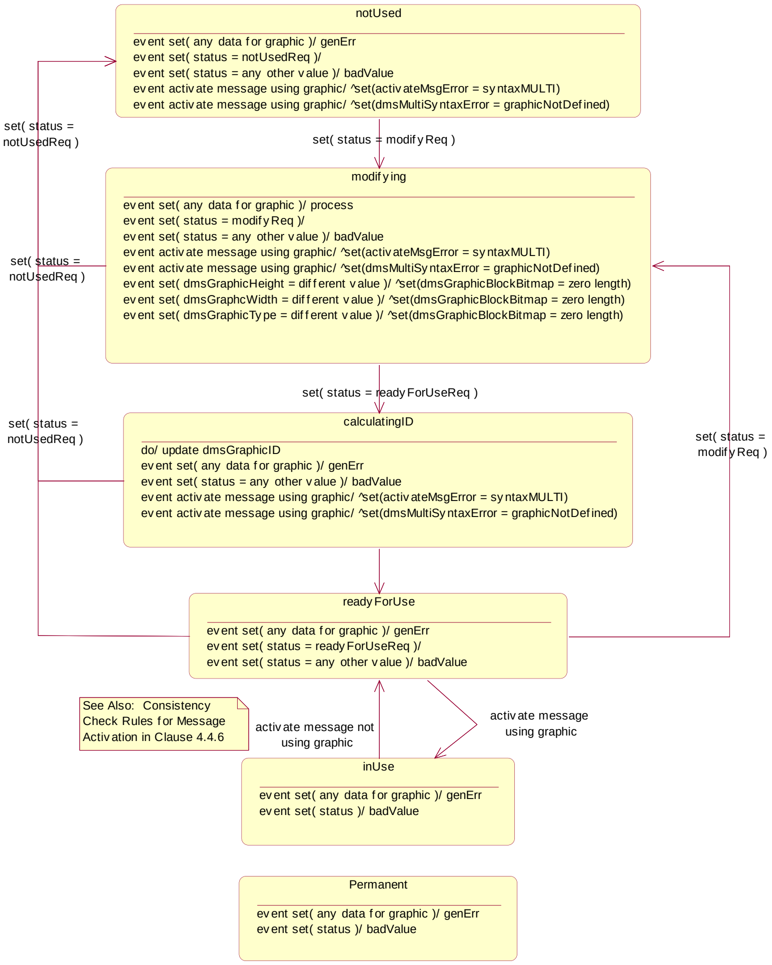
notUsed
event set(any data for graphic)/ genErr
event set(status = notUsedReq)/
event set(status = any other value)/ badValue
event activate message using graphic/ ^set(activateMsgError = syntaxMULTI)
event activate message using graphic/ ^set(dmsMultiSyntaxError = graphicNotDefined)
set(status = notUsedReq)
set(status = modifyReq)
modifying
event set(any data for graphic)/ process
event set(status = modifyReq)/
event set(status = any other value)/ badValue
event activate message using graphic/ ^set(activateMsgError = syntaxMULTI)
event activate message using graphic/ ^set(dmsMultiSyntaxError = graphicNotDefined)
event set(dmsGraphicHeight = different value)/ ^set(dmsGraphicBlockBitmap = zero length)
event set(dmsGraphcWidth = different value)/ ^set(dmsGraphicBlockBitmap = zero length)
event set(dmsGraphicType = different value)/ ^set(dmsGraphicBlockBitmap = zero length)
set(status = notUsedReq)
set(status = readyForUseReq)
calculatingID
do/ update dmsGraphicID
event set(any data for graphic)/ genErr
event set(status = any other value)/ badValue
event activate message using graphic/ ^set(activateMsgError = syntaxMULTI)
event activate message using graphic/ ^set(dmsMultiSyntaxError = graphicNotDefined)
set(status = notUsedReq)
set(status = modifyReq)
readyForUse
event set(any data for graphic)/ genErr
event set(status = readyForUseReq)/
event set(status = any other value)/ badValue
See Also: Consistency Check Rules for Message Activation in Clause 4.4.6
activate message not using graphic
activate message using graphic
inUse
event set(any data for graphic)/ genErr
event set(status)/ badValue
Permanent
event set(any data for graphic)/ genErr
event set(status)/ badValue

Figure 7 Graphic State Machine

4.3.2.1 Not Used State

When in the notUsed state, the following rules shall apply to the subject graphic:

a) The DMS shall allow the management station to SET the subject dmsGraphicStatus object to 'notUsedReq' or 'modifyReq'.
b) The DMS shall return a inconsistentValue error for a request to SET the subject dmsGraphicStatus object to any other value.
c) The DMS shall return a notWritable error for any request to SET any other settable graphic information for the subject graphic.
d) The DMS shall reject any attempt (internal event or external request) to activate a message using the

subject graphic and shall change the dmsActivateMsgError object to a value of 'syntaxMULTI' and change the dmsMultiSyntaxError object to a value of 'graphicNotDefined'.

4.3.2.2 Modifying State

When in the modifying state, the following rules shall apply to the subject graphic:

a) The DMS shall allow the management station to SET the subject dmsGraphicStatus object to 'notUsedReq', 'modifyReq', or 'readyForUseReq'.
b) If the management station SETs the dmsGraphicStatus to 'readyForUseReq', the DMS shall automatically update the value of dmsGraphicID prior to setting the state to 'readyForUse'.
c) The DMS shall return a inconsistentValue error for a request to SET the subject dmsGraphicStatus object to any other value.
d) The DMS shall allow a management station to SET any other settable graphic information for the subject graphic.
e) The DMS shall reject any attempt (internal event or external request) to activate a message using the subject graphic and shall change the dmsActivateMsgError object to a value of 'syntaxMULTI' and change the dmsMultiSyntaxError object to a value of 'graphicNotDefined'.
f) If the management station SETs the graphicHeight, graphicWidth, or graphicType to a different value, the corresponding dmsGraphicBlockBitmap objects shall be set to zero length.

4.3.2.3 Calculating ID State

When in the calculatingID state, the following rules shall apply to the subject graphic:

a) The DMS shall update the dmsGraphicID and then transition to the readyForUse state.
b) The DMS shall allow the management station to SET the subject dmsGraphicStatus object to 'notUsedReq'.
c) The DMS shall return a inconsistentValue error for a request to SET the subject dmsGraphicStatus object to any other value.
d) The DMS shall return a notWritable error for any request to SET any other settable information for the subject graphic.
e) The DMS shall reject any attempt (internal event or external request) to activate a message using the subject graphic and shall change the dmsActivateMsgError object to a value of 'syntaxMULTI' and change the dmsMultiSyntaxError object to a value of 'graphicNotDefined'.

4.3.2.4 Ready for Use State

When in the readyForUse state, the following rules shall apply to the subject graphic:

a) The DMS shall allow the management station to SET the subject dmsGraphicStatus object to 'notUsedReq', 'modifyReq', or 'readyForUseReq'.
b) The DMS shall return a inconsistentValue error for a request to SET the subject dmsGraphicStatus object to any other value.
c) The DMS shall return a notWritable error for any request to SET any other settable graphic information for the subject graphic.
d) The DMS shall allow the graphic to be used in a message being activated.
e) Upon the successful activation of a message using the graphic, the dmsGraphicStatus shall change to 'inUse'.

4.3.2.5 In Use State

When in the inUse state, the following rules shall apply to the subject graphic:

a) The DMS shall return a inconsistentValue error for a request to SET the subject dmsGraphicStatus object.
b) The DMS shall return a notWritable error for any request to SET any other settable graphic information for the subject graphic.
c) Upon the activation of another message using the subject graphic, the dmsGraphicStatus shall remain in the 'inUse' state.

d) Upon the successful activation of another message that does not use the subject graphic, the dmsGraphicStatus shall change to the 'readyForUse' state.

4.3.2.6 Permanent State

When in the permanent state, the following rules shall apply to the subject graphic:

a) The DMS shall return a inconsistentValue error for any request to SET the subject dmsGraphicStatus object.
b) The DMS shall return a notWritable error for any request to SET any other 'settable' graphic information for the subject graphic.
c) The DMS shall allow the graphic to be used in a message being activated.
d) Upon the successful activation of a message using the graphic, the dmsGraphicStatus shall remain 'permanent'.
e) Upon successful activation of a message that does not use the graphic, the dmsGraphicStatus shall remain 'permanent'.

4.3.2.7 Other Restrictions

The DMS shall return a notWritable to any SET request containing a dmsGraphicStatus object for a subject graphic and any other settable graphic information for the subject font.

The DMS shall not impose any restrictions on the operations of a subject graphic based on the status of any other graphic supported by the DMS. (e.g., Graphic 1 shall not be disabled because Graphic 2 is being modified).

The contents of the dmsGraphicID object shall only be considered valid when the dmsGraphicStatus is readyForUse, inUse, or permanent.

> Note: Modifying a graphic that is associated with a permanent message should be performed with extreme caution to prevent undesirable results.

> Note: DMS conforming to NTCIP 1203:1997 or its Amendment 1 do not support the graphic feature.

4.3.3 Control Mode State Machine Definition

See Figure 8.

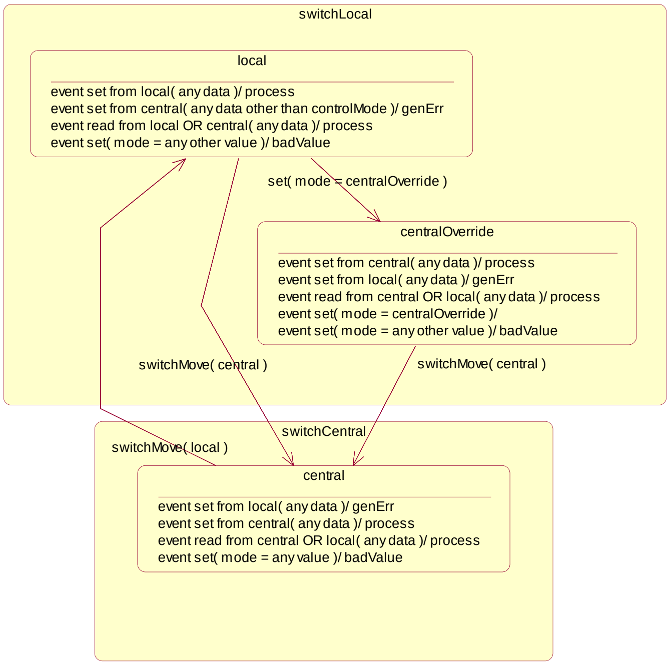
switchLocal
local
event set from local(any data)/ process
event set from central(any data other than controlMode)/ genErr
event read from local OR central(any data)/ process
event set(mode = any other value)/ badValue
set(mode = centralOverride)
centralOverride
event set from central(any data)/ process
event set from local(any data)/ genErr
event read from central OR local(any data)/ process
event set(mode = centralOverride)/
event set(mode = any other value)/ badValue
switchMove(central)
switchMove(central)
switchCentral
switchMove(local)
central
event set from local(any data)/ genErr
event set from central(any data)/ process
event read from central OR local(any data)/ process
event set(mode = any value)/ badValue

Figure 8 Control Mode State Machine

4.3.3.1 Central Mode

When in the 'central' mode, the following rules shall apply to the DMS:

a) If the Local Control Switch is switched to the 'switchLocal' state, the DMS controlMode shall transfer to the 'local' state.
b) The DMS shall return a inconsistentValue error for a request to SET the controlMode to any value.
c) The DMS shall allow a management station to GET any information stored in the DMS.
d) The DMS shall process any SET request from a management station connected via a central port.
e) The DMS shall return a noAccess error for any SET request from a management station connected via a local port. If the request included a SET on dmsActivateMessageV4.0, the DMS shall also update the value of dmsActivateMsgError to 'centralMode', shall update the value of dmsActivateErrorMsgCodeV4 to the message code sent in the request, and update the value of dmsMultiSyntaxError to 'none'.

4.3.3.2 Local Mode

When in the 'local' mode, the following rules shall apply to the DMS:

a) If the Local Control Switch is switched to the 'switchCentral' state, the DMS controlMode shall transfer to the 'central' state.
b) The DMS shall allow a management station (either local or central) to SET the controlMode to 'centralOverride'.
c) The DMS shall return a inconsistentValue error for a request to SET the controlMode to any other

value.

d) The DMS shall allow a management station to GET any information stored in the DMS.
e) The DMS shall process any SET request from a management station connected via a 'local' port.
f) The DMS shall return a noAccess error for any SET request from a management station connected via a 'central' port, except for SETting controlMode to 'centralOverride'. If the request included a SET on dmsActivateMessageV4.0, the DMS shall also update the value of dmsActivateMsgError to 'localMode', shall update the value of dmsActivateErrorMsgCodeV4 to the message code sent in the request, and update the value of dmsMultiSyntaxError to 'none'.

4.3.3.3 Central Override Mode

When in the 'centralOverride' mode, the following rules shall apply to the DMS:

a) If the Local Control Switch is switched to the 'switchCentral' state, the DMS controlMode shall transfer to the 'central' state.
b) The DMS shall allow a management station connected through a 'central' port to SET the controlMode to 'centralOverride'.
c) The DMS shall return a inconsistentValue error for a request to SET the controlMode to any other value.
d) The DMS shall allow a management station to GET any information stored in the DMS.
e) The DMS shall process any SET request from a management station connected via a central port.
f) The DMS shall return a noAccess error for any SET request from a management station connected via a local port. If the request included a SET on dmsActivateMessageV4.0, the DMS shall also update the value of dmsActivateMsgError to 'centralOverrideMode', shall update the value of dmsActivateErrorMsgCodeV4 to the message code sent in the request, and update the value of dmsMultiSyntaxError to 'none'.

4.3.3.4 Other Restrictions

The DMS shall return a genErr to any SET request containing the controlMode object and any other data.

4.3.4 Message Table State Machine Definition

See Figure 9.

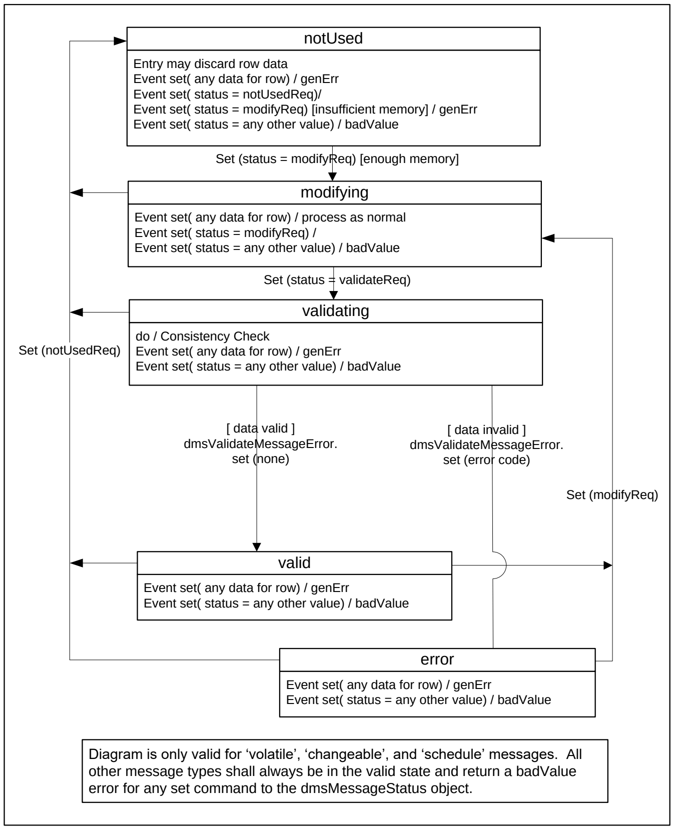
notUsed
Entry may discard row data
Event set(any data for row) / genErr
Event set(status = notUsedReq)/
Event set(status = modifyReq) [insufficient memory] / genErr
Event set(status = any other value) / badValue
Set (status = modifyReq) [enough memory]
modifying
Event set(any data for row) / process as normal
Event set(status = modifyReq) /
Event set(status = any other value) / badValue
Set (status = validateReq)
validating
do / Consistency Check
Event set(any data for row) / genErr
Event set(status = any other value) / badValue
Set (notUsedReq)
[data valid]
dmsValidateMessageError.
set (none)
[data invalid]
dmsValidateMessageError.
set (error code)
Set (modifyReq)
valid
Event set(any data for row) / genErr
Event set(status = any other value) / badValue
error
Event set(any data for row) / genErr
Event set(status = any other value) / badValue

Diagram is only valid for ‘volatile’, ‘changeable’, and ‘schedule’ messages. All other message types shall always be in the valid state and return a badValue error for any set command to the dmsMessageStatus object.

Figure 9 Message Table State Machine

Note: See Consistency Check rules in Section 4.3.5.

4.3.4.1 Not Used State

When in the 'notUsed' state, the following rules shall apply to the subject message:

a) Upon entry into this state, the DMS may discard any data for the message.
b) The DMS shall allow the management station to SET the subject dmsMessageStatus object to 'notUsedReq'.
c) If the management station attempts to SET the the subject dmsMessageStatus object to 'modifyReq', the DMS shall ensure that there is sufficient memory to store at least a minimal message. If the check indicates that there is sufficient memory, the state shall transfer to the 'modifying' state; otherwise it shall remain in the 'notUsed' state and the DMS shall respond with a resourceUnavailable.
d) The DMS shall return a inconsistentValue error for a request to SET the subject dmsMessageStatus object to any other value.
e) The DMS shall return a notWritable error for any request to SET any other settable message information for the subject message.

4.3.4.2 Modifying State

When in the 'modifying' state, the following rules shall apply to the subject message:

a) The DMS shall allow the management station to SET the subject dmsMessageStatus object to 'notUsedReq', 'modifyReq', or 'validateReq'.
b) The DMS shall return a inconsistentValue error for a request to SET the subject dmsMessageStatus object to any other value.
c) The DMS shall process any SET request for settable message information for the subject message.

4.3.4.3 Validating State

When in the 'validating' state, the following rules shall apply to the subject message:

a) Upon entry into this state, the DMS shall perform the following consistency check:
    1) If the requested message type is not supported by the sign, the DMS shall return a noCreation error and shall set the dmsActivateMsgError to 'memoryType'.
    2) If the requested message number is not supported by the sign, the DMS shall return a noCreation error and shall set the dmsActivateMsgError to messageNumber.
    3) If the object dmsMessageMultiString contains MULTI tags that are not supported by the sign or if the resulting message text (text and/or graphics) could not be displayed on the display panel due excess size, excess length, and/or formatting of the text, the dmsMessageStatus shall automatically transfer to the 'error' state and the DMS shall set the dmsValidateMessageError object to a value of 'syntaxMULTI'. In addition, the DMS shall set proper values for the following objects to the appropriate values: dmsMultiSyntaxError, dmsMultiSyntaxErrorPosition, and dmsMultiOtherErrorDescription.
    4) If object dmsMessageMultiString contains text and/or graphics that can not be supported by the DMS, the DMS shall set the dmsValidateMessageError object to 'syntaxMULTI' and shall set proper values for the dmsMultiSyntaxError, dmsMultiSyntaxErrorPosition, and dmsMultiOtherErrorDescription objects.
    5) If the message is not validated for any other reason, the DMS shall set the dmsValidateMessageError object to 'other'.
    6) Otherwise, the consistency check for this object passes, the dmsMessageStatus shall automatically transfer to the 'valid' state, and the DMS shall: set the dmsValidateMessageError object to 'none', set the dmsMultiSyntaxError object to a value of 'none', and set the dmsMultiSyntaxErrorPosition object to a value of zero (0).
b) The DMS shall allow the management station to SET the subject dmsMessageStatus object to 'notUsedReq'. Upon the receipt of such a request, the DMS shall terminate the validation process and transfer to the 'notUsed' state.
c) The DMS shall return a inconsistentValue error for a request to SET the subject dmsMessageStatus object to any other value.
d) The DMS shall return a notWritable error for any request to SET any other settable message information for the subject message.

4.3.4.4 Valid State

When in the 'valid' state, the following rules shall apply to the subject message:

a) The DMS shall allow the management station to SET the subject dmsMessageStatus object to 'notUsedReq' or 'modifyReq'.
b) The DMS shall return a inconsistentValue error for a request to SET the subject dmsMessageStatus object to any other value.
c) The DMS shall return a notWritable error for any request to SET any other settable message information for the subject message.

4.3.4.5 Error State

When in the 'error' state, the following rules shall apply to the subject message:

a) The DMS shall allow the management station to SET the subject dmsMessageStatus object to 'notUsedReq' or 'modifyReq'.

b) The DMS shall return a inconsistentValue error for a request to SET the subject dmsMessageStatus object to any other value.
c) The DMS shall return a notWritable error for any request to SET any other settable message information for the subject message.

4.3.4.6 Other Restrictions

The DMS shall return a genErr to any SET request containing a dmsMessageStatus object for a subject message and any other settable message information for the subject message.

The DMS shall not impose any restrictions on the operations of a subject message based on the status of any other message supported by the DMS.

The contents of the dmsMessageCRC object shall only be considered valid when the dmsMessageStatus is 'valid'.

4.3.5 Message Activation Consistency Check Definition

Whenever a message activation is attempted, whether by a management station SETting the dmsActivateMessageV4.0 object or via an internal message activation attempt (e.g., end duration, trigger event, etc), the DMS shall perform the following consistency checks, in order. If the message activation attempt was due to a management station SETting the dmsActivateMessageV4.0 object, the DMS shall return the response as indicated (otherwise there is no response message to be sent):

a) If the request is valid and received from a 'central' communications port, and the object dmsControlMode has a value of 'local', the DMS shall return a notWritable error and shall set the dmsActivateMsgError to 'localMode'.
b) If the request is valid and received from a 'local' communications port, and the object dmsControlMode has a value of 'central', the DMS shall return a notWritable error and shall set the dmsActivateMsgError to 'centralMode'.
c) If the request is valid and received from a 'local' communications port, and the object dmsControlMode has a value of 'centralOverride', the DMS shall return a notWritable error and shall set the dmsActivateMsgError to 'centralOverrideMode'.
d) If the requested message type is not supported by the sign, the DMS shall return a wrongValue error and shall set the dmsActivateMsgError to 'memoryType'.
e) If the requested message number is not supported by the sign, the DMS shall return a wrongValue error and shall set the dmsActivateMsgError to messageNumber.
f) If the requested message is supported by the sign but is not currently in the valid state, the DMS shall return an inconsistentValue error and shall set the dmsActivateMsgError to messageStatus.
g) If the requested message is in the valid state but has a different CRC value than indicated in the set request, the DMS shall return an inconsistentValue error and shall set the dmsActivateMsgError to messageCRC.
h) If the request is valid, but has insufficient priority to override the current message, the DMS shall return an inconsistentValue error and shall set the dmsActivateMsgError to priority.
i) If the request is valid and has sufficient priority to override the current message, but cannot be displayed due to some error in presenting the MultiString on the display panel, the DMS shall return a genErr and shall set the dmsActivateMsgError to 'syntaxMULTI'. In addition, the DMS shall set proper values for the dmsMultiSyntaxError, dmsMultiSyntaxErrorPosition, dmsMultiOtherErrorDescription, and dmsActivationErrorMsgCode objects.
j) If the request does not result in activating the requested message for any other reason, the DMS shall return a genErr and shall set the dmsActivateMsgError to 'other'.
k) Otherwise, the consistency check for this object passes and the DMS shall set the dmsActivateMsgError to 'none'.

Section 5
Management Information Base (MIB) [Normative]

This section defines those objects which are specific to dynamic message signs (DMS). The objects are defined using the OBJECT-TYPE macro as specified in NTCIP 8004, which is based on RFC 2578 and related standards. The text provided from Section 5.1 through the end of Section 5 (except the section headings) constitutes the NTCIP1203-Dms MIB.

This section generally presents the objects in lexigraphical order of their OBJECT IDENTIFIERS, which correspond to their physical location within the global naming tree. Most of the objects defined in this document reside under the “dms” node of the global naming tree. To aid in object management, the “dms” node has been subdivided into logical categories, each defined by a node under the “dms” node. The individual objects are then located under the appropriate node.

Conformance requirements for any object are determined by the use of the Requirements Traceability Matrix (RTM) in Annex A. To support any defined Requirement, an implementation shall support all objects to which the Requirement traces in the RTM. An implementation shall support the full standardized range of each supported object, unless otherwise noted in the object refinement table in A.6. An implementation shall indicate their level of conformance to this standard using an AGENT-CAPABILITIES statement, as defined by RFC 2580, which indicates supported OBJECT-GROUPs and object ranges.

This MIB is managed by the NTCIP DMS Working Group and proprietary features should be defined through vendor-specific nodes in vendor-specific extensions to this MIB. All values not explicitly defined (e.g., enumerated values not listed, bits not defined, etc.) are reserved for future use by the DMS Working Group and shall not be used by implementations until defined by the DMS WG.

A computer readable format of this information, called a Management Information Base, is available from https://github.com/ite-org/NTCIP1203. The MIB has been verified using https://www.simpleweb.org/ietf/mibs/validate/.

Previous versions of this document defined data elements that have been replaced to resolve ambiguities; however, central systems may need to interoperate with older equipment and support such data elements. These replaced objects have a status of 'deprecated' or 'obsolete' and indicate the version in which they were deprecated; Annex D documents the reason that the WG decided to deprecate the various objects within the current version.

5.1 Object Definitions

```
NTCIP1203-Dms DEFINITIONS ::= BEGIN
IMPORTS
MODULE-IDENTITY, OBJECT-IDENTITY, OBJECT-TYPE, Integer32, Unsigned32,
Counter32, IpAddress
                                               FROM SNMPv2-SMI
                                                 -- RFC 2578
TEXTUAL-CONVENTION, DisplayString, StorageType, RowStatus
                                               FROM SNMPv2-TC
                                                 -- RFC 2579
MODULE-COMPLIANCE, OBJECT-GROUP
                                               FROM SNMPv2-CONF
                                                 -- RFC 2580
SnmpAdminString
```

```
                                               FROM SNMP-FRAMEWORK-MIB
                                                 -- RFC 3411
TransportAddressType, TransportAddress
                                               FROM TRANSPORT-ADDRESS-MIB
                                                 -- RFC 3419
ITSCounter32, ITSDirection, ITSInteger8, ITSLatitude84, ITSLongitude84
                                               FROM ISO26048-1-FieldDevice-TC
fdSrsaTypeCode, fdSrsaPortIndex
                                               FROM ISO26048-1-SRSA
NtcipOwnerString, NtcipIntegerBitmap8, NtcipIntegerBitmap16,
NtcipOctetBitmap32, NtcipOctetBitmap512, NtcipOctetBitmap2040,
NtcipOctetBitmap3200, NtcipTwoColors, devices
                                               FROM NTCIP8004-Transportation;

dms MODULE-IDENTITY
  LAST-UPDATED "202405020500Z"
  ORGANIZATION "NTCIP DMS WG"
  CONTACT-INFO
   "name: NTCIP Coordinator
    email: ntcip@nema.org
    postal: National Electrical Manufacturers Association
            1300 North 17th Street, Suite 1752
            Rosslyn, Virginia 22209-3801"
  DESCRIPTION
  "<Definition> This MIB defines the structure of management information
    under the dms and adminDms nodes, which are used to manage dynamic
    message signs.
  <Object Identifier> 1.3.6.1.4.1.1206.4.2.3"

  REVISION "202405020500Z"
  DESCRIPTION
    "NTCIP 1203 v04 - Upgraded to SMIv2 format and incorporated objects from
      the ISO 20684 series into design."
  REVISION "201409300500Z"
  DESCRIPTION
    "NTCIP 1203 v03 - Addition of test procedures to the standard."
  REVISION "201011300500Z"
  DESCRIPTION
    "NTCIP 1203 v02 - Updated copyright, MIB distribution format, and
      incorporated NTCIP 8004 format."
  REVISION "200107030500Z"
  DESCRIPTION
    "NTCIP 1203 v01A1 - Amendment 1. Corrections based on user experience."
  REVISION "199912010500Z"
  DESCRIPTION
    "NTCIP 1203 v01 - Adopted by AASHTO and ITE in 1998; editorial changes
      for publication."
  REVISION "199703310500Z"
  DESCRIPTION
    "NEMA TS 3.6 - Initial version of MIB as approved by NEMA. "
  ::= { devices 6 }


MessageIDCode ::= TEXTUAL-CONVENTION
  DISPLAY-HINT  "1d.2dx2x"
  STATUS  current
  DESCRIPTION
  "The MessageIDCode consists of those parameters required to define a
    message within a dmsMessageTable. It is defined as an OCTET STRING
    containing the OER-encoding of the following ASN.1 structure:
      MessageIDCodeStructure ::= SEQUENCE
      {
      messageMemoryType INTEGER (0..255),
      messageNumber      INTEGER (0..65535),
      messageCRC         OCTET STRING (SIZE (2))
      }

    messageMemoryType (8 bits) shall indicate the dmsMessageMemoryType of
    the desired message. The vaule of 'currentBuffer' (5) in this field
    is discouraged unless the object type using this syntax indicates its
    support and interpretation. For example, the dmsShortPowerRecoveryMessage
    object type indicates that the value of 'currentBuffer' can be used to
    allow a message that was running prior to a power loss to run after the
    power loss without changing the contents of dmsShortPowerRecoveryMessage
    every time the activateMessage is changed

    messageNumber (16 bits) shall indicate the dmsMessageNumber of the
    desired message. In the case of a blank message, this field has an
    upper limit of 255; in the case of the scheduler or current buffer,
    this field shall always be one (1).

    messageCRC (16 bits) shall indicate the dmsMessageCRC of the
    desired message. In the case of a blank message, the scheduler,
    or the current buffer, this value shall be 0x0000.

    For example, given the MULTI string '[jp3]TEST [fl]Flashing[/fl]',
    stored in volatile memory slot 5 with no physical beacons and no pixel
    service, the message ID Code is '04 00 05 95 F9', which would be
    displayed as '4.5x95f', according to the
    display hint. "
  SYNTAX  OCTET STRING (SIZE(5))



MessageActivationCode ::= TEXTUAL-CONVENTION
  DISPLAY-HINT  "2dm1dp1d.2d#2x<1d.1d.1d.1d"
  STATUS  deprecated
  DESCRIPTION
  "The MessageActivationCode consists of those parameters required to
    activate a message on a DMS. It is defined as an OCTET STRING
    containing the OER-encoding of the following ASN.1 structure.

      MessageActivationCodeStructure ::= SEQUENCE
      {
      duration            INTEGER (0..65535),
      activatePriority    INTEGER (0..255),
      messageMemoryType   INTEGER (0..255),
      messageNumber       INTEGER (0..65535),
      messageCRC          OCTET STRING (SIZE (2)),
      sourceAddress       OCTET STRING (SIZE (4))
      }

    duration (16 bits) shall indicate the maximum amount of time, in
    minutes, the message may be displayed prior to activating the
    dmsDefaultEndDurationMessage. dmsMessageTimeRemaining shall be set to
    this value upon successful display of the indicated message. A
    value of 65535 shall indicate an infinite duration.

    activatePriority (8 bits) shall indicate the Activation Priority of
    the message. If this value is greater than or equal to the
    dmsMessageRunTimePriority of the currently displayed message, the new
    message shall be displayed unless errors are detected.

    messageMemoryType (8 bits) shall indicate the dmsMessageMemoryType of
    the desired message. The vaule of 'currentBuffer' (5) in this field
    shall be considered an error.

    messageNumber (16 bits) shall indicate the dmsMessageNumber of the
    desired message. In the case of a blank message, this field has an
    upper limit of 255; in the case of the scheduler, this field shall
    always be one (1).

    messageCRC (16 bits) shall indicate the dmsMessageCRC of the
    desired message.

    sourceAddress (32 bits) shall indicate the 4-byte IP address of the
    device which requested the activation.

    For example, given the MULTI string '[jp3]TEST [fl]Flashing[/fl]',
    stored in volatile memory slot 5 with no physical beacons and no pixel
    service, the message ID Code is '04 00 05 95 F9'. If this message
    is to be displayed for 267 minutes with activation priority 55 from
    IP address 103.8.9.10, the message Activation Code is
    '01 0B 37 04 00 05 95 F9 67 08 09 0A', which would be displayed as
    '267m55p4.5x95f9<103.8.9.10', according to the display hint.


    The dmsActivateMsgError object shall be used to indicate the success
    or failure of activating any message requested by an object with a
    SYNTAX of MessageActivationCode.

    This textual convention was deprecated in NTCIP 1203 v04 and superseded
    with MessageActivationCodeV4, which removes the source address since this
    can now be determined by the lower layer protocols (i.e., the NTCIP 2201
    profile is no longer allowed and the migration allows IPv6)."
  SYNTAX  OCTET STRING (SIZE(12))

MessageActivationCodeV4 ::= TEXTUAL-CONVENTION
  DISPLAY-HINT  "2dm1dp1d.2d#2x"
  STATUS  current
  DESCRIPTION
  "The MessageActivationCodeV4 consists of those parameters required to
    activate a message on a DMS. It is defined as an OCTET STRING
    containing the OER-encoding of the following ASN.1 structure.

      MessageActivationCodeStructureV4 ::= SEQUENCE
      {
      duration            INTEGER (0..65535),
      activatePriority    INTEGER (0..255),
      messageMemoryType   INTEGER (0..255),
      messageNumber       INTEGER (0..65535),
      messageCRC          OCTET STRING (SIZE (2))
      }

    duration (16 bits) shall indicate the maximum amount of time, in
    minutes, the message may be displayed prior to activating the
    dmsDefaultEndDurationMessage. dmsMessageTimeRemaining shall be set to
    this value upon successful display of the indicated message. A
    value of 65535 shall indicate an infinite duration.

    activatePriority (8 bits) shall indicate the Activation Priority of
    the message. If this value is greater than or equal to the
    dmsMessageRunTimePriority of the currently displayed message, the new
    message shall be displayed unless errors are detected.

    messageMemoryType (8 bits) shall indicate the dmsMessageMemoryType of
    the desired message. The vaule of 'currentBuffer' (5) in this field
    shall be considered an error.

    messageNumber (16 bits) shall indicate the dmsMessageNumber of the
    desired message. In the case of a blank message, this field has an
    upper limit of 255; in the case of the scheduler, this field shall
    always be one (1).

    messageCRC (16 bits) shall indicate the dmsMessageCRC of the
    desired message.

    The source address, which was previously included within the
    MessageActivationCode structure used in NTCIP 1203 v01 through v03, is
    no longer contained within this structure. Instead, the device shall
    determine the source address and security name from the lower layers
    of the request containing this structure.

    For example, given the MULTI string '[jp3]TEST [fl]Flashing[/fl]',
    stored in volatile memory slot 5 with no physical beacons and no pixel
    service, the message ID Code is '04 00 05 95 F9'. If this message
    is to be displayed for 267 minutes with activation priority 55 from
    UDP/IPv4 address of 103.8.9.10 on port 161 and a securityName of 'TMC
    operator', the activation code is '01 0B 37 04 00 05 95 F9', which would
    be displayed as '267m55p4.5x95f9', according to the display hint. Upon
    successful display, the field device would update
    dmsMsgRequesterAddressType to be udpIpv4 (1), dmsMsgRequesterAddress to
    be '67 08 09 0A 00 A1' and dmsMsgRequesterSecurityName to 'TMC operator'.


    The dmsActivateMsgError object shall be used to indicate the success
    or failure of activating any message requested by an object with a
    SYNTAX of MessageActivationCodeV4."
  SYNTAX  OCTET STRING (SIZE(8))

MultiString ::= TEXTUAL-CONVENTION
DISPLAY-HINT  "2048t"
STATUS        current
DESCRIPTION
"A string of zero (0) to dmsMaxMultiStringLength octets representing a
message for display on the sign face:
- written in MULTI-language as defined in Section 6,
- encoded as defined by defaultCharacterSet, and
- subranged by the restrictions defined by
  - the dmsMaxMultiStringLength,
  - the dmsSupportedMultiTagsV5, and
  - the values of defaultCharacterSet supported, and
  - the level of support for utf8Style encoding supported.
The value is not allowed to have any null character."
SYNTAX  OCTET STRING

DmsColorScheme ::= TEXTUAL-CONVENTION
STATUS        current
DESCRIPTION
"<Definition> An indication of a color scheme. The values are defined as:
  monochrome1bit (1): Only two states are available: on (1) and off (0).
  monochrome8bit (2): This color palette supports up to 256 shades ranging
      from 0 (off) to 255 (full intensity). Values between zero and 255 are
      scaled to the nearest intensity level supported. Implementations are
      not required to support true 8-bit (256 shade) capabilities.
  colorClassic (3): Color is indicated using the DMSClassicColor textual
      convention.
  color24bit (4): The color is defined by three octet, one each for red,
      green, and blue. Each color value for each octet ranges from 0 (off) to
      255 (full intensity). The combination of the red, green, and blue
      colors supports 16,777,216 colors. Values between zero and 255 are
      scaled to the nearest intensity level supported. Implementations are
      not required to support true 8-bit (256 shade) capability and can
      support a small subrange. For example, a physical beacon might only
support red
      or amber; the 24 bit value is rounded to the closest match that the
      equipment suports. It is the responsibility of the hardware
      specification to identify the hardware capabilities."
SYNTAX      INTEGER {
                 monochrome1bit (1),
                 monochrome8bit (2),
                 colorClassic (3),
                 color24bit(4)}


DmsClassicColor ::= TEXTUAL-CONVENTION
STATUS        current
DESCRIPTION
"An enumerated color code used by implementations that use the 'colorClassic'
value of dmsColorScheme. Each color code is to be mapped to a specific
similar color supported by the sign. If a color code is not mapped to a
supported color (e.g., the sign does not support an enumerated color), the
associated enumerated value shall not be supported by the sign. "
SYNTAX    INTEGER { black (0),
                    red (1),
                    yellow (2),
                    green(3),
                    cyan (4),
                    blue (5),
                    magenta (6),
                    white (7),
                    orange (8),
                    amber (9) }

DmsColor ::= TEXTUAL-CONVENTION
```

```
DISPLAY-HINT  "1dR1dG1dB"
STATUS        current
DESCRIPTION
"A color code interpreted based on the value of the dmsColorScheme object,
which is a read-only object that should not change in an implementation.

If dmsColorScheme is equal to 'monochrome1bit', the DmsColor shall be a one-
octet value of either off (0x00) or on (0x01). The actual color shall be
indicated by the monochromeColor object.

If dmsColorScheme is equal to 'monochrome8bit', the DmsColor shall be a one-
octet value representing a value as defined by DmsClassicColor intensity of
the single color on a scale of 0-255. The actual color shall be indicated by
the monochromeColor object.

If dmsColorScheme is equal to 'colorClassic', the DmsColor shall be a one-
octet value representing a value as defined by DmsClassicColor.

If dmsColorScheme is equal to 'color24bit', the DmsColor shall be a three-
octet RGB color scheme value. The first octet shall represent the intensity
of red, the second octet shall represent the intensity of green, and the
third octet shall represent the intensity of blue, each on a scale of 0-255."
  SYNTAX  OCTET STRING (SIZE(1 | 3))
```

5.2 Sign Configuration and Capability Objects

```
dmsSignCfg  OBJECT-IDENTITY
STATUS  current
DESCRIPTION
"This node is an identifier used to group all objects for DMS sign
configurations that are common to all DMS devices.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1"
::= { dms 1 }
```

5.2.1 Sign Access Parameter

```
dmsSignAccess  OBJECT-TYPE
SYNTAX      NtcipIntegerBitmap8
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the access method to the sign. Methods that are
defined are:
<Format>
  Bit 0- Other
  Bit 1- Walk-in access
  Bit 2- Rear access
  Bit 3- Front access
If a bit is set to one (1), then the associated feature exists; if the bit is
set to zero (0), then the associated feature does not exist.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.1"
::= { dmsSignCfg 1 }
```

5.2.2 Sign Type Parameter

```
dmsSignType  OBJECT-TYPE
SYNTAX      INTEGER{
                 other (1),
                 bos (2),
                 cms (3),
                 vmsChar (4),
                 vmsLine (5),
                 vmsFull (6),
                 portableOther (129),
                 portableBOS (130),
                 portableCMS (131),
                 portableVMSChar (132),
                 portableVMSLine (133),
                 portableVMSFull (134)}
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the type of sign.
<Format>The descriptions are:
  other:  Device not specified through any other definition, refer to
     device manual,
  bos: Device is a Blank-Out Sign,
  cms : Device is a Changeable Message Sign,
  vmsChar : Device is a Variable Message Sign with character matrix
     setup,
  vmsLine :  Device is a Variable Message Sign with line matrix setup,
  vmsFull: Device is a Variable Message Sign with full matrix setup.
  Same is true for all portable signs.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.2"
::= { dmsSignCfg 2 }
```

5.2.3 Sign Height Parameter

```
dmsSignHeight  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "millimeters"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the sign height in millimeters including the border
(dmsVerticalBorder).
<Parameter Type> Status<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.3"
::= { dmsSignCfg 3 }
```

5.2.4 Sign Width Parameter

```
dmsSignWidth  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "millimeters"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the sign width in millimeters including the border
(dmsHorizontalBorder).
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.4"
::= { dmsSignCfg 4 }
```

5.2.5 Horizontal Border Parameter

```
dmsHorizontalBorder  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "millimeters"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the minimum border distance, in millimeters, that
exists on the left and right sides of the sign.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.5"
::= { dmsSignCfg 5 }
```

5.2.6 Vertical Border Parameter

```
dmsVerticalBorder  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "millimeters"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the minimum border distance, in millimeters, that
exists on the top and bottom of the sign.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.6"
::= { dmsSignCfg 6 }
```

5.2.7 Legend Parameter

```
dmsLegend  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1), -- deprecated
                 noLegend (2),
                 legendExists (3)}
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates if a Legend is shown on the sign.
<Format> In v02, the enumerated value of 'other' was retired to improve
interoperability and since this is a logical Boolean value with no real other
alternative possible.
<Informative> The value of other was deprecated in v02 to improve
interoperability.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.7"
::= { dmsSignCfg 7 }
```

5.2.8 Beacon Type Parameter

```
dmsBeaconType  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1),
                 none (2),
                 oneBeacon (3),
                 twoBeaconSyncFlash (4),
                 twoBeaconsOppFlash (5),
                 fourBeaconSyncFlash (6),
                 fourBeaconAltRowFlash (7),
                 fourBeaconAltColumnFlash (8),
                 fourBeaconAltDiagonalFlash (9),
                 fourBeaconNoSyncFlash (10),
                 oneBeaconStrobe (11),
                 twoBeaconStrobe (12),
                 fourBeaconStrobe (13)}
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the configuration of the type, numbers and flashing
patterns of beacons on a sign.
<Format> The definitions are:
  other: other types, numbers and patterns of beacons attached to the
     sign display.
  none: no beacons attached to the sign display
  oneBeacon: one flashing beacon
  twoBeaconSyncFlash: two beacons, synchronized flashing
  twoBeaconsOppFlash: two beacons, opposing flashing
  fourBeaconSyncFlash:  four beacons, synchronized flashing
  fourBeaconAltRowFlash: four beacons, alternate row flashing
  fourBeaconAltColumnFlash: four beacons, alternate column flashing
  fourBeaconAltDiagonalFlash: four beacons, alternate diagonal
     flashing
  fourBeaconNoSyncFlash: four beacons, no synchronized flashing
  oneBeaconStrobe: one beacon, strobe light
  twoBeaconStrobe: two beacons, strobe light
  fourBeaconStrobe: four beacons, strobe light
<Superseded by> dmsBeaconTable and dmsBeaconPlanTable
<Informative> This object was deprecated in v05 in favor of a more robust
beacon design.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.8"
::= { dmsSignCfg 8 }
```

5.2.9 Sign Technology Parameter

```
dmsSignTechnology  OBJECT-TYPE
SYNTAX      NtcipIntegerBitmap16
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the utilized technology in a bitmap format  (Hybrids
will have to set the bits for all technologies that the sign utilizes).
<Format>
  Bit 0- Other,
  Bit 1- LED,
  Bit 2- Flip Disk,
  Bit 3- Fiber Optics,
  Bit 4- Shuttered,
  Bit 5- Bulb,
  Bit 6- Drum
  Bits 7-15- Reserved

If a bit is set to one (1), then the associated feature exists; if the
bit is set to zero (0), then the associated feature does not exist.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.9"
::= { dmsSignCfg 9 }
```

5.2.10 Sign Display Latitude

```
dmsDisplayLatitude  OBJECT-TYPE
SYNTAX      ITSLatitude84
UNITS       "microdegrees"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the latitude of the sign display as determined by the
sum of fdConfiguredLatitude (which identifies the location of the GNSS
antenna or similar reference point) and dmsDisplayOffsetLatitude.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.10"
::= { dmsSignCfg 10 }
```

5.2.11 Sign Display Longitude

```
dmsDisplayLongitude  OBJECT-TYPE
SYNTAX      ITSLongitude84
UNITS       "microdegrees"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the longitude of the sign display as determined by
the sum of fdConfiguredLongitude (which identifies the location of the GNSS
antenna or similar reference point) and dmsDisplayOffsetLongitude.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.11"
::= { dmsSignCfg 11 }
```

5.2.12 Sign Display Direction

```
dmsDisplayDirection  OBJECT-TYPE
SYNTAX      ITSDirection
UNITS       "degrees"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the direction that the sign faces. A value of 0
indicates a northward facing sign providing messages for southbound traffic.
A value of 90 indictes an eastward facing sign providing messages for
westbound traffic.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.12"
::= { dmsSignCfg 12 }
```

5.2.13 Sign Display Offset Latitude

```
dmsDisplayOffsetLatitude  OBJECT-TYPE
SYNTAX      ITSLatitude84
UNITS       " microdegrees"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the difference in latitude between the GNSS antenna
location (or other point upon which fdConfiguredLongitude is based) and the
center of the sign display.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.13"
::= { dmsSignCfg 13 }
```

5.2.14 Sign Display Offset Longitude

```
dmsDisplayOffsetLongitude  OBJECT-TYPE
SYNTAX      ITSLongitude84
UNITS       "microdegrees"
MAX-ACCESS  read-write
STATUS  current
DESCRIPTION
"<Definition> Indicates the difference in longitude between the GNSS antenna
location (or other point upon which fdConfiguredLongitude is based) and the
center of the sign display.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.14"
::= { dmsSignCfg 14 }
```

5.2.15 Beacon Operation Objects

```
dmsBeacon  OBJECT-IDENTITY
STATUS  current
DESCRIPTION
"<Definition> A node used to provide information about physical beacons
associated with the sign.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15"
::= { dmsSignCfg 15 }
```

5.2.15.1 Number of Beacons

```
dmsBeaconCount  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "beacons"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the number of physical beacons managed by the DMS.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.1"
::= { dmsBeacon 1 }
```

5.2.15.2 Beacon Table

```
dmsBeaconTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF DmsBeaconEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"<Definition> A table containing the information needed to identify the
physical beacons associated with the DMS.
<Table Type> static
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.2"
::= { dmsBeacon 2 }

dmsBeaconEntry  OBJECT-TYPE
SYNTAX      DmsBeaconEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"<Definition> A row within the dmsBeaconTable, which contains the information
needed to identify the physical beacons associated with the DMS.
<Table Type> static
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.2.1"
INDEX {dmsBeaconIndex}
```

```
::= { dmsBeaconTable 1 }

DmsBeaconEntry ::= SEQUENCE {
      dmsBeaconIndex            Integer32,
      dmsBeaconXLocation        Integer32,
      dmsBeaconYLocation        Integer32,
      dmsBeaconWidth            Integer32,
      dmsBeaconHeight           Integer32,
      dmsBeaconColorScheme      DmsColorScheme,
      dmsBeaconFlashResolution  Integer32 }
```

5.2.15.2.1 Beacon Index Parameter

```
dmsBeaconIndex  OBJECT-TYPE
SYNTAX      Integer32 (1..255)
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"<Definition> Sequential index of the beacon starting with beacon 1.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.2.1.1"
::= { dmsBeaconEntry 1 }
```

5.2.15.2.2 Beacon X Location Parameter

```
dmsBeaconXLocation  OBJECT-TYPE
SYNTAX      Integer32
UNITS       "millimeters"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> The horizontal distance from the left-most point of the sign
border to the center of the beacon. A positive value shall indicate a
location to the right; a negative value shall indicate a location to the
left.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.2.1.2"
::= { dmsBeaconEntry 2 }
```

5.2.15.2.3 Beacon Y Location Parameter

```
dmsBeaconYLocation  OBJECT-TYPE
SYNTAX      Integer32
UNITS       "millimeters"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> The vertical distance from the lowest point of the sign border
to the center of the beacon. A positive value shall indicate a location above
the lowest point of the border; a negative value shall indicate a location
below.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.2.1.3"
::= { dmsBeaconEntry 3 }
```

5.2.15.2.4 Beacon Width Parameter

```
dmsBeaconWidth  OBJECT-TYPE
SYNTAX      Integer32
UNITS       "millimeters"
MAX-ACCESS  read-only
STATUS      current
```

```
DESCRIPTION
"<Definition> The width of the illumination area of the beacon. Circular and
square beacons shall have identical values for dmsBeaconWidth and
dmsBeaconHeight.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.2.1.4"
::= { dmsBeaconEntry 4 }
```

5.2.15.2.5 Beacon Height Parameter

```
dmsBeaconHeight  OBJECT-TYPE
SYNTAX      Integer32
UNITS       "millimeters"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> The height of the illumination area of the beacon. Circular and
square beacons shall have identical values for dmsBeaconWidth and
dmsBeaconHeight.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.2.1.5"
::= { dmsBeaconEntry 5 }
```

5.2.15.2.6 Beacon Color Scheme Parameter

```
dmsBeaconColorScheme  OBJECT-TYPE
SYNTAX      DmsColorScheme
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the color scheme supported by the beacon.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.2.1.6"
::= { dmsBeaconEntry 6 }
```

5.2.15.2.7 Beacon Flash Resolution Parameter

```
dmsBeaconFlashResolution  OBJECT-TYPE
SYNTAX      Integer32
UNITS       "milliseconds"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the resolution of the flash duration control. For
example, a value of 500 would indicate that the beacon flash parameters can
control the flash (on) period and flash cycle in steps of 500 ms (i.e., 0.5
s).
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.2.1.7"
::= { dmsBeaconEntry 7 }
```

5.2.15.3 Beacon Plan Table

```
dmsBeaconPlanTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF DmsBeaconPlanEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"<Definition> A table containing the operational plan information used to
control a set of physical beacons associated with the DMS.
<Table Type> dynamic
```

```
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.3"
::= { dmsBeacon 3 }

dmsBeaconPlanEntry  OBJECT-TYPE
SYNTAX      DmsBeaconPlanEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"<Definition> A row within the dmsBeaconPlanTable, which contains the
operational plan information used to control a set of physical beacons
associated with the DMS.
<Table Type> dynamic
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.3.1"
INDEX {dmsBeaconPlanIndex}
::= { dmsBeaconPlanTable 1 }

DmsBeaconPlanEntry ::= SEQUENCE {
      dmsBeaconPlanIndex        Integer32,
      dmsBeaconPlanDescr        SnmpAdminString,
      dmsBeaconPlanCycle        Unsigned32,
      dmsBeaconPlanStorageType  StorageType,
      dmsBeaconPlanRowStatus    RowStatus }
```

5.2.15.3.1 Beacon Plan Index Parameter

```
dmsBeaconPlanIndex  OBJECT-TYPE
SYNTAX      Integer32 (1..255)
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"<Definition> Index of the beacon plan.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.3.1.1"
::= { dmsBeaconPlanEntry 1 }
```

5.2.15.3.2 Beacon Plan Description Parameter

```
dmsBeaconPlanDescr  OBJECT-TYPE
SYNTAX      SnmpAdminString
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
"<Definition> Textual description of the beacon plan.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.3.1.2"
::= { dmsBeaconPlanEntry 2 }
```

5.2.15.3.3 Beacon Plan Cycle Parameter

```
dmsBeaconPlanCycle  OBJECT-TYPE
SYNTAX      Unsigned32 (0..60000)
UNITS       "milliseconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
"<Definition> The duration required to complete one entire flash cycle. The
flash cycle zero-point shall be based on the previously occuring local
midnight, if the local clock feature is supported, and midnight UTC if not.
The plan cycle shall be a multiple of dmsBeaconFlashSupport.
<Infomative> Longer flash cycles allow lights to flash in order across
multiple beacons. Coordinating the flash cycles at local midnight allows
```

```
coordinated flashing across multiple devices. It is recommended that the
cycle length is a denominator of 86,400,000 to prevent cycle anomalies at
midnight on a regular basis. Ideally, the cycle length should be a
denominator of 1,000 to prevent anomailes under nearly all conditions (e.g.,
leap seconds).
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.3.1.3"
DEFVAL { 1000 }
::= { dmsBeaconPlanEntry 3 }
```

5.2.15.3.4 Beacon Storage Type Parameter

```
dmsBeaconPlanStorageType  OBJECT-TYPE
SYNTAX      StorageType
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
"<Definition> This object defines whether this beacon plan and all of its
component details are kept in volatile storage and lost upon reboot or if
this row is stored in non-volatile or permanent storage. Conceptual rows
having the value 'permanent' shall allow write access to
dmsBeaconPlanRowStatus (i.e., to change to either 'actuve' or
'notInService').
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.3.1.4"
::= { dmsBeaconPlanEntry 4 }
```

5.2.15.3.5 Beacon Row Status Parameter

```
dmsBeaconPlanRowStatus  OBJECT-TYPE
SYNTAX      RowStatus
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
"<Definition> The status of this beacon plan. A 'destroy' command for this
object shall also destroy all dmsBeaconPlanDetailEntries associated with this
plan.

Only this object can be modified when the value of this object is 'active'.

This object shall only have a value of 'active' when its dmsBeaconPlanCycle
is divisible by all associated dmsBeaconPlanDetailCycle objects.
Implementations shall support both createAndGo and createAndWait.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.3.1.5"
::= { dmsBeaconPlanEntry 5 }
```

5.2.15.4 Beacon Plan Detail Table

```
dmsBeaconPlanDetailTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF DmsBeaconPlanDetailEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"<Definition> A table containing the operational details used to control a
single physical beacon as a part of a beacon plan.
<Table Type> dynamic
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.4"
::= { dmsBeacon 4 }
```

```
dmsBeaconPlanDetailEntry  OBJECT-TYPE
SYNTAX      DmsBeaconPlanDetailEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"<Definition> A row within the dmsBeaconPlanDetailTable, which contains the
operational details used to control a single physical beacon as a part of a
beacon plan.
<Table Type> dynamic
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.4.1"
INDEX {dmsBeaconPlanIndex, dmsBeaconIndex }
::= { dmsBeaconPlanDetailTable 1 }

DmsBeaconPlanDetailEntry ::= SEQUENCE {
      dmsBeaconPlanDetailCycle   Unsigned32,
      dmsBeaconPlanDetailOffset  Unsigned32,
      dmsBeaconPlanDetailFlash   Unsigned32,
      dmsBeaconPlanDetailColor   DmsColor }
```

5.2.15.4.1 Beacon Plan Detail Cycle Parameter

```
dmsBeaconPlanDetailCycle  OBJECT-TYPE
SYNTAX      Unsigned32 (0..60000)
UNITS       "milliseconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
"<Definition> The duration required to complete one entire flash cycle for
the specific beacon. The flash cycle zero-point shall be based on the
previously occuring local midnight, if the local clock feature is supported,
and midnight UTC if not. The value of dmsBeaconPlanCycle shall be divisible
by this value prior to dmsBeaconPlanRowStatus becoming active. The flash
cycle shall be a multiple of dmsBeaconFlashSupport.
<Infomative> By setting the dmsBeaconPlanDetailCycle to be half of the
dmsBeaconPlanCycle, a beacon can flash twice during one plan cycle.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.4.1.1"
DEFVAL { 1000 }
::= { dmsBeaconPlanDetailEntry 1 }
```

5.2.15.4.2 Beacon Plan Detail Offset Parameter

```
dmsBeaconPlanDetailOffset  OBJECT-TYPE
SYNTAX      Unsigned32 (0..60000)
UNITS       "milliseconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
"<Definition> The point within the dmsBeaconPlanDetailCycle period at which
the beacon illuminates and begins its flash period. The offset shall be less
than dmsBeaconPlanDetailCycle and and a multiple of dmsBeaconFlashSupport
prior to dmsBeaconPlanRowStatus becoming active.
<Infomative> An alternate flash pattern can be configured by setting one
beacon to have a zero offset and a second beacon to have a offset that is
equal to one half of the cycle length used for the two beacons.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.4.1.2"
DEFVAL { 0 }
::= { dmsBeaconPlanDetailEntry 2 }
```

5.2.15.4.3 Beacon Plan Detail Flash Parameter

```
dmsBeaconPlanDetailFlash  OBJECT-TYPE
SYNTAX      Unsigned32 (0..60000)
UNITS       "milliseconds"
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
"<Definition> The duration that the beacon will be illuminated when it
flashes (i.e., the 'flash period'). This value shall be a multiple of
dmsBeaconFlashSupport.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.4.1.3"
DEFVAL { 500 }
::= { dmsBeaconPlanDetailEntry 3 }
```

5.2.15.4.4 Beacon Plan Detail Color Parameter

```
dmsBeaconPlanDetailColor  OBJECT-TYPE
SYNTAX      DmsColor
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
"<Definition> The color of the beacon flash.

If the dmsBeaconColorScheme is monochrome1bit, the value can only be 1. If
the dmsBeaconColorScheme is monochrome8bit, the DMS shall accept any single
octet value and will transpose the value to the closest value that the beacon
supports (i.e., the beacon does not have to support all 255 levels of
illumination). If dmsBeaconColorScheme is colorClassic, the DMS shall only
accept values supported by the DMS. If the value is color24bit, the DMS shall
accept any three-octet value and will transpose the value to the closest
value that the beacon supports (i.e., the beacon does not have to support all
possible color combinations).
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.15.4.1.4"
::= { dmsBeaconPlanDetailEntry 4 }
```

5.3 VMS Configuration Objects

```
vmsCfg  OBJECT-IDENTITY
STATUS  current
DESCRIPTION
"This subnode is an identifier used to group all objects for support of VMS
sign configurations that are common to all VMS devices.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.2"
::= { dms 2 }
```

5.3.1 Character Height in Pixels Parameter

```
vmsCharacterHeightPixels  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "pixels"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the height of a single character in pixels. The value
zero (0) indicates a variable character height, which implies a full-matrix
sign.
<Parameter Type> Status
```

```
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.2.1"
::= { vmsCfg 1 }
```

5.3.2 Character Width in Pixels Parameter

```
vmsCharacterWidthPixels  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "pixels"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the width of a single character in Pixels. The value
zero (0) indicates a variable character width, which implies either a full-
matrix or line-matrix sign.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.2.2"
::= { vmsCfg 2 }
```

5.3.3 Sign Height in Pixels Parameter

```
vmsSignHeightPixels  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "pixels"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the number of rows of pixels for the entire sign.
<Informative> To determine the number of lines for a line matrix or character
matrix sign, divide the vmsSignHeightPixels object value by the
vmsCharacterHeightPixels object value. This should result in a whole number,
the number of lines in the sign.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.2.3"
::= { vmsCfg 3 }
```

5.3.4 Sign Width in Pixels Parameter

```
vmsSignWidthPixels  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "pixels"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the number of columns of pixels for the entire sign.
<Informative> To determine the number of characters per line for a character
matrix sign, divide the vmsSignWidthPixels object value by the
vmsCharacterWidthPixels object value.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.2.4"
::= { vmsCfg 4 }
```

5.3.5 Horizontal Pitch Parameter

```
vmsHorizontalPitch  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "millimeters"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the horizontal distance from the center of one pixel
to the center of the neighboring pixel in millimeters. The horizontal pitch
```

```
on a character matrix DMS does not apply to the spacing between characters
but does apply to the distance between pixels within a character.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.2.5"
::= { vmsCfg 5 }
```

5.3.6 Vertical Pitch Parameter

```
vmsVerticalPitch  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "millimeters"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the vertical distance from the center of one pixel to
the center of the neighboring pixel in millimeters. The vertical pitch on a
line matrix DMS does not apply to the spacing between lines but does apply to
the distance between pixels within a line. The vertical pitch on a character
matrix DMS does not apply to the spacing between characters but does apply to
the distance between pixels within a character.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.2.6"
::= { vmsCfg 6 }
```

5.3.7 Monochrome Color Parameter

```
monochromeColor  OBJECT-TYPE
SYNTAX      NtcipTwoColors
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the color supported by a monochrome sign. If the
'monochrome1Bit' or 'monochrome8Bit' scheme is used, then this object will
contain six octets. The first 3 octets shall, in this order, indicate the
red, green, and blue component values of the color when the pixels are turned
'ON'. The last 3 octets shall, in this order, indicate the red, green, and
blue component values of the color when the pixels are turned 'OFF'. If the
sign is a non-monochrome sign, then the value of this object shall be an
octet string of six zeros (0x00 0x00 0x00 0x00 0x00 0x00).
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.2.7"
::= { vmsCfg 7 }
```

5.4 Font Definition Objects

```
fontDefinition  OBJECT-IDENTITY
STATUS  current
DESCRIPTION
"This node is an identifier used to group all objects for DMS font
configurations that are common to DMS devices.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3"
::= { dms 3 }
```

5.4.1 Number of Fonts Parameter

```
numFonts  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "fonts"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
```

```
"<Definition> Indicates the maximum number of fonts that the sign can store.
See the Specification in association with the supplemental requirements for
fonts to determine the number of fonts that the DMS must support.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.3.1"
::= { fontDefinition 1 }
```

5.4.2 Font Table Parameter

```
fontTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF FontEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "
<Definition> A table containing the information needed to configure/define a
particular font. Changing an object in a font or character table while the
font is used by a displayed message yields unpredictable results.
<Informative>  The DMS WG recognizes that the message display on the sign
could be unpredictable during the download of a font when using the unmanaged
state (V1 compatibility). Those specifying authorities or application
developers who are sensitive to this issue can blank the display during a
font download.
<Table Type> static
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.3.2"
::= {fontDefinition 2}

fontEntry OBJECT-TYPE
SYNTAX      FontEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "<Definition> Parameters of the Font Table."
INDEX {fontIndex}
::= {fontTable 1}

FontEntry ::= SEQUENCE{
   fontIndex       Integer32,
   fontNumber      Integer32,
   fontName        SnmpAdminString,
   fontHeight      Integer32,
   fontCharSpacing Integer32,
   fontLineSpacing Integer32,
   fontVersionID   Integer32,
   fontStatus      INTEGER
   }
```

5.4.2.1 Font Index Parameter

```
fontIndex  OBJECT-TYPE
SYNTAX      Integer32 (1..255)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the row number of the entry. An implementation shall
support exactly the number of rows defined by nomFonts, startng at 1 and be
sequential.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.3.2.1.1"
::= { fontEntry 1 }
```

5.4.2.2 Font Number Parameter

```
fontNumber  OBJECT-TYPE
SYNTAX      Integer32 (1..255)
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> A unique, user-specified number for a particular font which can
be different from the value of the fontIndex-object. This is the number
referenced by MULTI when specifying a particular font.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.3.2.1.2"
::= { fontEntry 2 }
```

5.4.2.3 Font Name Parameter

```
fontName  OBJECT-TYPE
SYNTAX      SnmpAdminString (SIZE (0..64))
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the name of the font.
<Informative> Prior to NTCIP 1203 v04, this object type only supported NVT-
ASCII characters. New implementations should specify their supported values
for this object type in their AGENT-CAPABILITIES statement. Attempting to
store a value that includes characters not supported by an implementation can
result in an error response.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.3.2.1.3"
::= { fontEntry 3 }
```

5.4.2.4 Font Height Parameter

```
fontHeight  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "pixels"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the height of the font in pixels. Changing the value
of this object invalidates this fontTable row, sets all corresponding
characterWidth objects to zero (0), and sets all corresponding
characterBitmap objects to zero length. Character Matrix and Line Matrix VMS
shall subrange this object either to a value of zero (0) or the value of
vmsCharacterHeightPixels; a Full Matrix VMS shall subrange this object to the
range of zero (0) to the value of vmsSignHeightPixels or 255, whichever is
less.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.3.2.1.4"
::= { fontEntry 4 }
```

5.4.2.5 Font Character Spacing Parameter

```
fontCharSpacing  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "pixels"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the default horizontal spacing (in pixels) between
each of the characters within the font.  If the font changes on a line, then
```

```
the average character spacing of the two fonts, rounded up to the nearest
whole pixel, shall be used between the two characters where the font changes.
Character Matrix VMS shall ignore the value of this object; Line Matrix and
Full Matrix VMS shall subrange this object to the range of zero (0) to the
smaller of 255 or the value of vmsSignWidthPixels.
See also the MULTI tag 'spacing character [sc]'.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.3.2.1.5"
::= { fontEntry 5 }
```

5.4.2.6 Font Line Spacing Parameter

```
fontLineSpacing  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "pixels"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the default vertical spacing (in pixels) between each
of the lines within the font for Full Matrix VMS. The line spacing for a line
is the largest font line spacing of all fonts used on that line. The number
of pixels between adjacent lines is the average of the 2 line spacings of
each line, rounded up to the nearest whole pixel. Character Matrix VMS and
Line Matrix VMS shall ignore the value of this object; Full Matrix VMS shall
subrange this object to the range of zero (0) to the smaller of 255 or the
value of vmsSignHeightPixels.
See also the MULTI tag 'new line [nl]'.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.3.2.1.6"
::= { fontEntry 6 }
```

5.4.2.7 Font Version ID Parameter

```
fontVersionID  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Each font that has been downloaded to a sign shall have a
relatively unique ID. This ID shall be calculated using the CRC-16 algorithm
defined in ISO 3309 and the associated OER-encoded (as defined in ITU-T
X.696) FontVersionByteStream.
The sign shall respond with the version ID value that is valid at the time.

FontVersionByteStream consists of the main font characteristics followed by n
rows of CharacterInfoList, as shown by the following ASN.1 construct:
  FontVersionByteStream ::= SEQUENCE {
        fontInformation    FontInformation,
        characterInfoList  CharacterInfoList }

FontInformation describes the characteristics of the font which are common to
each character and defines the order in which this information appears when
constructing the byte stream which will be used to calculate the CRC. There
is only one row of data for this SEQUENCE for a specific font, as defined by
the following ASN.1 construct:
  FontInformation ::= SEQUENCE {
        fontNumber            INTEGER (1..255),
        fontHeight            INTEGER (0..255),
        fontCharSpacing       INTEGER (0..255),
        fontLineSpacing       INTEGER (0..255) }

CharacterInfoList describes the characteristics of each defined character
(e.g., where characterWidth is greater than 0) for the fontNumber indicated
within the fontInformation field. The CharacterInformation is ordered by the
characterNumber in an increasing format per the following ASN.1 construct:
  CharacterInfoList ::=  SEQUENCE OF CharacterInformation

CharacterInformation describes the characteristics of a single character and
defines the objects and order of the objects within one row of
CharacterInfoList, per the following ASN.1 construct:
  CharacterInformation  SEQUENCE {
      characterNumber         INTEGER (1..65535),
      characterWidth          INTEGER (0..255),
      characterBitmap         OCTET STRING }

Complete definitions for these referenced objects are contained elsewhere in
this document.

The following is an example of developing the FontVersionByteStream value.
Assume the following values for this example, where we only have 2 characters
defined:
fontNumber = 2,
fontHeight = 7,
fontCharSpacing = 1,
fontLineSpacing = 3,
characterWidth.52 = 7,
characterBitmap.52 = 1C 59 34 6F E1 83 00,
characterWidth.65 = 6,
characterBitmap.65 = 7B 3C FF CF 3C C0

The resulting string in hex would be:
FontVersionByteStream = 02 07 01 03 01 02 00 34 07 07 1C 59 34 6F E1 83 00 00
41 06 06 7B 3C FF CF 3C C0

CRC = 0x52ED
fontVersionID = 0xED52

Clarifications:
 a) characterNumber is a two-byte unsigned integer.
 b) characterBitmap is defined as OCTET STRING without a size constraint.
(the length octets shall be present)
 c) CharacterInfoList is defined as SEQUENCE-OF that requires a quantity
field (unconstrained unsigned integer) 'with a value equal to the number of
times the componentType is repeated within the value field'.

The resulting graphic depictions of those 2 defined characters are:
0001110
0010110
0100110
1000110
1111111
0000110
0000110

and
```

```
011110
110011
110011
111111
110011
110011
110011
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.3.2.1.7"
::= { fontEntry 7 }
```

5.4.2.8 Font Status Parameter

```
fontStatus  OBJECT-TYPE
SYNTAX      INTEGER {
                 notUsed (1),
                 modifying (2),
                 calculatingID (3),
                 readyForUse (4),
                 inUse (5),
                 permanent (6),
                 modifyReq (7),
                 readyForUseReq (8),
                 notUsedReq (9),
                 unmanagedReq (10), -- obsolete
                 unmanaged (11)}    -- obsolete
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> This object defines a state machine allowing to manage fonts
stored within a DMS.
<Format> The definitions of the possible values are:
notUsed (1) - a state indicating that this row in this table is currently not
used.
modifying (2) - a state indicating that the objects defined in this row can
be modified.
calculatingID (3) - a state indicating that the  fontVersionID for this row
is currently being calculated.
readyForUse (4) - a state indicating that the font defined in this row can be
used  for message display.
inUse (5) - a state indicating that the font defined in this row is currently
being used for the displayed message.
permanent (6) - a state indicating that the font defined in this row is a
permanent font that cannot be modified. This font is provided by the sign
vendor and can be used for message display.
modifyReq (7) -  command sent to request the transition to the modifying
state..
readyForUseReq (8) -  command sent to request the transition to the
readyForUse state.
notUsedReq (9) -  command sent to request the transition to the notUsed
state.
unmanagedReq (10) -  command sent to request the transition to the unmanaged
state.
unmanaged (11) - a state indicating that the font defined in this row is a
font that is not managed using the fontStatus object. This state can be used
to manage the font as in NTCIP 1203 v1.
<Informative> The fontStatus object was added in version 2; the values of
'unmanagedReq' and 'unmanaged' were provided for backwards compatibility with
```

```
version 1 devices. With the introduction of SNMPv3 in NTCIP 1203 v04,
backwards compatibility is lost and these two values are considered obsolete.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.3.2.1.8"
::= { fontEntry 8 }
```

5.4.3 Maximum Characters per Font Parameter

```
maxFontCharacters  OBJECT-TYPE
SYNTAX      Integer32 (1..65535)
UNITS       "characters"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the maximum number of rows in the character table
that can exist for any given font.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.3.3"
::= { fontDefinition 3 }
```

5.4.4 Character Table Parameter

```
characterTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF CharacterEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "<Definition> A table containing the information needed to
configure/define each character of a particular font.
<Table Type> static
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.3.4"
::= {fontDefinition 4}

characterEntry OBJECT-TYPE
SYNTAX      CharacterEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "<Definition> Parameters of the Character Configuration Table."
INDEX {fontIndex, characterNumber}
::= {characterTable 1}

CharacterEntry ::= SEQUENCE {
   characterNumber   Integer32,
   characterWidth    Integer32,
   characterBitmap   OCTET STRING}
```

5.4.4.1 Character Number Parameter

```
characterNumber  OBJECT-TYPE
SYNTAX      Integer32 (1..65535)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the code point associated with this character of this
font.
<Informative>For example, if the font set followed the ASCII or Unicode
numbering schemes, the character giving the bitmap of 'A' would be
characterNumber 65 and would be encoded as 0x41 in a MULTI string (in either
the eightBit or utf8Style encoding). Likewise, if the font set followed the
Unicode numbering scheme, the character giving the bitmap of the British
pound sign would be character number 163. A MULTI string accessing this
```

```
character would use the code 0xA3 if using eightBit encoding and would use
0xC2A3 if using the utf8Style encoding. Code points above 255 can only be
accessed in the eightBit encoding scheme by using the hexadecimal character
tag (e.g., [hc0100] to display character 256); the UTF8Style is able to
encode all 65,535 potential characters supported by a single font within 3
octets
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.3.4.1.1"
::= { characterEntry 1 }
```

5.4.4.2 Character Width Parameter

```
characterWidth  OBJECT-TYPE
UNITS       "pixels"
SYNTAX      Integer32 (0..255)
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the width of this character in pixels. A width of
zero (0) indicates this row is invalid. A Character Matrix VMS shall subrange
this object either to a value of zero (0) or the value of the
vmsCharacterWidthPixels object; a Line Matrix or Full Matrix VMS shall
subrange this object to a range of zero (0) to vmsSignWidthPixels.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.3.4.1.2"
::= { characterEntry 2 }
```

5.4.4.3 Character Bitmap Parameter

```
characterBitmap  OBJECT-TYPE
SYNTAX      OCTET STRING
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> A bitmap that defines each pixel within a rectangular region as
being either displayed in the foreground color (bit=1) or transparent
(bit=0). If the pixel is transparent, it will remain whatever color existed
in the message before drawing the character. This might be the background
color, a color rectangle (see MULTI tag) or a graphic. The result of this
bitmap is how the character appears on the sign.

The octet string is treated as a binary bit string. The most significant bit
defines the state of the pixel in the upper left corner of the rectangular
region. The rectangular region is processed by rows, left to right, then top
to bottom. The size of the rectangular region is defined by the fontHeight
and characterWidth objects; any remaining bits shall be ignored, except for
use in the calculation of the CRC.

This object shall be subranged by the device to the maximum number of bytes
as indicated by fontMaxCharacterSize.

<Informative>Version 1 Compatibility:  Version 1 of this standard defined the
bits as ON (foreground color) or OFF (background color).
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.3.4.1.3"
::= { characterEntry 3 }
```

5.4.5 Maximum Character Size Parameter

```
fontMaxCharacterSize  OBJECT-TYPE
```

```
UNITS       "octets"
SYNTAX      Integer32 (0..65535)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> An indication of the maximum size, in octets, that the DMS
supports for each character's characterBitmap object.

The largest value of this object must be equal to or smaller than the value
defined by ROUNDUP(total number of pixels of the sign / 8).
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.3.5"
::= { fontDefinition 5 }
```

5.5 Multi-Configuration Objects

```
multiCfg  OBJECT-IDENTITY
STATUS  current
DESCRIPTION
"<Definition> This subnode is an identifier used to group all objects for
support of MULTI language configuration such as all default tag values.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4"
::= { dms 4 }
```

5.5.1 Default Background Color Parameter

```
defaultBackgroundColor  OBJECT-TYPE
SYNTAX      DmsClassicColor
MAX-ACCESS  read-write
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the color of the background shown on the sign for the
'colorClassic' scheme (see the dmsColorScheme object). If a different color
scheme is used, a genErr shall be returned. If a color is requested that is
not supported, then a genErr shall be returned.
<Superseded by> defaultBackgroundRGB
<Informative> This object was deprecated in v04 and should not be used in
native SNMPv3 implementations.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.1"
DEFVAL  { black }
::= { multiCfg 1 }
```

5.5.2 Default Foreground Color Parameter

```
defaultForegroundColor  OBJECT-TYPE
SYNTAX      DmsClassicColor
MAX-ACCESS  read-write
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the color of the foreground (the actual text) shown
on the sign for the 'colorClassic' scheme (see the dmsColorScheme object). If
a different color scheme is used, a genErr shall be returned. If a color is
requested that is not supported, then a genErr shall be returned.
<Superseded by> defaultForegroundRGB
<Informative> This object was deprecated in v04 and should not be used in
native SNMPv3 implementations.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.2"
::= { multiCfg 2 }
```

5.5.3 Default Flash On Time Parameter

```
defaultFlashOn  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "tenths of a second"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the default flash on time, in tenths of a second, for
flashing text. If the time is set to zero (0), the default is NO FLASHing but
the text remains visible. This object may be sub-ranged by an implementation;
see Section 3.5.2.2.2.3 for more information.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.3"
DEFVAL  {5}
::= { multiCfg 3 }
```

5.5.4 Default Flash On Time Parameter at Activation

```
defaultFlashOnActivate  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "tenths of a second"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the value of defaultFlashOn at activation of the
currently active message for the purpose of determining what the value was at
the time of activation. The value shall be created (overwritten) at the time
when the message was copied into the currentBuffer.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.17"
::= { multiCfg 17 }
```

5.5.5 Default Flash Off Time Parameter

```
defaultFlashOff  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "tenths of a second"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the default flash off time, in tenths of a second,
for flashing text. If the time is set to zero (0), the default is NO FLASHing
but the text remains visible. This object may be sub-ranged by an
implementation; see Section 3.5.2.2.2.3 for more information.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.4"
DEFVAL  {5}
::= { multiCfg 4 }
```

5.5.6 Default Flash Off Time Parameter at Activation

```
defaultFlashOffActivate  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "tenths of a second"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the value of defaultFlashOff at activation of the
currently active message for the purpose of determining what the value was at
```

```
the time of activation. The value shall be created (overwritten) at the time
when the message was copied into the currentBuffer.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.18"
::= { multiCfg 18 }
```

5.5.7 Default Font Parameter

```
defaultFont  OBJECT-TYPE
SYNTAX      Integer32 (1..255)
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the default font number (fontNumber-object) for a
message.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.5"
DEFVAL  {1}
::= { multiCfg 5 }
```

5.5.8 Default Font Parameter at Activation

```
defaultFontActivate  OBJECT-TYPE
SYNTAX      Integer32 (1..255)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the value of defaultFont at activation of the
currently active message for the purpose of determining what the value was at
the time of activation. The value shall be created (overwritten) at the time
when the message was copied into the currentBuffer.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.19"
::= { multiCfg 19 }
```

5.5.9 Default Line Justification Parameter

```
defaultJustificationLine  OBJECT-TYPE
SYNTAX      INTEGER {
                 other(1), -- retired
                 left(2),
                 center(3),
                 right(4),
                 full(5) }
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the default line justification for a message. This
object may be sub-ranged by an implementation; see Section 3.5.2.2.2.5 for
more information.
<Informative> In v02, the enumerated value of 'other' is RETIRED to improve
interoperability.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.6"
DEFVAL  {center}
::= { multiCfg 6 }
```

5.5.10 Default Line Justification Parameter at Activation

```
defaultJustificationLineActivate  OBJECT-TYPE
SYNTAX      INTEGER {
```

```
                 left(2),
                 center(3),
                 right(4),
                 full(5) }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the value of defaultJustificationLine at activation
of the currently active message for the purpose of determining what the value
was at the time of activation. The value shall be created (overwritten) at
the time when the message was copied into the currentBuffer.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.20"
::= { multiCfg 20 }
```

5.5.11 Default Page Justification Parameter

```
defaultJustificationPage  OBJECT-TYPE
SYNTAX      INTEGER {
                 --other(1), -retired
                 top(2),
                 middle(3),
                 bottom(4) }
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the default page justification for a message. This
object may be sub-ranged by an implementation; see Section 3.5.2.2.2.6 for
more information.
<Informative> In v02, the enumerated value of 'other' is RETIRED to improve
interoperability.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.7"
DEFVAL  {middle}
::= { multiCfg 7 }
```

5.5.12 Default Page Justification Parameter at Activation

```
defaultJustificationPageActivate  OBJECT-TYPE
SYNTAX      INTEGER {
                 top(2),
                 middle(3),
                 bottom(4) }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the value of defaultJustificationPage at activation
of the currently active message for the purpose of determining what the value
was at the time of activation. The value shall be created (overwritten) at
the time when the message was copied into the currentBuffer.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.21"
::= { multiCfg 21 }
```

5.5.13 Default Page On Time Parameter

```
defaultPageOnTime  OBJECT-TYPE
SYNTAX      Integer32 (1..255)
UNITS       "tenths of a second"
MAX-ACCESS  read-write
```

```
STATUS      current
DESCRIPTION
"<Definition> Indicates the default page on time, in tenths (1/10) of a
second. If the message is only one page, this value is ignored, and the page
is continuously displayed. This object may be sub-ranged by an
implementation; see Section 3.5.2.2.2.7 for more information.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.8"
DEFVAL  {30}
::= { multiCfg 8 }
```

5.5.14 Default Page On Time Parameter at Activation

```
defaultPageOnTimeActivate  OBJECT-TYPE
SYNTAX      Integer32 (1..255)
UNITS       "tenths of a second"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the value of defaultPageOnTime at activation of the
currently active message for the purpose of determining what the value was at
the time of activation. The value shall be created (overwritten) at the time
when the message was copied into the currentBuffer.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.22"
::= { multiCfg 22 }
```

5.5.15 Default Page Off Time Parameter

```
defaultPageOffTime  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "tenths of a second"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the default page off time, in tenths (1/10) of a
second. If the message is only one page, this value is ignored, and the page
is continuously displayed. This object may be sub-ranged by an
implementation; see Section 3.5.2.2.2.7 for more information.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.9"
DEFVAL  {0}
::= { multiCfg 9 }
```

5.5.16 Default Page Off Time Parameter at Activation

```
defaultPageOffTimeActivate  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "tenths of a second"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the value of defaultPageOffTime at activation of the
currently active message for the purpose of determining what the value was at
the time of activation. The value shall be created (overwritten) at the time
when the message was copied into the currentBuffer.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.23"
::= { multiCfg 23 }
```

5.5.17 Default Background Color RGB Parameter

```
defaultBackgroundRGB  OBJECT-TYPE
SYNTAX      DmsColor
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the color of the background shown on the sign if not
changed by the 'Page Background Color' MULTI tag or the 'Color Rectangle'
MULTI tag.

This object may be sub-ranged by an implementation; see Section 3.5.2.2.2.2
for more information. If a color is requested that is not supported, then a
wrongValue error shall be returned.

<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.12"
::= { multiCfg 12 }
```

5.5.18 Default Background Color RGB Parameter at Activation

```
defaultBackgroundRGBActivate  OBJECT-TYPE
SYNTAX      DmsColor
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the value of defaultBackgroundRGB at activation of
the currently active message for the purpose of determining what the value
was at the time of activation. The value shall be created (overwritten) at
the time when the message was copied into the currentBuffer.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.24"
::= { multiCfg 24 }
```

5.5.19 Default Foreground Color RGB Parameter

```
defaultForegroundRGB  OBJECT-TYPE
SYNTAX      DmsColor
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the color of the foreground shown on the sign unless
changed by the 'Color Foreground' MULTI tag. This is the color used to
illuminate the 'ON' pixels of displayed characters.

This object may be sub-ranged by an implementation; see Section 3.5.2.2.2.2
for more information. If a color is requested that is not supported, then a
wrongValue error shall be returned.

<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.13"
::= { multiCfg 13 }
```

5.5.20 Default Foreground Color RGB Parameter at Activation

```
defaultForegroundRGBActivate  OBJECT-TYPE
SYNTAX      DmsColor
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the value of defaultForegroundRGB at activation of
```

```
the currently active message for the purpose of determining what the value
was at the time of activation. The value shall be created (overwritten) at
the time when the message was copied into the currentBuffer.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.25"
::= { multiCfg 25 }
```

5.5.21 Default Character Set Parameter

```
defaultCharacterSet  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1),
                 eightBit (2),
                 utf8Style (3)}
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the algorithm used to encode each characterNumber
within a MULTI string.
<Format>
  other (1): an encoding scheme other than those listed below, refer to the
device manual.
  eightBit (2): each characterNumber of the current font is encoded as an 8-
bit unsigned integer. This encoding scheme only supports native
representation of character numbers 1 through 255. Higher character numbers
can only be displayed using the hexadecimal character tag ([hcx]).
  utf8Style (4): each characterNumber of the current font is encoded the way
in which UTF-8 encodes code points.
    - Values less than 128 are encoded in the 7 available bits in
      the mask '0xxx xxxx';
    - Values greater than 127 but less than 2048 are encoded in the
      eleven available bits of the mask '110x xxxx 10xx xxxx'; and
    - Remaining values are ecoded in the 16 available bits in the
      mask '1110 xxxx 10xx xxxx 10xx xxxx'
    - This format does not support 4-octet UTF-8 codes because each
      font is limited to 65535 characters.

This object may be sub-ranged by an implementation (e.g., an implementation
might only support eightBit encoding); see Section 3.5.2.2.2.8 for more
information.
<Informative>
Example 1: A MULTI string with an encoded value of 0x41 E2 9C 8B,
would be decoded as follows according to the different encoding
schemes:
  eightBit: four 8-bit characters with codes 65, 226, 156, and 139
  utf8Style: an 8-bit character followed by a 24-bit character. The
    first character is code 65 and the second character is:
      E    2    9    C    8    B     in hex
    1110 0010 1001 1100 1000 1011    in binary
         0010   01 1100   00 1011    removing mask
              0010 0111 0000 1011    compressing
                2    7    0    B     in hex
                             9995    in decimal

Example 2: The MULTI string  '[fo2]', which is a tag to indicate font 2, is
encoded as '0x5b 66 6f 32 5d' in both the eightBit and utf8Style encoding
(i.e., all characters have codes less than 128)
```

```
Example 3: While it is often convenient to align characterNumbers and
characterBitmaps with ASCII or UTF-8 character codes, this is not a
requirement. For example, Character 65 in one font might represent the letter
'A', but in another font it can represent the grinning face emoticon. As
such, any character bitmap (including those represented by Unicode codepoints
greater than 65535) can be defined and supported for display.

Example 4: Per the rules of Multistring syntax, the left and right bracket
characters ('[' and ']') are reserved for tags and to display either, the
character must appear two times in a row within the MULTI string. Thus, to
display '[', the MULTI string would be '0x5b5b' for both 'eightBit' and
'utf8Style' encoding.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.10"
DEFVAL  {eightBit}
::= { multiCfg 10 }
```

5.5.22 Color Scheme Parameter

```
dmsColorScheme  OBJECT-TYPE
SYNTAX      DmsColorScheme
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the color scheme for pixels supported by the DMS.

DMS with dmsColorScheme set to color24bit shall interpret MULTI tags with a
single color parameter (e.g. [cfx]) according to the codes defined in
DmsClassicColor.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.11"
DEFVAL  { monochrome1bit }
::= { multiCfg 11 }
```

5.5.23 Supported MULTI Tags Parameter

```
dmsSupportedMultiTags  OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE (4))MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> An indication of the MULTI Tags supported by the device. This
object is a bitmap representation of tag support.
<Format>
Bit 0 : color background[cbx] / [cbr,g,b] -- deprecated
Bit 1 : color foreground[cfx] / [cfr,g,b]
Bit 2 : flashing[fltxoy] / [floytx]
Bit 3 : font[fox] / [fox,cccc]
Bit 4 : graphic [gn] / [gn,x,y] / [gn,x,y,cccc]
Bit 5 : hexadecimal character[hcx]
Bit 6 : justification line[jlx]
Bit 7 : justification page[jpx]
Bit 8 : manufacturer specific[msx,y]
Bit 9 : moving text[mvtdw,s,r,text]
Bit 10 : new line[nlx]
Bit 11 : new page[np]
Bit 12 : page time[ptxoy]
Bit 13 : spacing character[scx]
Bit 14 : field local time 12 hour[f1]
Bit 15 : field local time 24 hour[f2]
Bit 16 : ambient temperature Celsius[f3]
Bit 17 : ambient temperature Fahrenheit[f4]
Bit 18 : speed km/h[f5]
Bit 19 : speed m/h[f6]
Bit 20 : day of week[f7]
Bit 21 : date of month[f8]
Bit 22 : month of year[f9]
Bit 23 : year 2 digits[f10]
Bit 24 : year 4 digits[f11]
Bit 25 : local time 12 hour AM/PM[f12]
Bit 26 : local time 12 hour am/pm[f13]
Bit 27 : text rectangle [trx,y,w,h]
Bit 28 : color rectangle [crx,y,w,h,z] / [crx,y,w,h,r,g,b]
Bit 29 : Page  background [pbz] / [pbr,g,b]
<Superseded by> dmsSupportedMultiTagsV4
<Informative> This object was deprecated in NTCIP 1203 v05 due to ambiguities
in the bit ordering.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.14"
::= { multiCfg 14 }
```

5.5.24 Maximum Number of Pages Parameter

```
dmsMaxNumberPages  OBJECT-TYPE
SYNTAX      Integer32 (1..255)
UNITS       "pages"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the maximum number of pages allowed in the
dmsMessageMultiString.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.15"
::= { multiCfg 15 }
```

5.5.25 Maximum MULTI String Length Parameter

```
dmsMaxMultiStringLength  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "octets"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the maximum number of octets allowed within the
dmsMessageMultiString.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.16"
::= { multiCfg 16 }
```

5.5.26 Supported MULTI Tags Parameter

```
dmsSupportedMultiTagsV4  OBJECT-TYPE
SYNTAX      BITS { colorBackground (0),
                   colorForeground (1),
                   flashing (2),
                   font (3),
                   graphic (4),
                   hexadecimal (5),
                   justificationLine (6),
                   justificationPage (7),
                   manufacturerSpecific (8),
                   movingText (9),
                   newLine (10),
                   newPage (11),
                   pageTime (12),
                   spacingCharacter (13),
                   localTime12hour (14),
                   localTime24hour (15),
                   temperatureCelsius (16),
                   temperatureFahrenheit (17),
                   speedKPH (18),
                   speedMPH (19),
                   dayOfWeek (20),
                   dateOfMonth (21),
                   monthOfYear (22),
                   year2digit (23),
                   year4digit (24),
                   time12hourAMPM (25),
                   time12hourampm (26),
                   textRectangle (27),
                   colorRectangle (28),
                   pageBackground (29) }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> An indication of the MULTI Tags supported by the device. This
object is a bitmap representation of tag support.
<Valid Value Rule>
colorBackground: [cbx] / [cbr,g,b] -- deprecated
colorForeground: [cfx] / [cfr,g,b]
flashing: [fltxoy] / [floytx]
font: [fox] / [fox,cccc]
graphic: [gn] / [gn,x,y] / [gn,x,y,cccc]
hexadecimal: character[hcx]
justificationLine:[jlx]
justificationPage: [jpx]
manufacturerSpecific: [msx,y]
movingText: [mvtdw,s,r,text]
newLine: [nlx]
newPage: [np]
pageTime: [ptxoy]
spacingCharacter: [scx]
localTime12hour: [f1]
localTime24hour: [f2]
temperatureCelsius: [f3]
temperatureFahrenheit: [f4]
speedKPH: [f5]
speedMPH: [f6]
dayOfWeek: [f7]
dateOfMonth: [f8]
monthOfYear: [f9]
year2digits: [f10]
year4digits: [f11]
time12hourAMPM: [f12]
time12hourampm: [f13]
textRectangle: [trx,y,w,h]
colorRectangle: [crx,y,w,h,z] / [crx,y,w,h,r,g,b]
```

```
pageBackground: [pbz] / [pbr,g,b]
<Informative> [cbx] and [cbr,g,b] were deprecated in v02 but the bit
assignment can still be used to indicate support for these deprecated codes.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.4.17"
::= { multiCfg 17 }
```

5.6 Message Objects

```
dmsMessage  OBJECT-IDENTITY
STATUS  current
DESCRIPTION
"<Definition> This node is an identifier used to group all objects for
support of DMS Message Table functions that are common to DMS devices.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.5"
::= { dms 5 }
```

5.6.1 Number of Permanent Messages Parameter

```
dmsNumPermanentMsg  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "messages"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the current number of Messages stored in non-
volatile, non-changeable memory (e.g., EPROM). For CMS and BOS, this is the
number of different messages that can be assembled.
See the Specifications in association with Requirement 3.6.7.1 to determine
the messages that must be supported.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.5.1"
::= { dmsMessage 1 }
```

5.6.2 Number of Changeable Messages Parameter

```
dmsNumChangeableMsg  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "messages"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the current number of valid Messages stored in non-
volatile, changeable memory. For CMS and BOS, this number shall be zero (0).
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.5.2"
::= { dmsMessage 2 }
```

5.6.3 Maximum Number of Changeable Messages Parameter

```
dmsMaxChangeableMsg  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "messages"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the maximum number of Messages that the sign can
store in non-volatile, changeable memory. For CMS and BOS, this number shall
be zero (0).
See the Specifications in association with Requirement 3.5.6.2 to determine
the messages that must be supported.
```

```
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.5.3"
::= { dmsMessage 3 }
```

5.6.4 Free Bytes within Changeable Memory Parameter

```
dmsFreeChangeableMemory  OBJECT-TYPE
SYNTAX      Integer32 (0..2147483647)
UNITS       "octets"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the number of octets (i.e., bytes) available within
non-volatile, changeable memory. For CMS and BOS, this number shall be zero
(0).
<Informative> See the Specifications in association with Requirement 3.5.6.2
to determine the total memory that must be provided. This value only
indicates the free changeable memory available to this DMS context (i.e., it
may be different than what is available to the entire field device).
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.5.4"
::= { dmsMessage 4 }
```

5.6.5 Number of Volatile Messages Parameter

```
dmsNumVolatileMsg  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "messages"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the current number of valid Messages stored in
volatile, changeable memory. For CMS and BOS, this number shall be zero (0).
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.5.5"
::= { dmsMessage 5 }
```

5.6.6 Maximum Number of Volatile Messages Parameter

```
dmsMaxVolatileMsg  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "messages"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the maximum number of Messages that the sign can
store in volatile, changeable memory. For CMS and BOS, this number shall be
zero (0).
<Informative> See the Specifications in association with Requirement 3.5.6.3
to determine the messages that must be supported.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.5.6"
::= { dmsMessage 6 }
```

5.6.7 Free Bytes within Volatile Memory Parameter

```
dmsFreeVolatileMemory  OBJECT-TYPE
SYNTAX      Integer32 (0..2147483647)
UNITS       "octets"
MAX-ACCESS  read-only
STATUS      current
```

```
DESCRIPTION
"<Definition> Indicates the number of octets (i.e., bytes) available within
volatile, changeable memory. For CMS and BOS, this number shall be zero (0).
<Informative> See the Specifications in association with Requirement 3.5.6.3
to determine the total memory that must be provided. This value only
indicates the free volatile memory available to this DMS context (i.e., it
may be different than what is available to the entire field device).
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.5.7"
::= { dmsMessage 7 }
```

5.6.8 Message Table Parameter

```
dmsMessageTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF DmsMessageEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"<Definition> A table containing the information needed to activate a Message
on a sign. The values of a columnar object (except the dmsMessageStatus)
cannot be changed when the dmsMessageStatus-object of that particular row is
any value other than 'modifying'.
<Table Type> static
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.5.8"
::= {dmsMessage 8}

dmsMessageEntry OBJECT-TYPE
SYNTAX      DmsMessageEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"<Definition> Parameters of the Message Table."
INDEX {dmsMessageMemoryType, dmsMessageNumber}
::= {dmsMessageTable 1}

DmsMessageEntry ::= SEQUENCE {
   dmsMessageMemoryType      INTEGER,
   dmsMessageNumber          Integer32,
   dmsMessageMultiString     MultiString,
   dmsMessageOwner           NtcipOwnerString,
   dmsMessageCRC             Integer32,
   dmsMessageBeacon          Integer32,
   dmsMessagePixelService    Integer32,
   dmsMessageRunTimePriority Integer32,
   dmsMessageStatus          INTEGER
   }
```

5.6.8.1 Message Memory Type Parameter

```
dmsMessageMemoryType  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1), -- deprecated
                 permanent (2),
                 changeable (3),
                 volatile (4),
                 currentBuffer (5),
                 schedule (6),
                 blank (7)}
MAX-ACCESS  read-only
```

```
STATUS      current
DESCRIPTION
"<Definition> Indicates the memory-type used to store a message. Also
provides access to current message (currentBuffer) and currently scheduled
message (schedule). The rows associated with the 'currentBuffer', 'schedule',
and 'blank' message types cannot be written into, because these are either
filled in by the controller (currentBuffer and schedule) or pre-defined and
not modifiable (blank).

The definitions of the enumerated values are:
  other - any other type of memory type that is not listed within one of
     the values below, refer to device manual;
  permanent - non-volatile and non-changeable;
  changeable - non-volatile and changeable;
  volatile - volatile and changeable;
  currentBuffer - contains the information regarding the currently
     displayed message (basically a copy of the message table row
     contents of the message that was successfully activated).
     Only one entry in the table can have the
     value of currentBuffer and the value of the dmsMessageNumber
     object shall be one (1). The content of the
     dmsMessageMultiString object shall be the currently displayed
     message (including a scheduled message), not the content of a
     failed message activation attempt;
  schedule - this entry contains information regarding the currently
     scheduled message as determined by the time-base scheduler (if
     present). Only one entry in the table can have the value of
     'schedule' and the value of dmsMessageNumber for this entry
     shall be 1. Displaying a message through this table row shall set
     the dmsMsgSourceMode object value to 'timebasedScheduler'.
     When no message is currently active based upon the schedule
     or if the schedule currently does not point to any message within
     the message table, the schedule entry shall contain a copy of
     dmsMessageMemoryType 7 (blank) with a dmsMessageNumber value of 1.
  blank - there shall be 255 (message numbers 1 through 255)
     pre-defined, static rows with this message type. These rows are
     defined so that message codes (e.g., objects with SYNTAX of
     either MessageIDCode or MessageActivationCodeV4) can blank the
     sign at a stated run-time priority. The run-time priority of the blank
     message is equal to the message number (e.g., blank message
     number 1 has a run time priority of 1 and so on). The
     dmsMessageCRC for all messages of this type shall be 0x0000 and
     the dmsMessageMultiString shall be an OCTET STRING with a length of
     zero (0). The activation priority shall be determined from the
     activation priority of the MessageActivationCodeV4.

<Informative> In v02, the enumerated value of 'other' was retired to improve
interoperability.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.5.8.1.1"
::= { dmsMessageEntry 1 }
```

5.6.8.2 Message Number Parameter

```
dmsMessageNumber  OBJECT-TYPE
SYNTAX      Integer32 (1..65535)
MAX-ACCESS  read-only
STATUS      current
```

```
DESCRIPTION
"<Definition> Enumerated listing of row entries within the value of the
primary index to this table (dmsMessageMemoryType -object). When the primary
index is 'currentBuffer' or 'schedule', then this value must be one (1). When
the primary index is 'blank', this value shall be from 1 through 255 and all
compliant devices must support all 255 of these 'blank' rows.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.5.8.1.2"
::= { dmsMessageEntry 2 }
```

5.6.8.3 Message MULTI String Parameter

```
dmsMessageMultiString  OBJECT-TYPE
SYNTAX      MultiString
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Contains the message in the format defined by the MultiString
textual convention.

When the primary index is 'schedule', 'blank', 'currentBuffer' or
'permanent', this object shall return a notWritable error to any SET-request.
When the primary index is 'schedule', the object shall return the MULTI
string of the currently scheduled message in response to a GET-request
(regardless whether this message is actually being displayed).
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.5.8.1.3"
::= { dmsMessageEntry 3 }
```

5.6.8.4 Message Owner Parameter

```
dmsMessageOwner  OBJECT-TYPE
SYNTAX      NtcipOwnerString
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the owner or author of this row.
<Informative> Prior to NTCIP 1203 v04, this object was restricted to NVT
ASCII. Implementations can restrict which character values are supported.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.5.8.1.4"
::= { dmsMessageEntry 4 }
```

5.6.8.5 Message CRC Parameter

```
dmsMessageCRC  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the CRC-16 (polynomial defined in ISO/IEC 3309) value
created using the values of the dmsMessageMultiString (MULTI-Message), the
dmsMessageBeacon, and the dmsMessagePixelService objects in the order listed,
not including the OER type or length fields. For messages of the 'blank'
message type, the above algorithm shall be ignored and the dmsMessageCRC
value shall always be zero (0). For messages of the 'schedule' message type,
the CRC value of the currently scheduled message shall always be returned
(regardless whether this message is actually being displayed).
<Informative> Prior to NTCIP 1203 v04, dmsMessageBeacon and
dmsMessagePixelService could be interpretted as being optional to support. In
```

```
this case, the calculated CRC assumed the value zero for the missing data.
Starting with v04, all implementations are required to support these objects,
even if they do not support the related functionality.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.5.8.1.5"
::= { dmsMessageEntry 5 }
```

5.6.8.6 Message Beacon Parameter

```
dmsMessageBeacon  OBJECT-TYPE
SYNTAX      Integer32(0..255)
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the dmsBeaconPlanIndex of the dmsBeaconPlanEntry to
be used to control physical beacons during the display of the message. A
value of zero (0) shall disable the beacons. Implementations that do not
support physical beacons shall support this object but restrict its range to
zero (0).
When the primary index is 'schedule', 'blank', 'currentBuffer', or
'permanent', this object shall return a inconsistentValue error to any SET-
request.
When the primary index is 'schedule', the object shall return the
dmsMessageBeacon setting of the currently scheduled message in response to a
GET-request (regardless whether this message is actually being displayed).
When the dmsMessageMemoryType is 'permanent', the object shall return the
dmsMessageBeacon setting of the factory-preset value in response to a GET-
request.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.5.8.1.6"
DEFVAL  {0}
::= { dmsMessageEntry 6 }
```

5.6.8.7 Message Pixel Service Parameter

```
dmsMessagePixelService  OBJECT-TYPE
SYNTAX      Integer32 (0..1)
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates whether pixel service shall be enabled (1) or
disabled (0) while this message is active. Implementations that do not
support pixel service shall support this object but restrict its range to
zero (0).
When the primary index is 'schedule', 'blank', 'currentBuffer', or
'permanent', this object shall return a inconsistentValue error to any SET-
request.
When the primary index is 'schedule', the object shall return the
dmsMessagePixelService setting of the currently scheduled message in response
to a GET-request (regardless whether this message is actually being
displayed).
When the primary index is 'permanent', the object shall return the
dmsMessagePixelService setting of the factory-preset value in response to a
GET-request.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.5.8.1.7"
DEFVAL  {0}
::= { dmsMessageEntry 7 }
```

5.6.8.8 Message Run Time Priority Parameter

```
dmsMessageRunTimePriority  OBJECT-TYPE
SYNTAX      Integer32 (1..255)
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the run time priority assigned to a particular
message. The value of 1 indicates the lowest level, the value of 255
indicates the highest level. When the dmsMessageMemoryType is 'schedule,' the
value set in this object (e.g. dmsMessageRunTimePriority.6.1) shall override
the run-time priority of the scheduled message. When the dmsMessageMemoryType
is 'blank', the value returned shall be equal to the dmsMessageNumber of that
particular message.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.5.8.1.8"
::= { dmsMessageEntry 8 }
```

5.6.8.9 Message Status Parameter

```
dmsMessageStatus  OBJECT-TYPE
SYNTAX      INTEGER {
                 notUsed (1),
                 modifying (2),
                 validating (3),
                 valid (4),
                 error (5),
                 modifyReq (6),
                 validateReq (7),
                 notUsedReq (8) }
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the current state of the message. This state-machine
allows for defining a message, validating a message, and deleting a message.
See Section 4.3.4 for additional details regarding the state-machine.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.5.8.1.9"
::= { dmsMessageEntry 9 }
```

5.6.9 Validate Message Error Parameter

```
dmsValidateMessageError  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1),
                 none (2),
                 beacons (3),
                 pixelService (4),
                 syntaxMULTI (5) }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> This is an error code used to identify why a message was not
validated. If multiple errors occur, only the first value is indicated. The
syntaxMULTI error is further detailed in the dmsMultiSyntaxError,
dmsMultiSyntaxErrorPosition and dmsMultiOtherErrorDescription objects.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.5.9"
::= { dmsMessage 9 }
```

5.7 Sign Control Objects

```
signControl  OBJECT-IDENTITY
STATUS  current
DESCRIPTION
"<Definition> This node is an identifier used to group all objects for
support of DMS sign control functions that are common to DMS devices.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6"
::= { dms 6 }
```

5.7.1 Control Mode Parameter

```
dmsControlMode  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1),        -- deprecated
                 local (2),
                 external (3),     -- obsolete
                 central (4),
                 centralOverride (5),
                 simulation (6)    -- obsolete
                 }
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> A value indicating the mode that is currently controlling the
sign.
<Format>
The possible modes are:
  other - (deprecated) Other control mode supported by the device (refer to
device manual);
  local - Local control mode (control is at DMS controller);
  external - (obsolete) External control mode;
  central - Central control mode;
  centralOverride - Central station took control over 'local' control, even
     though the control switch at the sign was set to 'local';
  simulation  - (obsolete) controller is in a mode where it accepts every
     command and it pretends that it would execute them but this does not
     happen because the controller only simulates.
<Informative> In v02, the enumerated values of 'other', 'external', and
'simulation' were deprecated to improve interoperability. the values of
external and simulation were never widely used and made obsolete in v04.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.1"
DEFVAL  {central}
::= { signControl 1 }
```

5.7.2 Software Reset Parameter

```
dmsSWReset  OBJECT-TYPE
SYNTAX      Integer32 (0..1)
MAX-ACCESS  read-write
STATUS      deprecated
DESCRIPTION
"<Definition> A software interface to initiate a controller reset. The
execution of the controller reset shall set this object to the value 0.
Setting this object to a value of 1 causes the controller to reset.
<Format> Defined values are
zero (0) = no reset,
one (1) = reset.
```

```
<Superseded by> ISO26048-1-Controller:fdControllerReset
<Informative> This object was deprecated in NTCIP 1203 v04 in favor of the
more generic object defined in ISO 26048-1.
<Parameter Type> Control
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.2"
DEFVAL  {0}
::= { signControl 2 }
```

5.7.3 Activate Message Parameter

```
dmsActivateMessage  OBJECT-TYPE
SYNTAX      MessageActivationCode
MAX-ACCESS  read-write
STATUS      deprecated
DESCRIPTION
"<Definition> A code indicating the active message. The value of this object
may be SET by a management station or modified by logic internal to the DMS
(e.g., activation of the end duration message, etc.).

When modified by internal logic with a reference to a message ID code, the
duration indicates 65535 (infinite), the activate priority indicates 255, and
the source address indicates an address of 127.0.0.1.

If a GET is performed on this object, the DMS shall respond with the value
for the last message that was successfully activated.
The dmsActivateMsgError object shall be updated appropriately upon any
attempt to update the value of this object, whether from an internal or
external source.

If a message activation error occurs (e.g., dmsActivateMsgError is updated to
a value other than 'none'), the new message shall not be activated and, if
the activation request originated from a SET request, a genErr shall be
returned. A management station should then GET the dmsActivateMsgError object
as soon as possible to minimize the chance of additional activation attempts
from overwriting the dmsActivateMsgError.

If a message is attempted to be activated via the scheduler or any internal
message (e.g., end duration message, etc.) and the message to be activated
contains an error, than the following objects shall be set to the appropriate
values (as defined within these objects):
- dmsActivateMsgError,
- dmsActivateErrorMsgCode,
- dmsMultiSyntaxError,
- dmsMultiSyntaxErrorPosition (if supported),
- dmsMultiOtherErrorDescription (if supported),
- dmsDrumStatus (if supported)

A 'criticalTemperature' alarm shall have no effect on the 'activation' of a
message, it will only affect the display of the active message. Thus, a
message activation may occur during a 'criticalTemperature' alarm and the
sign controller will behave as if the message is displayed. However, the
shortErrorStatus will indicate a criticalTemperature alarm and the sign face
illumination will be off. As soon as the DMS determines that the
'criticalTemperature' alarm is no longer present, the DMS shall display the
message stored in the currentBuffer.

<Superseded by> dmsActivateMessageV4
```

```
<Informative> This object was deprecated in NTCIP 1203 v04 in favor of
MessageActivationCodeV4, which is designed to support IPv6 and to more
securely record the source of the request.
<Parameter Type> Control
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.3"
::= { signControl 3 }
```

5.7.4 Message Display Time Remaining Parameter

```
dmsMessageTimeRemaining  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "minutes"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the amount of remaining time in minutes that the
current message shall be active. The time shall be accurate to the nearest
second and rounded up to the next full minute. For example, a value of 2
shall indicate that the time remaining is between 1 minute and 0.1 seconds
and 2 minutes.
When a new message is activated with a minute-based duration, or this object
is directly SET, the minute-based duration value shall be multiplied by 60 to
determine the number of seconds that the message shall be active. Thus, if a
message activation is for 2 minutes, the DMS will be assured to display the
message for 120 seconds.
The value 65535 indicates an infinite duration. A value of zero (0) shall
indicate that the current message display duration has expired.

A SET operation on this object shall allow a Central Computer to extend or
shorten the duration of the message. Setting this object to zero (0) shall
result in the immediate display of the dmsEndDurationMessage.
<Parameter Type> Control
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.4"
DEFVAL  {65535}
::= { signControl 4 }
```

5.7.5 Message Table Source Parameter

```
dmsMsgTableSource  OBJECT-TYPE
SYNTAX      MessageIDCode
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Identifies the message number used to generate the currently
displayed message. This object is written to by the device when the new
message is loaded into the currentBuffer of the dmsMessageTable. The value of
this object contains the message ID code of the message that was copied into
the 'currentBuffer'. This value can only be of message type 'permanent',
'volatile', 'changeable', or 'blank'.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.5"
::= { signControl 5 }
```

5.7.6 Message Requester ID Parameter

```
dmsMsgRequesterID  OBJECT-TYPE
SYNTAX      IpAddress
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
```

```
"<Definition> A copy of the source-address field from the dmsActivateMessage-
object used to activate the current message. If the current message was not
activated by the dmsActivateMessage-object, then the value of this object
shall be zero (0).
<Superceded by> dmsMsgRequesterAddressType, dmsMsgRequesterAddress, and
dmsMsgRequesterSecurityName
<Informative> This object was deprecated in v04 to more accurately and
securely capture the source of the request.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.6"
REFERENCE  "RFC 1155, May 1990"
::= { signControl 6 }
```

5.7.7 Message Source Mode Parameter

```
dmsMsgSourceMode  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1),
                 local (2),
                 external (3),
                 otherCom1( 4), -- obsolete
                 otherCom2 (5), -- obsolete
                 otherCom3 (6), -- obsolete
                 otherCom4 (7), -- obsolete
                 central (8),
                 timebasedScheduler (9),
                 powerRecovery (10),
                 reset (11),
                 commLoss (12),
                 powerLoss (13),
                 endDuration (14)
                 }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the source that initiated the currently displayed
message.
<Format> The enumerations are defined as:
  other (1) - the currently displayed message was activated based on a
     condition other than the ones defined below. This would include any
     auxiliary devices.
  local (2) - the currently displayed message was activated at the sign
     controller using either an onboard terminal or a local interface.
  external (3) - the currently displayed message was activated from a locally
    connected device using serial (or other type of) connection to the sign
    controller such as a laptop or a PDA. This mode shall only be used, if
    the sign controller is capable of distinguishing between a local input
    (see definition of 'local (2)') and a serial connection.
  central (8) - the currently displayed message was activated from the
    central computer.
  timebasedScheduler (9) - the currently displayed message was activated from
    the timebased scheduler as configured within the sign controller.
  powerRecovery (10) - the currently displayed message was activated based
    on the settings within the dmsLongPowerRecoveryMessage,
    dmsShortPowerRecoveryMessage, and the dmsShortPowerLossTime objects.
  reset (11) - the currently displayed message was activated based on the
    settings within the dmsResetMessage object.
  commLoss (12) - the currently displayed message was activated based on
```

```
    the settings within the dmsCommunicationsLossMessage object.
  powerLoss (13) - the currently displayed message was activated based on
    the settings within the dmsPowerLossMessage object. Note: it may not be
    possible to point to this message depending on the technology, e.g. it
    may not be possible to display a message on pure LED or fiber-optic signs
    DURING power loss.
  endDuration (14) - the currently displayed message was activated based on
    the settings within the dmsEndDurationMessage object.

<Informative> In v02, the enumerated values of 'otherComX' were deprecated to
improve interoperability.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.7"
::= { signControl 7 }
```

5.7.8 Short Power Loss Recovery Message Parameter

```
dmsShortPowerRecoveryMessage  OBJECT-TYPE
SYNTAX      MessageIDCode
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the message that shall be activated after a power
recovery following a short power loss affecting the device. Specifying the
dmsMessageMemoryType of 'currentBuffer', dmsMessageNumber of 1 and
dmsMessageCRC of 0 indicates that the message displayed prior to the power
loss will be reactivated after the power recovery. The message shall be
activated with:
- a duration of 65535 (infinite), unless the message type is 'currentBuffer'.
In the latter case, the duration is determined by the value of the
dmsMessageTimeRemaining object (at the time of the power loss) minus the
power outage time;
- an activation priority of 255.
Upon activation of the message, the run-time priority value shall be obtained
from the message table row specified by this object.
The length of time that defines a short power loss is indicated in the
dmsShortPowerLossTime-object.
<Informative> The default value indicates a messageMemoryType of 7 (blank), a
messageNumber of 1, and a messageCRC 0f 0x0000
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.8"
DEFVAL { '0700010000'H }
::= { signControl 8 }
```

5.7.9 Long Power Loss Recovery Message Parameter

```
dmsLongPowerRecoveryMessage  OBJECT-TYPE
SYNTAX      MessageIDCode
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the message that shall be activated after a power
recovery following a long power loss affecting the device. The message shall
be activated with
- a duration of 65535 (infinite), (if this object points to a value of
'currentBuffer', the duration is determined by the value of the
dmsMessageTimeRemaining object minus the power outage time)
- an activation priority of 255.
```

```
Upon activation of the message, the run-time priority value shall be obtained
from the message table row specified by this object.
The length of time that defines a long power loss is indicated in the
dmsShortPowerLossTime-object.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.9"
DEFVAL { '0700010000'H }
::= { signControl 9 }
```

5.7.10 Short Power Loss Time Definition Parameter

```
dmsShortPowerLossTime  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "seconds"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the time, in seconds, from the start of power loss to
the threshold where a short power loss becomes a long power loss. If the
value is set to zero (0), all power failures are defined as long power
losses.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.10"
DEFVAL { 0 }
::= { signControl 10 }
```

5.7.11 Reset Message Parameter

```
dmsResetMessage  OBJECT-TYPE
SYNTAX      MessageIDCode
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the message that shall be activated after a Reset
(either software or hardware) of the device. This assumes that the device can
differentiate between a reset and a power loss. The message shall be
activated with
- a duration of 65535 (infinite) (if this object points to a value of
'currentBuffer', the duration is determined by the value of the
dmsMessageTimeRemaining object minus the power outage time);
- an activation priority of 255.
Upon activation of the message, the run-time priority value shall be obtained
from the message table row specified by this object.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.11"
DEFVAL { '0700010000'H }
::= { signControl 11 }
```

5.7.12 Communications Loss Message Parameter

```
dmsCommunicationsLossMessage  OBJECT-TYPE
SYNTAX      MessageIDCode
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the message that shall be activated when the time
since the last communications from a management station exceeds the
dmsTimeCommLoss time. The message shall be activated with
```

```
- a duration of 65535 (infinite) (if this object points to a value of
'currentBuffer', the duration is determined by the value of the
dmsMessageTimeRemaining object);
- an activation priority of 255.
If the value referenced by this object is invalid, the sign will display a
blank message.
Upon activation of the message, the run-time priority value shall be obtained
from the message table row specified by this object.
The value of this object shall not be implemented when the value of the
dmsControlMode is set to 2 (local). Once the value of the dmsControl Mode
object is set to 4 (central) or 5 (centralOverride) and the value of the
dmsTimeCommLoss parameter has been reached, the value of this object shall be
implemented.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.12"
DEFVAL { '0700010000'H }
::= { signControl 12 }
```

5.7.13 Communication Loss Time Definition Parameter

```
dmsTimeCommLoss  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "minutes"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Defines the maximum time (inclusive), in minutes, between
successive Application Layer messages that can occur before a communication
loss is assumed. If this object is set to zero (0), communications loss shall
be ignored.

The countdown timer associated with this parameter shall be suspended while
the sign control parameter has a value of 'local (2)', e.g., the sign is in
local control. The countdown timer shall be restarted (reset and started
again) once the sign control parameter value is switched to 'central (4)' or
'centralOverride (5)'.
<Informative> This timer differs from the Data Link Layer timers (T1 to T4).
A dial-up circuit may have short time-outs at the DL Layer, but central might
only dial up once a month to confirm operation, in which case this object
would be set to ~ 35 days.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.13"::= { signControl 13 }
```

5.7.14 Power Loss Message Parameter

```
dmsPowerLossMessage  OBJECT-TYPE
SYNTAX      MessageIDCode
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the message that shall be activated DURING the loss
of power of the device. The message shall be activated with:
a duration of 65535 (infinite) (if this object points to a value of
'currentBuffer', the duration is determined by the value of the
dmsMessageTimeRemaining object);
an activation priority of 255.
Upon activation of the message, the run-time priority value shall be obtained
from the message table row specified by this object.
```

```
<Informative> Not all technologies support the means to display a message
while the power is off.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.14"
DEFVAL { '0700010000'H }
::= { signControl 14 }
```

5.7.15 End Duration Message Parameter

```
dmsEndDurationMessage  OBJECT-TYPE
SYNTAX      MessageIDCode
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the message that shall be activated after the
indicated duration for a message has expired and no other Message had been
activated (see dmsActivateMessage). The message shall be activated with
- a duration of 65535 (infinite) (if this object points to a value of
'currentBuffer', the duration is determined by the value of the
dmsMessageTimeRemaining object);
- an activation priority of 255.
Upon activation of the message, the run-time priority value shall be obtained
from the message table row specified by this object.

If the end duration message does not activate because this object is an
invalid value, the sign shall blank with the default value of this object.

<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.15"
DEFVAL { '0700010000'H }
::= { signControl 15 }
```

5.7.16 Memory Management Parameter

```
dmsMemoryMgmt  OBJECT-TYPE
SYNTAX      INTEGER  {
                 other (1), -- deprecated
                 normal (2),
                 clearChangeableMessages (3),
                 clearVolatileMessages (4) }
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Allows the system to manage the device's memory.

<Format>
SNMP Get operations on this object should always return normal (2).

   clearChangeableMessages (3): the controller shall set dmsMessageStatus for
all changeable messages to notUsed (1), and release all memory associated
with changeable messages.  This action does not affect any changeable
graphics or fonts.

   clearVolatileMessages (4): the controller shall set dmsMessageStatus for
all volatile messages to notUsed (1), and release all memory associated with
volatile messages.  This action does not affect any changeable graphics or
fonts.
```

```
<Informative> In v02, the enumerated value of 'other' was deprecated to
improve interoperability.

<Parameter Type> Control
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.16"
DEFVAL  {normal}
::= { signControl 16 }
```

5.7.17 Activate Message Error Parameter

```
dmsActivateMsgError  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1),
                 none (2),
                 priority (3),
                 messageStatus (4),
                 messageMemoryType (5),
                 messageNumber (6),
                 messageCRC (7),
                 syntaxMULTI (8),
                 localMode (9),
                 centralMode (10),
                 centralOverrideMode (11) }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> This is an error code used to identify why a message was not
displayed. Even if multiple errors occur, only one error is indicated.
<Format>
  other (1):  any error not defined below.
  none (2): no error.
  priority(3):  the activation priority in the MessageActivationCodeV4 is
     less than the run time priority of the  currently displayed message.
     If this error occurs, the corresponding bit (message error) within
     the 'shortErrorStatus' object shall be set.
  messageStatus(4):  the 'dmsMessageStatus' of the message to be
     activated is not 'valid'. If this error  occurs, the corresponding bit
     (message error) within the 'shortErrorStatus' object shall be set.
     Note: In the 1997 version of NTCIP 1203, this bit was assigned
     the name of 'underValidation'. It has been renamed to better
     reflect the fact that this bit can be set due to the message being
     in a number of different states, not just the 'validating' state.
  messageMemoryType(5):  the message memory type in the
     MessageActivationCodeV4 is not supported by the  device. If this
     error occurs, the corresponding bit (message error) within the
     'shortErrorStatus' object shall be set.
  messageNumber(6):  the message number in the
     MessageActivationCodeV4 is not supported or is not defined
     (populated) by the device. If this error occurs, the corresponding
     bit (message error) within the 'shortErrorStatus' object shall be set.
  messageCRC(7):  the checksum in the MessageActivationCodeV4 is
     different than the CRC value  contained in the 'dmsMessageCRC'.
     If this error occurs, the corresponding bit (message error) within
     the 'shortErrorStatus' object shall be set.
  syntaxMULTI(8):  a MULTI syntax error was detected during
     message activation. The error is further detailed in the
     'dmsMultiSyntaxError', 'dmsMultiSyntaxErrorPosition', and
     'dmsMultiOtherErrorDescription' objects. If this error occurs, the
     corresponding bit (message error) within the 'shortErrorStatus' object
     shall be set.
  localMode(9):  the central system attempted to activate a message
     while the 'dmsControlMode' object is  'local'. This error shall NOT
     be set if the value of the 'dmsControlMode' is set to
     'central',  or 'centralOverride'. If this error occurs, the
     corresponding bit (message error) within the 'shortErrorStatus'
     object shall be set.
  centralMode (10):  a locally connected system attempted to activate
     a message while the 'dmsControlMode' object is 'central'.
     This error shall NOT be set if the value of the 'dmsControlMode'
     is set to 'local'. If this error occurs, the corresponding
     bit (message error) within the 'shortErrorStatus'
     object shall be set.
  centralOverrideMode (11):  a locally connected system attempted to activate
     a message while the 'dmsControlMode' object is 'centralOverride', even
     though the local switch is set to local control.
     If this error occurs, the corresponding bit (message error)
      within the 'shortErrorStatus' object shall be set.

A 'criticalTemperature' alarm shall have no effect on the 'activation' of a
message, it only effects the display of the active message. Thus, a message
activation may occur during a 'criticalTemperature' alarm and the sign
controller behaves as if the message is displayed. However, the
shortErrorStatus indicates a criticalTemperature alarm and the sign face
illumination is off. As soon as the DMS determines that the
'criticalTemperature' alarm is no longer present, the DMS shall display the
message stored in the currentBuffer.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.17"
::= { signControl 17 }
```

5.7.18 MULTI Syntax Error Parameter

```
dmsMultiSyntaxError  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1),
                 none (2),
                 unsupportedTag (3),
                 unsupportedTagValue (4),
                 textTooBig (5),
                 fontNotDefined (6),
                 characterNotDefined (7),
                 fieldDeviceNotExist (8),
                 fieldDeviceError (9),
                 flashRegionError (10),
                 tagConflict (11),
                 tooManyPages (12),
                 fontVersionID (13),
                 graphicID (14),
                 graphicNotDefined (15),
                 invalidCharacterEncoding (16) }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> This is an error code used to identify the first detected
syntax error within the MULTI message.
<Format>
```

```
  other (1):  An error other than one of those listed.
  none (2):  No error detected.
  unsupportedTag (3):  The tag is not supported by this device.
  unsupportedTagValue (4):  The tag value is not supported by this
     device.
  textTooBig (5):  Too many characters on a line, too many lines for a
     page, or font is too large for the display.
  fontNotDefined (6):  The font is not defined in this device.
  characterNotDefined (7):  The character is not defined in the
     selected font.
  fieldDeviceNotExist (8):  The field device does not exist / is not
     connected to this device.
  fieldDeviceError (9):  This device is not receiving input from the
     referenced field device and/or the field device has a  fault.
  flashRegionError (10):  The flashing region cannot be flashed by this
     device.
  tagConflict (11):  The message cannot be displayed with the
     combination of tags and/or tag implementation cannot be resolved.
  tooManyPages (12):  Too many pages of text exists in the message.
  fontVersionID (13):  The fontVersionID contained in the MULTI tag
     [fox,cccc] does not match the fontVersionID for the fontNumber
     indicated.
  graphicID (14):  The dmsGraphicID contained in the
     MULTI tag [gx,cccc] does not match the dmsGraphicID for the
     dmsGraphicIndex indicated.
  graphicNotDefined (15):  The graphic is not defined in this device.
  invalidCharacterEncoding (16): The encoding of a character cannot be
     decoded according to the current defaultCharacterSet.
<Informative> Within eightBit encoding, all encodings are valid except for
0x00; implementations may report 'characterNotDefined' for this situation to
support backwards compatibility but invalidCharacterEncoding is preferred.
Within utf8Style encoding, there are many encodings that are invalid (e.g.,
any octet with all four high-order bits set is invalid).
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.18"
::= { signControl 18 }
```

5.7.19 Position of MULTI Syntax Error Parameter

```
dmsMultiSyntaxErrorPosition  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "characters"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> This is the offset from the first character (e.g. first
character has offset 0, second is 1, etc.) of the MULTI string where the
SYNTAX error occurred.
<Informative>The offset is measured in characters, which can be different
than the number of octets when defaultCharacterSet has a value of utf8Style.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.19"
::= { signControl 19 }
```

5.7.20 Other MULTI Error Parameter

```
dmsMultiOtherErrorDescription  OBJECT-TYPE
SYNTAX      SnmpAdminString (SIZE (0..50))
MAX-ACCESS  read-only
```

```
STATUS      current
DESCRIPTION
"<Definition> Indicates vendor-specified error message descriptions.
Associated errors occurred due to vendor-specific MULTI-tag responses. The
value of this object is valid only if dmsValidateMessageError has a value of
'syntaxMULTI(5)' or dmsActivateMsgError has a value of 'syntaxMULTI(8)' and
dmsMultiSyntaxError is 'other(1)'. The language used for this text-based
message shall be as defined by ISO26048-1-Controller:fdDefaultLanguage.
<Informative> Prior to NTCIP 1203 v04, this object type only supported NVT-
ASCII characters.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.20"
::= { signControl 20 }
```

5.7.21 Pixel Service Duration Parameter

```
vmsPixelServiceDuration  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "seconds"
MAX-ACCESS  read-write
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the number of seconds to perform pixel service on an
entire sign. If the vmsPixelServiceDuration expires during a pixel service
routine, that routine shall be completed before stopping or restarting a new
pixel service routine due to vmsPixelServiceFrequency. A value of zero
disables pixel service.
<Superseded by> vmsPixelServiceDurationV4
<Informative> This object was deprecated in NTCIP 1203 v04 to provide better
and more appropriate resolution for the duration (i.e., ms rather than s)
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.21"
::= { signControl 21 }
```

5.7.22 Pixel Service Frequency Parameter

```
vmsPixelServiceFrequency  OBJECT-TYPE
SYNTAX      Integer32 (0..1440)
UNITS       "minutes"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the pixel service cycle time (period) in minutes. A
value of zero indicates continuous pixel service from vmsPixelServiceTime to
the epoch of midnight. A value of 1440 indicates one pixel service in a 24-
hour period.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.22"
DEFVAL  {1440}
::= { signControl 22 }
```

5.7.23 Pixel Service Time Parameter

```
vmsPixelServiceTime  OBJECT-TYPE
SYNTAX      INTEGER (0..1440)
UNITS       "minutes"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the base time at which the first pixel service shall
```

```
occur. Time is expressed in minutes from the epoch of midnight of each day.
Midnight shall be determined by local time, if the local clock feature (i.e.,
ISO 26048-1, 8.3.2) is supported; otherwise, it shall be based on UTC time.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.23"
DEFVAL  {1}
::= { signControl 23 }
```

5.7.24 Message Code of the Activation Error Parameter

```
dmsActivateErrorMsgCode  OBJECT-TYPE
SYNTAX      MessageActivationCode
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the MessageActivationCode that resulted in the
current value of the dmsActivateMsgError object.
<Superseded by> dmsActivateErrorMsgCodeV4
<Informative> This object was deprecated in NTCIP 1203 v04 to use the updated
MessageActivationCodeV4
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.24"
::= { signControl 24 }
```

5.7.25 Activate Message State Parameter

```
dmsActivateMessageState  OBJECT-TYPE
SYNTAX      INTEGER {
              fastActivationSign(1),
              slowActivatedOK(2),
              slowActivatedError(3),
              slowActivating(4) }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Signs that are able to change their message with fast
activation always return 'fastActivationSign(1)'. This allows a central to
use this object to determine whether or not the sign does fast activation
(that is, whether the sign can immediately change the display). Signs that do
slow activation (such as a rotary drum sign) shall set this object to
'slowActivating(4)' during the changing of the display and when the message
change has completed shall change it to 'slowActivatedOK(2)' if successful or
'slowActivatedError(3)' if an error occurred during the display change.

A sign with fast activation uses this object only to indicate that it is a
fast activation sign. Such a sign provides an immediate response to a SET of
dmsActivateMessageV4 (or dmsActivateMessage).

With a slow activation sign there are two opportunities to detect an error.
The first comes when the SET of dmsActivateMessageV4 is performed, just as in
the fast activation sign. It could be a bad message number or other error. If
such an error is received, the message change does not occur and therefore
this object can be ignored. If the SET of dmsActivateMessageV4 succeeds, then
the central must wait for either slowActivatedOK or slowActivatedError in
this object. If the sign detects an error, it shall set this object to
slowActivatedError and set the 'message error' bit in the shortErrorStatus
object. When a central receives slowActivatedError, it shall examine other
status objects specific to the sign, such as the rotary drum status objects,
to determine the precise error.
```

```
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.25"
::= { signControl 25 }
```

5.7.26 Activate Message V4 Parameter

```
dmsActivateMessageV4  OBJECT-TYPE
SYNTAX      MessageActivationCodeV4
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> A code indicating the active message or message to be
activated. The value of this object may be SET by a management station or
modified by logic internal to the DMS (e.g., activation of the end duration
message, etc.).

When modified by internal logic with a reference to a message ID code, the
duration indicates 65535 (infinite) and the activate priority indicates 255.

If a GET is performed on this object, the DMS shall respond with the value
for the last message that was successfully activated.

Upon any attempt to update the value of this object, whether from an internal
or external source, the following objects shall be updated appropriately:
- dmsActivateMsgError,
- dmsActivateErrorMsgCodeV4,
- dmsMultiSyntaxError,
- dmsMultiSyntaxErrorPosition (if supported),
- dmsMultiOtherErrorDescription (if supported),
- dmsDrumStatus (if supported)

If a message activation error occurs (e.g., dmsActivateMsgError is updated to
a value other than 'none'), the new message shall not be activated and, if
the activation request originated from a SET request, the appropriate error
code (NTCIP 1203 v04, Clause 4.3.5) shall be returned. A management station
should then GET the dmsActivateMsgError object as soon as possible to
minimize the chance of additional activation attempts from overwriting the
dmsActivateMsgError.

A 'criticalTemperature' alarm shall have no effect on the 'activation' of a
message, it will only affect the display of the active message. Thus, a
message activation may occur during a 'criticalTemperature' alarm and the
sign controller will behave as if the message is displayed. However, the
shortErrorStatus will indicate a criticalTemperature alarm and the sign face
illumination will be off. As soon as the DMS determines that the
'criticalTemperature' alarm is no longer present, the DMS shall display the
message stored in the currentBuffer.
<Parameter Type> Control
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.26"
::= { signControl 26 }
```

5.7.27 Message Code of the Activation Error V4 Parameter

```
dmsActivateErrorMsgCodeV4  OBJECT-TYPE
SYNTAX      MessageActivationCodeV4
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
```

```
"<Definition> Indicates the MessageActivationCodeV4 that resulted in the
current value of the dmsActivateMsgError object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.27"
::= { signControl 27 }
```

5.7.28 Message Requester Address Type Parameter

```
dmsMsgRequesterAddressType  OBJECT-TYPE
SYNTAX      TransportAddressType
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> An indication of the type of address used by the data packet
that contained the request to activate the current message. If the current
message was activated by internal logic (e.g., scheduling logic), the value
of this object shall be local (13).
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.28"
REFERENCE  "RFC 3419, December 2002"
::= { signControl 28 }
```

5.7.29 Message Requester Address Parameter

```
dmsMsgRequesterAddress  OBJECT-TYPE
SYNTAX      TransportAddress
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> The source address of the data packet that contained the
request used to activate the current message. If the current message was
activated by internal logic (e.g., scheduling logic), the value of this
object is implementation specific.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.29"
REFERENCE  "RFC 3419, December 2002"
::= { signControl 29 }
```

5.7.30 Message Requester Security Name Parameter

```
dmsMsgRequesterSecurityName  OBJECT-TYPE
SYNTAX      SnmpAdminString
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> The security name associated with the data packet that
contained the request used to activate the current message. If the current
message was activated by internal logic (e.g., scheduling logic), the value
of this object shall be the security name associated with the internal
process that called the message (e.g., the ISO26048-1-
Action:fdActionSecurityName if called by the action table).
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.30"
::= { signControl 30 }
```

5.7.31 Pixel Service Duration V4 Parameter

```
vmsPixelServiceDurationV4  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "milliseconds"
MAX-ACCESS  read-write
```

```
STATUS      current
DESCRIPTION
"<Definition> Indicates the number of milliseconds to perform pixel service
on an entire sign (i.e., 0 - 65.535 seconds). If the
vmsPixelServiceDurationV4 expires during a pixel service routine, that
routine shall be completed before stopping or restarting a new pixel service
routine due to vmsPixelServiceFrequency. A value of zero disables pixel
service.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.6.31"
::= { signControl 31 }
```

5.8 Illumination/Brightness Objects

```
illum  OBJECT-IDENTITY
STATUS  current
DESCRIPTION
"<Definition> This node is an identifier used to group all objects supporting
DMS sign illumination functions that are common to DMS devices.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.7"
::= { dms 7 }
```

5.8.1 Illumination Control Parameter

```
dmsIllumControl  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1),         -- deprecated
                 photocell (2),
                 timer (3),
                 manual (4),        -- obsolete
                 manualDirect (5),
                 manualIndexed (6) }
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the method used to select the Brightness Level.
A DMS may subrange the values supported, as indicated.
<Format>
  other (1) - indicates that the Brightness Level is based on a
     mechanism not defined by this standard; see manufacturer
     documentation. This mode is DEPRECATED.
  photocell (2) - indicates that the Brightness Level is based on
     photocell status. Support for this mode shall be supported if
     user need 2.5.1.5 is selected.
  timer (3) - indicates that the Brightness Level is set by an internal
     timer. The details of this timer are not defined by this standard.
  manual (4) - indicates that the Brightness Level must be changed via
     the dmsIllumManLevel object. This mode is OBSOLETE.
  manualDirect (5) - indicates that a user can change the brightness output
     to any of the brightness levels (dmsIllumNumBrightLevels) supported by
     the sign. This is not the same as the number of brightness levels
     defined within the table of the dmsIllumBrightnessValues object.
  manualIndexed (6) - indicates that a user can change the brightness output
     to any of the rows defined within the table of the
     dmsIllumBrightnessValues object.
```

```
The DMS shall support manualDirect, manualIndex, or both. Support for all
other values is optional.

When switching to any of the manual modes (manual, manualDirect,
manualIndexed) from any other mode, the current brightness level shall
automatically be loaded into the dmsIllumManLevel object.

<Informative> In v02, the enumerated values of 'other' and 'manual' were
deprecated to improve interoperability. In v04, the value of 'manual' was
made obsolete.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.7.1"
DEFVAL  {photocell}
::= { illum 1 }
```

5.8.2 Maximum Illumination Photocell Level Parameter

```
dmsIllumMaxPhotocellLevel  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the maximum value given by the
dmsIllumPhotocellLevelStatus-object.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortMaxValue.FAL
<Informative> This object was deprecated in v04 and replaced with the more
universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.7.2"
::= { illum 2 }
```

5.8.3 Status of Illumination Photocell Level Parameter

```
dmsIllumPhotocellLevelStatus  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the level of Ambient Light as a value ranging from 0
(darkest) to the value of dmsIllumMaxPhotocellLevel object (brightest), based
on the photocell detection. The dmsIllumPhotocellLevelStatus object is
considered a virtual photocell level in that it may be algorithmically
determined from one or more photocells and is the value used for calculations
dealing with the brightness table. The algorithm used to determine the
virtual level from the actual photocell readings is manufacturer specific to
accommodate various hardware needs.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeValueStatsMfrValue.FAL
<Informative> This object was deprecated in v04 and replaced with the more
universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.7.3"
::= { illum 3 }
```

5.8.4 Number of Illumination Brightness Levels Parameter

```
dmsIllumNumBrightLevels  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
```

```
"<Definition> Indicates the number of individually selectable Brightness
Levels supported by the device, excluding the OFF level (=value of zero [0]).
This value indicates the total levels of brightness that this device
supports, not the number of rows defined in the table of the
dmsIllumBrightnessValues object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.7.4"
::= { illum 4 }
```

5.8.5 Status of Illumination Brightness Level Parameter

```
dmsIllumBrightLevelStatus  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the current Brightness Level of the device, ranging
from 0 (OFF) to the maximum value given by the dmsIllumNumBrightLevels-
object (Brightest).
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.7.5"
::= { illum 5 }
```

5.8.6 Illumination Manual Level Parameter

```
dmsIllumManLevel  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the desired value of the Brightness Level as a value
ranging from 0 to the value of the dmsIllumNumBrightLevels-object when under
manual control.
When the dmsIllumControl object is set to a value of 'manualDirect (5)' then
the maximum value that this object can have is the total levels of brightness
that this device supports. A user can calculate the direct manual light
output as (65535 * (dmsIllumManLevel object value / dmsIllumNumBrightLevels
object value)).
When the dmsIllumControl object is set to a value of 'manualIndexed (6)' then
the maximum value that this object can be set to is the number of rows
defined in the table of the dmsIllumBrightnessValues object.
If the device supports version 1 and the dmsIllumControl object is set to a
value of 'manual (4)', then the deployment could be either (contact your
vendor to determine which way is implemented)
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.7.6"
::= { illum 6 }
```

5.8.7 Illumination Brightness Values Parameter

```
dmsIllumBrightnessValues  OBJECT-TYPE
SYNTAX      OCTET STRING
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> . An OCTET STRING describing the sign's light output in
relationship to the Photocell(s) detection of ambient light.
<Format> For each light output level, there is a corresponding range of
photocell levels. The number of light output levels transmitted is defined by
the first byte of the data packet, but cannot exceed the value of the
```

```
dmsIllumNumBrightLevels object. Setting the value of this object to a non-
supported or erroneous value shall lead to a wrongValue or inconsistentValue
error, as appropriate. Cause of this error shall be denoted by the
dmsIllumBrightnessValuesError object.
After a SET, an implementation may interpolate these entries to create a
table with as many entries as needed, but the value of the object shall not
be affected by such interpolations.
For each light output level, there are three 16-bit values that occur in the
following order: Light output level, Photocell level down, Photocell level
up.
The light output level is a value between 0 (no light output) and 65535
(maximum light output). Each step is 1/65535 of the maximum light output
(linear scale).
The Photocell-level-down is the lowest photocell level allowed to maintain
the light output level. If the photocell level goes below this point, the
light output level goes down one light output level.
The Photocell-level-up is the highest photocell level for this light output
level. If the photocell level goes above this point, the light output level
goes up one light output level.
The photocell level (Up and Down) values may not exceed the value of the
dmsIllumMaxPhotocellLevel object.
The points transmitted should be selected so that there is no photocell level
which does not have a light output level. Hysteresis is possible by defining
the photocell-level-up at a level higher than the upper level's photocell-
level-down.
The encoding of the structure shall consist of a one byte integer value
indicating the number of rows in the table. This is followed by a series of
OER encoded Strings of the following structure:
  SEQUENCE {
      lightOutput            INTEGER (0..65535),
      photocellLevelDown     INTEGER (0..65535),
      photocellLevelUp       INTEGER (0..65535) }

If the sign does not support photocell and the dmsIllumControl object value
is set to 'manualIndexed', then the values for the 'photocellLevelDown' and
'photocellLevelUp' still need to be entered that the table does not cause any
errors as defined in the dmsIllumBrightnessValuesError object. However, since
no photocell is supported, the entered values for 'photocellLevelDown' and
'photocellLevelUp' for the various 'lightOutputs' are meaningless.

<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.7.7"
::= { illum 7 }
```

5.8.8 Brightness Values Error Parameter

```
dmsIllumBrightnessValuesError  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1),
                 none (2),
                 photocellGap (3),
                 negativeSlope (4),
                 tooManyLevels (5),
                 invalidData (6) }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
```

```
"<Definition> Indicates the error encountered when the brightness table was
SET.
<Format>
other(1) - is for a manufacturer specific indication when none of the
     other possible values can be used.
none(2)  - indicates that no error was encountered.
photocellGap(3) - indicates that certain photocell levels do not have
     an associated brightness level.
negativeSlope(4) - indicates that the photocell range used to select a
     brighter brightness level is lower or overlaps the photocell range
     used to select a dimmer brightness level. Note that some signs
     may allow a negative slope for special conditions without
     generating an error; e.g., external illumination for a reflective sign
     may be allowed to turn off during daylight conditions rather than
     getting brighter.
tooManyLevels(5) - indicates that more brightness levels are defined
     than are reported by dmsIllumNumBrightLevels.
invalidData(6) - indicates a manufacturer defined condition of invalid
     data not described by the other options.

<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.7.8"
::= { illum 8 }
```

5.8.9 Status of Illumination Light Output Parameter

```
dmsIllumLightOutputStatus  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the current physical light output value ranging from
0 (darkest) to 65535 (maximum output).
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.7.9"
::= { illum 9 }
```

5.9 Scheduling Action Objects

```
dmsSchedule  OBJECT-IDENTITY
STATUS  current
DESCRIPTION
"<Definition> This node is an identifier used to group all DMS device-
specific objects supporting DMS sign timebased scheduling.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.8"
::= { dms 8 }
```

5.9.1 Action Table Entries Parameter

```
numActionTableEntries  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "entries"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the number of rows that are stored in the
dmsActionTable. See the Specification in association with Requirement
3.5.10.4 to determine the number of actions required.
<Parameter Type> Status
```

```
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.8.1"
::= { dmsSchedule 1 }
```

5.9.2 Action Table Parameter

```
dmsActionTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF DmsActionEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "<Definition> A table containing a list of message codes. The
scheduler (as defined in the dayPlanTable within NTCIP 1201) determines when
a message shall be displayed. This table determines which message shall be
activated.
Implementations that support the dmsActionTable (this table) and the
fdActionTable (as definedin ISO 26048-1) shall allow any instance of
fdActionPointer to reference any row of the dmsActionTable.
<Informative> This action table identifies the message to be placed into the
schedule row of the dmsMessageTable (e.g., the MultiString is stored in
dmsMessageMultistring.6.1)
<Table Type> static
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.8.2"
::= {dmsSchedule 2}

dmsActionEntry OBJECT-TYPE
SYNTAX      DmsActionEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "<Definition> Parameters of the DMS Action Table."
INDEX {dmsActionIndex}
::= {dmsActionTable 1}

DmsActionEntry ::= SEQUENCE {
   dmsActionIndex        Integer32,
   dmsActionMsgCode      MessageIDCode }
```

5.9.2.1 Action Index Parameter

```
dmsActionIndex  OBJECT-TYPE
SYNTAX      Integer32 (1..255)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Enumerated listing of row entries. The value of this object
cannot exceed the value of the numActionTableEntries - object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.8.2.1.1"
::= { dmsActionEntry 1 }
```

5.9.2.2 Action Message Code Parameter

```
dmsActionMsgCode  OBJECT-TYPE
SYNTAX      MessageIDCode
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> A code indicating the message memory type, the message number
and the associated message-specific CRC as indicated within the message
table.
Setting the CRC portion of this object to all zeros allows a message to
become activated without the CRC validation process.
```

```
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.8.2.1.2"
DEFVAL  {'0000000000'h}
::= { dmsActionEntry 2 }
```

5.10 Auxiliary I/O Objects

```
-- The objects originally defined under this node were moved to the
-- 'global' node and NTCIP 1201 with the publication of NTCIP 1201 v02
-- Amendment 2.
```

5.11 Sign Status

```
dmsStatus  OBJECT-IDENTITY
STATUS  current
DESCRIPTION
"<Definition> This node is an identifier used to group all objects supporting
  DMS sign status monitoring functions that are common to DMS devices.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9"
::= { dms 9 }
```

5.11.1 Core Status

5.11.1.1 Number of Rows in MULTI Field Table Parameter

```
statMultiFieldRows  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "rows"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the number of rows in the statMultiFieldTable that
are currently being used.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.1"
::= { dmsStatus 1 }
```

5.11.1.2 MULTI Field Table Parameter

```
statMultiFieldTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF StatMultiFieldEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "<Definition> A table containing the currently displayed value of
a specified Field. The number of rows  is given by the value of
statMultiFieldRows-object.
<Table Type> static
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.2"
::= {dmsStatus 2}

statMultiFieldEntry OBJECT-TYPE
SYNTAX      StatMultiFieldEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "<Definition> Parameters of the Status Multi Field Table."
INDEX {statMultiFieldIndex}
::= {statMultiFieldTable 1}

StatMultiFieldEntry ::= SEQUENCE {
   statMultiFieldIndex        Integer32,
   statMultiFieldCode         Integer32,
   statMultiCurrentFieldValue MultiString}
```

5.11.1.2.1 MULTI Field Index Parameter

```
statMultiFieldIndex  OBJECT-TYPE
SYNTAX      Integer32 (1..255)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> The index into this table indicating the sequential order of
the field within the MULTI-string.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.2.1.1"
::= { statMultiFieldEntry 1 }
```

5.11.1.2.2 Code of MULTI Field Parameter

```
statMultiFieldCode  OBJECT-TYPE
SYNTAX      Integer32 (1..255)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the ID of the statMultiCurrentFieldValue-object. The
field codes are indicated under the 'Field'-tag in MULTI.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.2.1.2"
::= { statMultiFieldEntry 2 }
```

5.11.1.2.3 Current Value of the MULTI Field Parameter

```
statMultiCurrentFieldValue  OBJECT-TYPE
SYNTAX      MultiString (SIZE (0..50))
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the value of the field in the currently displayed
MULTI-message.
<Informative> The value provides the value of the field, which can contain
embedded tags (e.g., to change font) but not embedded field tags.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.2.1.3"
::= { statMultiFieldEntry 3 }
```

5.11.1.3 Current Speed Parameter

```
dmsCurrentSpeed  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "kilometers per hour"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> The current speed value detected by the attached device. The
speed is in kilometers per hour (km/h). This value may vary from the
displayed speed value due to application specific implementation.
<Superseded by> fdSrsaPortValue.VSS.1
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced by
the more generic SRSA table that offers greater capabilities and can support
multiple sensors.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.3"
::= { dmsStatus 3 }
```

5.11.1.4 Current Speed Limit Parameter

```
dmsCurrentSpeedLimit  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "kilometers per hour"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the current speed limit in kilometers per hour
(km/h).
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.4"
::= { dmsStatus 4 }
```

5.11.1.5 Watchdog Failure Count Parameter

```
watchdogFailureCount  OBJECT-TYPE
SYNTAX      Counter32
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> A counter indicating the number of watchdog failures that have
been detected.
<Superseded by> ISO26048-1-Controller:fdWatchdogFailures
<Informative> This object was deprecated in v04 and replaced with the more
universal fdWatchdogFailures object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.5"
::= { dmsStatus 5 }
```

5.11.1.6 Open Door Status Parameter

```
dmsStatDoorOpen  OBJECT-TYPE
SYNTAX      INTEGER (0..255)
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates whether any of the doors to the controller cabinet or
the sign housing are open. This is a bitmap; if a bit is set (= 1) then the
door is open; if a bit not is not set, then the associated door is closed.
Each door is associated with a bit (bit-door correlation order specified by
manufacturer) allowing for up to 8 doors.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeWarning.FDO
<Informative> This object was deprecated in v04 and replaced with the more
universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.6"
::= { dmsStatus  6 }
```

5.11.2 Status Error Objects

```
statError  OBJECT-IDENTITY
STATUS  current
DESCRIPTION
"<Definition> This node is an identifier used to group all objects supporting
DMS sign message error status functions that are common to DMS devices.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7"
::= { dmsStatus 7 }
```

5.11.2.1 Controller Errors

5.11.2.1.1 Short Error Status Parameter

```
shortErrorStatus  OBJECT-TYPE
SYNTAX      NtcipIntegerBitmap16
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> A bitmap of summary errors. When a bit is set, the error is
presently active. When a bit is clear the error is not currently active. If
no sensor is present or supported (for a corresponding bit), the bit
shall not be set.
<Format>  The bits are defined as follows:
Bit 0- other - not defined by this standard, see the value of
      dmsOtherShortErrorDescr.
      The value of other was deprecated in v02 and reinstated in v04.
Bit 1- communications error - this bit shall be set if any error
      associated with the communications between the central
      computer and the device occurs.
Bit 2- power error - this bit shall be set if any error is detected for any
      power supply associated with any component (i.e., any bit set in
      fdSrsaPortTypeWarning.FPA, FPS, or FPV) or if the generator fuel
      level is below its minimum threshold or has an error
      (i.e., any bit set in fdSrsaPortTypeWarning.FGF).
Bit 3- attached device error - this bit shall be set if any error
      associated with attached (supported, and enabled) external
      devices occurs (i.e., any bit set in any fdSrsaPortTypeWarning object
      that is not otherwise specified by other bots of this object).
Bit 4- lamp error - this bit shall be set if any errors associated with any
      lamp occurs (i.e., any bit set in fdSrsaPortTypeWarning.MLS) This bit
      is only applicable to devices that support lamps such as fiber optic
      signs or front-illuminated reflective signs. This bit is not applicable
      to lamps or fluorescent lights illuminating the housing or cabinet.
Bit 5- pixel error - this bit shall be set if any errors associated with any
      pixel occurs (see the objects pixelFailureTableNumRows for NTCIP 1203
      v01 deployments, and pixelFailureTableNumRows, and/or
      dmsPixelFailureTestRows and dmsPixelFailureMessageRows for NTCIP 1203
      v02 deployments.).
      This bit is only applicable to devices that support illumination of
      individual pixels, but not to drum signs.
      Note that certain sign technologies such as flip disk only sign may
      not be able to determine pixel errors.
Bit 6- photocell error - this bit shall be set if any errors associated with
      the supported light sensors occurs (i.e., any bit set in
      fdSrsaPortTypeWarning.FAL).
Bit 7- message error - this bit shall be set if any errors associated with
      activating and/or displaying a message occurs (see the object
      dmsActivateMsgError).
Bit 8- controller error - this bit shall be set if any errors associated
      with the controller occurs (i.e., any bit set in fdControllerStatus).
Bit 9- temperature warning - this bit shall be set if any of the
      temperature values detected by the device exceed
      non-standardized temperature values (i.e., any bit set in
      fdSrsaPortTypeWarning.FAT FET, or FPT). This bit is included to allow
      vendors or agencies to define vendor- or agency-specific
      threshold objects that indicate temperature changes that are of
      interest but not dangerous to the life-expectancy of the device
      (see also the 'critical temperature' bit)
Bit 10- climate-control system error - this bit shall be set if any errors
      associated with the climate control systems such as fans and/or
      heaters occurs (i.e., any bit set in fdSrsaPortTypeWarning.FAC, FDC,
      FFO, FHO, or FTT).
Bit 11- critical temperature error - this bit shall be set if the critical
      temperature as defined by the value of the critical temperature
      objects have been exceeded (i.e., any bit set in
      fdSrsaPortTypeStatus.FAT, FET, or FPT).
Bit 12- Drum-sign Rotor error - This bit shall be set if any errors
      associated with the rotor of a drum sign occurs (i.e., any bit set in
      fdSrsaPortTypeWarning.MDS).
Bit 13- This bit shall be set if any door to any DMS field component
      (cabinet or housing) is open (i.e., any bit set in
      fdSrsaPortTypeWarning.FDO).
Bit 14- Humidity warning - This bit shall be set if any humidity sensor
      sensor is reporting a humidity warning (i.e., any bit set in
      fdSrsaPortTypeWarning.FAH or FEH) .

<Informative> To track a history of transient error conditions utilize the
event logging table located in the Global Objects Definitions (NTCIP 1201).
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.1"
::= { statError 1 }
```

5.11.2.1.2 Controller Error Status Parameter

```
controllerErrorStatus  OBJECT-TYPE
SYNTAX      NtcipIntegerBitmap8
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> A bitmap of controller related errors where the bits are
defined as follows:
<Format>
  Bit 0- other controller error
  Bit 1- PROM error
  Bit 2- program/processor error
  Bit 3- RAM error
  Bit 4- Controller to display interface error
If a bit is set to one (1), then the associated error is existing; if the bit
is set to zero (0), then the associated error is not existing.
<Superseded by> ISO26048-1-Controller:fdControllerStatus
<Informative> This object was deprecated in NTCIP 1203 v05 for the new global
object solution.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.10"
::= { statError 10 }
```

5.11.2.1.3 Other Short Error Description Parameter

```
dmsOtherShortErrorDescr  OBJECT-TYPE
SYNTAX      SnmpAdminString
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> A textual description of the 'other' error causing Bit 0 of
shortErrorStatus to be set. This shall be a zero-length string when Bit 0 of
shortErrorStatus is not set. The language used for this text-based message
shall be as defined by ISO26048-1-Controller:fdDefaultLanguage.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.1"
::= { statError 39 }
```

5.11.2.2 Power Status Data

5.11.2.2.1 Power Failure Status Map Parameter

```
dmsPowerFailureStatusMap  OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE (0..64))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates whether each power supply within the sign has failed.
If a power supply has failed, its associated bit is set to one (1). The size
of this object shall always present one bit for each power supply supported
by the system, but shall not contain more than seven bits that are not
associated with any power supply.
A power supply is a local supply of subsystem power, such as a voltage
regulator. Further information about each failed subsystem may be found in
the dmsPowerStatusTable. If any bit within this object is set, then the
'power error' bit within the shortErrorStatus object shall also be set.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeStatus.FPS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.11"
::= { statError 11 }
```

5.11.2.2.2 Number of Rows in Power Table Parameter

```
dmsPowerNumRows  OBJECT-TYPE
SYNTAX      Integer32 (0..512)
UNITS       "rows"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the number of rows in the dmsPowerStatusTable.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeCount.FPS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.12"
::= { statError 12 }
```

5.11.2.2.3 Power Status Table Parameter

```
dmsPowerStatusTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF DmsPowerStatusEntry
MAX-ACCESS  not-accessible
STATUS      deprecated
DESCRIPTION "<Definition> A table containing status information for each
power supply within a DMS.
<Table Type> static
<Superseded by> ISO26048-1-Srsa:fdSrsaPortTable
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.13"
```

```
::= {statError 13}

dmsPowerStatusEntry OBJECT-TYPE
SYNTAX      DmsPowerStatusEntry
MAX-ACCESS  not-accessible
STATUS      deprecated
DESCRIPTION "<Definition> An entry in the power status table.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortEntry
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object."
INDEX { dmsPowerIndex }
::= {dmsPowerStatusTable 1}

DmsPowerStatusEntry ::= SEQUENCE {
   dmsPowerIndex        Integer32,
   dmsPowerDescription  DisplayString,
   dmsPowerMfrStatus    DisplayString,
   dmsPowerStatus       INTEGER,
   dmsPowerVoltage      Integer32,
   dmsPowerType         INTEGER}
```

5.11.2.2.3.1 Power Index Parameter

```
dmsPowerIndex  OBJECT-TYPE
SYNTAX      Integer32 (1..512)
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Index of the power supply status table. This index corresponds
to the bit position within the dmsPowerFailureStatusMap bitmap: the row with
index 1 corresponds to the low-order bit of the dmsPowerFailureStatusMap,
etc.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortIndex.FPS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.13.1.1"
::= { dmsPowerStatusEntry 1 }
```

5.11.2.2.3.2 Power Description Parameter

```
dmsPowerDescription  OBJECT-TYPE
SYNTAX      DisplayString (SIZE (0..64))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Human-readable description of the power supply. This value
should provide enough information for maintenance personnel to identify the
physical location of the power supply within the DMS. The description shall
include a meaningful definition of the location where the power supply
defined in this row is located within the DMS.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortDescription.FPS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.13.1.2"
::= { dmsPowerStatusEntry 2 }
```

5.11.2.2.3.3 Power Manufacturer-Defined Status Parameter

```
dmsPowerMfrStatus  OBJECT-TYPE
SYNTAX      DisplayString (SIZE (0..64))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the current manufacturer-defined status of the power
supply. This object allows a vendor to provide the operator with additional
information.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortTextStatus.FPS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.13.1.3"
::= { dmsPowerStatusEntry 3 }
```

5.11.2.2.3.4 Power Status Parameter

```
dmsPowerStatus  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1), -- deprecated
                 noError (2),
                 powerFail (3),
                 voltageOutOfSpec (4),
                 currentOutOfSpec (5) }
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the current status of the indicated power supply.
<Format>
  other (1): Current status not defined by this standard. This value was
    deprecated in NTCIP 1203 v02.
  noError (2): No error detected.
  powerFail (3): The power supply is producing no output.
  voltageOutOfSpec (4): The power supply is producing voltage outside
    the vendor recommended specification
  currentOutOfSpec (5): The power supply is producing current outside of
    the vendor recommended specification.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortValue.FPS
<Informative> This object was deprecated in NTCIP 1203 v04. The replacement
object is limited to errors within the power supply; input and output
voltages and currents are defined in separate entries within the SRSA table.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.13.1.4"
::= { dmsPowerStatusEntry 4 }
```

5.11.2.2.3.5 Power Voltage Status Parameter

```
dmsPowerVoltage  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "hundredths of a volt"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> A voltage measurement in units of hundredths (1/100) of a volt.
The maximum value (0xFFFF) corresponds to a voltage of 655.35 volts. AC
voltages are given in RMS (Root Mean Squared) value.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortValue.FxV, where x indicates the
voltage being measured
```

```
<Informative> This object was deprecated in NTCIP 1203 v04. The SRSA table
disambiguates between the input and output voltage.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.13.1.5"
::= { dmsPowerStatusEntry 5 }
```

5.11.2.2.3.6 Power Status Type Parameter

```
dmsPowerType  OBJECT-TYPE
SYNTAX      INTEGER {
                  other (1),
                  acLine (2),
                  generator (3),
                  solar (4),
                  batteryUPS (5),
                  ledSupply (6),
                  lampSupply (7) }
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the type of power source or power supply represented
by the table row.
<Format>
   other: indicates that the power source or supply is not one of the
      types listed  below (see device manual), in which case the
      corresponding dmsPowerDescription field provides a
      description of the entity represented by the row.
   acLine: indicates that the row represents a source of AC  power  This
      is also reflected in the lineVolts object.;
   generator: indicates that the row represents a generator;
   solar: indicates that the row represents solar equipment;
   batteryUPS: indicates that the row represents a
      battery or UPS with no significant charging occurring. This
      is also reflected in the signVolts object.
   ledSupply: indicates that the row represents the power supply to
      one or more LED pixels.
   lampSupply: indicates that the row represents the power supply to
      one or more display lamps.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeCode
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.13.1.6"
::= { dmsPowerStatusEntry 6 }
```

5.11.2.3 Climate Control Status Data

5.11.2.3.1 Fan Failure Parameter

```
fanFailures  OBJECT-TYPE
SYNTAX  OCTET STRING (SIZE (0..4))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates whether each fan (system) within a DMS is capable of
operating, expressed as a bitmap. If a fan (system) failed, its associated
bit is set to one (1). Each fan system is associated with a bit (bit-fan
correlation order specified by manufacturer) allowing for up to 32 fan
systems to report failure status. A fan system is defined as a single fan,
group of fans, sensors, or filter systems. Whether each bit specifies a fan
or fan system is dependent on the manufacturer.
<Superseded by> dmsClimateCtrlStatusMap
<Informative> This object was deprecated in NTCIP 1203 v02 and replaced by a
more generic object
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.8"
::= { statError 8 }
```

5.11.2.3.2 Fan Test Activation Parameter

```
fanTestActivation  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1),
                 noTest (2),
                 test (3) }
MAX-ACCESS  read-write
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the state of the fan testing. The actual test routine
can vary among different manufacturers. The results of the fan test shall be
stored in either the fanFailures-objects. Setting the value to test starts
the test, meaning this test is executed once. The sign controller
automatically sets the value of this object back to noTest after completion.
<Superseded by> dmsClimateCtrlTestActivation
<Informative> This object was deprecated in NTCIP 1203 v02, which also
specifically discouraged the use of the value 'other'.
<Parameter Type> Control
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.9"
::= { statError 9 }
```

5.11.2.3.3 Climate-control System Failure Status Map Parameter

```
dmsClimateCtrlStatusMap  OBJECT-TYPE
SYNTAX  OCTET STRING (SIZE (0..64))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates whether each climate-control subsystem within the
sign has failed. If a subsystem has failed, its associated bit is set to one
(1).  The size of this object shall always present one bit for each climate
control-subsystem supported by the system, but shall not contain more than
seven bits that are not associated with any climate-control subsystem.
Further information about each failed subsystem may be found in the
dmsClimateCtrlStatusTable. If any bit within this object is set, then the
'climate-control system error' bit within the shortErrorStatus object shall
also be set.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeStatus for FAC, FDC, FFO, FHO, and
FTT
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.14"
::= { statError 14 }
```

5.11.2.3.4 Number of Rows in Climate-control Status Table Parameter

```
dmsClimateCtrlNumRows  OBJECT-TYPE
SYNTAX      Integer32
```

```
UNITS       "rows"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the number of rows in the dmsClimateCtrlStatusTable.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeCount for FAC, FDC, FFO, FHO, and
FTT
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.16"
::= { statError 16 }
```

5.11.2.3.5 Climate-control System Failure Status Table Parameter

```
dmsClimateCtrlStatusTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF DmsClimateCtrlStatusEntry
MAX-ACCESS  not-accessible
STATUS      deprecated
DESCRIPTION "<Definition> A table containing status information for each
climate-control subsystem within a DMS.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortTable for FAC, FDC, FFO, FHO, and
FTT
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.<Table Type> static
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.17"
::= {statError 17}

dmsClimateCtrlStatusEntry OBJECT-TYPE
SYNTAX      DmsClimateCtrlStatusEntry
MAX-ACCESS  not-accessible
STATUS      deprecated
DESCRIPTION "<Definition> An entry in the climate-control status
table.<Superseded by> ISO26048-1-Srsa:fdSrsaPortEntry for FAC, FDC, FFO, FHO,
and FTT
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object."
INDEX { dmsClimateCtrlIndex }
::= {dmsClimateCtrlStatusTable 1}

DmsClimateCtrlStatusEntry ::= SEQUENCE {
  dmsClimateCtrlIndex          Integer32,
  dmsClimateCtrlDescription    DisplayString,
  dmsClimateCtrlMfrStatus      DisplayString,
  dmsClimateCtrlErrorStatus    INTEGER,
  dmsClimateCtrlOnStatus       Integer32,
  dmsClimateCtrlTestActivation INTEGER,
  dmsClimateCtrlAbortReason    DisplayString,
  dmsClimateCtrlType           INTEGER}
```

5.11.2.3.5.1 Climate-control Index Parameter

```
dmsClimateCtrlIndex  OBJECT-TYPE
SYNTAX      Integer32 (1..512)
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Index of the climate control table. This index corresponds to
```

```
the bit position within the dmsClimateCtrlStatusMap bitmap: the row with
index 1 corresponds to the low-order bit of the dmsClimateCtrlStatusMap, etc.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortIndex for FAC, FDC, FFO, FHO, and
FTT
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.17.1.1"
::= { dmsClimateCtrlStatusEntry 1 }
```

5.11.2.3.5.2 Climate-control Description Parameter

```
dmsClimateCtrlDescription  OBJECT-TYPE
SYNTAX  DisplayString (SIZE (0..64))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Human-readable description of the subsystem. This value should
provide enough information for maintenance personnel to identify the type
(AC, dehumidifier, heater, fan, etc) and physical location of the subsystem
within the DMS. The description shall include a meaningful definition of the
location where the sensor defined in this row is located within the DMS.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortDescription for FAC, FDC, FFO, FHO,
and FTT
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.17.1.2"
::= { dmsClimateCtrlStatusEntry 2 }
```

5.11.2.3.5.3 Climate-Control Manufacturer-Defined Status Parameter

```
dmsClimateCtrlMfrStatus  OBJECT-TYPE
SYNTAX  DisplayString (SIZE (0..64))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the current manufacturer-defined status of the
climate-control equipment. This object allows a vendor to provide the
operator with additional information.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortTextStatus for FAC, FDC, FFO, FHO,
and FTT
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.17.1.3"
::= { dmsClimateCtrlStatusEntry 3 }
```

5.11.2.3.5.4 Climate-control System Error Status Parameter

```
dmsClimateCtrlErrorStatus  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1), -- deprecated
                 noError (2),
                 fail (3),
                 notMonitored (4)}
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the current status of the indicated subsystem.
```

```
<Superseded by> ISO26048-1-Srsa:fdSrsaPortStatus for FAC, FDC, FFO, FHO, and
FTT
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.17.1.4"
::= { dmsClimateCtrlStatusEntry 4 }
```

5.11.2.3.5.5 Climate-control On Status Parameter

```
dmsClimateCtrlOnStatus  OBJECT-TYPE
SYNTAX      Integer32 (0..1)
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates whether the indicated climate-control subsystem is
currently active. The bit orientation of 1 (set) indicates the system is on
and a value of 0 (cleared) indicates the system is off.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortValue for FAC, FDC, FFO, FHO, and
FTT
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.17.1.5"
::= { dmsClimateCtrlStatusEntry 5 }
```

5.11.2.3.5.6 Climate-control Test Activation Parameter

```
dmsClimateCtrlTestActivation  OBJECT-TYPE
SYNTAX      INTEGER {
                        noTest(2),
                        test(3),
                        testAborted(4) }
MAX-ACCESS  read-write
STATUS      deprecated
DESCRIPTION
"<Definition> Set to test(3) to activate the test for the climate-control
device indicated by this row of the table. If the test completes normally,
upon completion the sign shall set this object to noTest(2), with the results
of the test appearing in the dmsClimateCtrlStatusMap and the
dmsClimateCtrolErrorStatus objects and optionally in the
dmsClimateCtrlMfrStatus object. If the test does not complete normally
(either the sign declined to run the test at all or the test was started but
aborted), the sign shall set this object to testAborted(4) and shall specify
the reason for the abort in the dmsClimateCtrlAbortReason object. In the case
of an abort, the dmsClimateCtrlStatusMap, dmsClimateCtrolErrorStatus,  and
dmsClimateCtrlMfrStatus objects arenot be changed due to the test. At any
time, this object can be set to noTest(2) to end any test in progress (in
this case a subsequent read sees noTest(2) and not testAborted(4)). The value
testAborted(4) is a read-only status - this object cannot be set to
testAborted(4).
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeTestActivation for FAC, FDC, FFO,
FHO, and FTT
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Control
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.17.1.6"
::= { dmsClimateCtrlStatusEntry 6 }
```

5.11.2.3.5.7 Climate-control Test Activation Abortion Parameter

```
dmsClimateCtrlAbortReason  OBJECT-TYPE
SYNTAX  DisplayString (SIZE (0..64))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> If the dmsClimateCtrlTestActivation object has a value of
testAborted(4), this object indicates the manufacturer-defined reason as to
why the climate-control test was aborted. This object is meaningless if
dmsClimateCtrlTestActivation has any value other than testAborted(4).
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeTestAbortReason for FAC, FDC, FFO,
FHO, and FTT
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.17.1.7"
::= { dmsClimateCtrlStatusEntry 7 }
```

5.11.2.3.5.8 Climate-control Device Type Parameter

```
dmsClimateCtrlType  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1),
                 fansVentilation (2),
                 fansSignFace (3),
                 dehumidifier (4),
                 heatCabinet (5),
                 heatHousing (6),
                 heatSignFace (7),
                 airConditioningCabinet (8),
                 airConditioningHousing (9)}
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the type of the climate control device described in
this row of the table.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeCode for FAC, FDC, FFO, FHO, and
FTT
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.17.1.8"
::= { dmsClimateCtrlStatusEntry 8 }
```

5.11.2.4 Pixel Failure Data

5.11.2.4.1 Number of Rows in Pixel Failure Table Parameter

```
pixelFailureTableNumRows  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "rows"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> The total number of rows contained in the pixelFailureTable
each indicating failed pixels.
The value is the sum of the dmsPixelFailureTestRows and the
dmsPixelFailureMessageRows objects.
<Parameter Type> Status
```

```
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.2"
::= { statError 2 }
```

5.11.2.4.2 Pixel Failure Table Parameter

```
pixelFailureTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF PixelFailureEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "<Definition> A table containing the X and Y location of a failed
pixel. The number of rows is given by the value of pixelFailureTableNumRows -
object.
<Table Type> static
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.3"
::= {statError 3}

pixelFailureEntry OBJECT-TYPE
SYNTAX      PixelFailureEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "<Definition> Parameters of the Pixel Failure Table. The
detection of pixel failures during message displays shall be appended to the
end of the table."
INDEX { pixelFailureDetectionType, pixelFailureIndex}
::= {pixelFailureTable 1}

PixelFailureEntry ::= SEQUENCE {
   pixelFailureDetectionType  INTEGER,
   pixelFailureIndex          Integer32,
   pixelFailureXLocation      Integer32,
   pixelFailureYLocation      Integer32,
   pixelFailureStatus         NtcipIntegerBitmap8}
```

5.11.2.4.2.1 Pixel Failure Detection Type Parameter

```
pixelFailureDetectionType  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1), -- deprecated
                 pixelTest (2),
                 messageDisplay(3) }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the type of test/display that leads to the pixel
failure entry.
Once a pixel is detected as failed, it is entered in the table with a type of
either pixelTest or messageDisplay. In either case the failed pixel stays in
the table until pixelTestActivation is set to either test or clearTable.
<Format>
Detection type pixelTest and messageDisplay are considered different methods
of testing for failed pixels.
other: Not defined by this document; this value was deprecated in NTCIP 1203
    v02.
pixelTest: This is considered a foreground processing method of failed pixel
    detection. Failed pixels detected during a foreground pixel test are
    entered in the pixelTest pixel failure type. During a foreground pixel
    test, the message on the display may or may not stay present on the
    display.
messageDisplay: This is considered a background processing method of failed
```

```
    pixel detection. During a background test, the readability/legibility of
    the message shall not be affected by the test. If the manufacturer
    supports background pixel test, failed pixels detected during a
    background pixel test are entered in the messageDisplay pixel failure
    type.
<Informative> In v02, the enumerated value of 'other' was deprecated to
improve interoperability.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.3.1.1"
::= { pixelFailureEntry 1 }
```

5.11.2.4.2.2 Pixel Failure Index Parameter

```
pixelFailureIndex  OBJECT-TYPE
SYNTAX      Integer32 (1..65535)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Enumerated listing of row entries. Within each
pixelFailureDetectionType, entries shall start with one (1) and be
sequential.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.3.1.2"
::= { pixelFailureEntry 2 }
```

5.11.2.4.2.3 Pixel Failure X Location Parameter

```
pixelFailureXLocation  OBJECT-TYPE
SYNTAX      Integer32 (1..65535)
UNITS       "pixels"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the X location of the failed pixel. The X direction
is the horizontal direction. The X location is counted from the left-most
pixel in number of pixels.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.3.1.3"
::= { pixelFailureEntry 3 }
```

5.11.2.4.2.4 Pixel Failure Y Location Parameter

```
pixelFailureYLocation  OBJECT-TYPE
SYNTAX      Integer32 (1..65535)
UNITS       "pixels"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the Y location of the failed pixel. The Y direction
is the vertical direction. The Y location is counted from the top-most pixel
in number of pixels.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.3.1.4"
::= { pixelFailureEntry 4 }
```

5.11.2.4.2.5 Pixel Failure Status Parameter

```
pixelFailureStatus  OBJECT-TYPE
SYNTAX      NtcipIntegerBitmap8
MAX-ACCESS  read-only
STATUS      current
```

```
DESCRIPTION
"<Definition> Indicates the current status of the specified pixel and the
operation which made this determination. This is a bit field with the
following format:
<Format>
Bit 0:   0: Not stuck on / 1: Stuck On
Bit 1:   0: No Color Error / 1: Color Error
Bit 2:   0: no electrical error / 1: electrical error
Bit 3:   0: no mechanical error / 1: mechanical error
Bit 4:   0: Not stuck off / 1: Stuck off
Bit 5:   0: No partial failure / 1: Partial failure - a partial failure
indicates a loss of pixel functionality that does not affect the full
luminance or visible area of a pixel. For example, if an LED DMS uses
multiple redundant LEDs at each pixel, the failure of a single LED at a given
pixel would be flagged as a partial failure. A partial failure indicates that
the pixel is still functioning, but with reduced visibility.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.3.1.5"
::= { pixelFailureEntry 5 }
```

5.11.2.4.3 Pixel Test Activation Parameter

```
pixelTestActivation  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1), -- deprecated
                 noTest (2),
                 test (3),
                 clearTable (4) }
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the state of the pixel testing. The actual test
routine can vary among different manufacturers. The results of the pixel
failure test shall be stored in the pixel failure table.
<Format>
  other: the meaning of this value is not defined in this document; this
    value was deprecated in NTCIP 1203 v02.
  noTest: Receiving this value in response to a GetRequest indicates that
    the sign is not currently undergoing a pixel test. The value 'noTest' is
    not valid in a SetRequest.
  test: Setting the value to 'test' clears both the pixel failure table and
    the pixelFailureTableNumRows object (for both 'messageDisplay' and
    'pixelTest' types) and starts the pixel test, meaning this test is
    executed once. Pixel failures identified by this test are entered into
    the pixelFailureTable with pixelFailureDetectionType equal to
    'pixelTest'. The sign controller automatically sets the value of this
    object back to 'noTest' after completion. Receiving this value in
    response to a GetRequest indicates that the sign is still undergoing a
    previous test.
  clearTable: Setting the value to 'clearTable' clears both the pixel failure
    table and the pixelFailureTableNumRows object for both 'messageDisplay'
    and 'pixelTest' types. The sign controller automatically sets the value
    of this object back to 'noTest' after completion. Receiving this value in
    response to a GetRequest indicates that the sign is still clearing the
    table from a previous command.

<Informative> Versions of this object prior to NTCIP 1203 v04 were ambiguous
regarding (1) the device transitioning the value 'clearTable' to 'noTest'
```

```
once the table was cleared, (2) the possibility of using the 'noTest' value
in a SetRequest, and (3) the meaning of a 'clearTable' value in a response to
a GetRequest.
In v02, the enumerated value of 'other' is RETIRED to improve
interoperability.
<Parameter Type> Control
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.4"
DEFVAL  {noTest}
::= { statError 4 }
```

5.11.2.4.4 Pixel Status Table Parameter

```
pixelStatusTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF PixelStatusEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "<Definition> A table containing a bitmap of all pixels. Because
the bitmap may be too large for a single data packet, the bitmap is broken
into blocks (represented by a dmsPixelStatus object). The number of rows is
determined by the number of pixels in the sign and the maximum size of the
dmsPixelStatus object.
<Table Type> static
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.18"
::= {statError 18}

pixelStatusEntry  OBJECT-TYPE
SYNTAX      PixelStatusEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "<Definition> Parameters of the Pixel Status Table."
INDEX { dmsPixelStatusIndex}
::= {pixelStatusTable 1}

PixelStatusEntry ::= SEQUENCE {
   dmsPixelStatusIndex    Integer32,
   dmsPixelStatus         OCTET STRING,
   dmsPixelStatusV4       NtcipOctetBitmap3200 }
```

5.11.2.4.4.1 Pixel Status Index Parameter

```
dmsPixelStatusIndex  OBJECT-TYPE
SYNTAX      Integer32 (1..255)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Index of the pixel status table. This index corresponds to one
entry of maximum size of 400 octets containing the status of each pixel
within the sign.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.18.1.1"
::= { pixelStatusEntry 1 }
```

5.11.2.4.4.2 Pixel Status Parameter

```
dmsPixelStatus  OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE(1..400))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates whether a pixel within the sign has failed.
```

```
Indicates the status of each pixel within the sign. Each bit within this
object is associated with an individual pixel. If a pixel has an error the
associated bit shall be one (1). If a pixel has no error, the associated bit
shall be zero (0).

The lowest-order bit corresponds to the top-left pixel of the sign face; the
next bit corresponds to the next pixel to the right, etc. At the end of a
pixel row, the next bit corresponds to the leftmost bit of the row below. If
any bit within this object is set, then the 'pixel error' bit within the
shortErrorStatus object shall also be set. This object value is changed when
any type of pixel test within pixelTestActivation has completed.

Each row entry of this table contains a maximum of 400 octets which is
equivalent to 3200 pixels per row entry. The last entry of this table does
not need to be 400 octets but the preceding entries do. Any remaining bits
within the final byte of the last entry of this table shall be zero.

<Superseded by> dmsPixelStatusV4
<Informative> This object was deprecated in NTCIP 1203 v04 because the
definition of this object was determined to be ambiguous since an OCTET
STRING does not have a clearly defined 'lowest order bit'.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.18.1.2"
::= { pixelStatusEntry 2 }
```

5.11.2.4.4.3 Pixel Status Parameter

```
dmsPixelStatusV4  OBJECT-TYPE
SYNTAX      ITSBitmap (SIZE (0..400))
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates whether a pixel within the sign has failed.

Indicates the status of each pixel within the sign. Each bit within this
object is associated with an individual pixel. If a pixel has an error the
associated bit shall be one (1). If a pixel has no error, the associated bit
shall be zero (0).

The highest-order bit of the first octet of the first entry table corresponds
to the top-left pixel of the sign face; the next bit (i.e., the second-to-
highest-order bit of the first octet of the first entry) corresponds to the
next pixel to the right, etc. At the end of an octet, the next bit is the
highest-order bit of the following octet. At the end of a pixel row, the next
pixel is the leftmost pixel of the row below. At the end of 3200 pixels (400
octets) the next pixel is encoded in the highest-order bit of the first octet
of the second entry in the table, etc.

If any bit within any instance of this object type has a value of one (1),
the 'pixel error' bit within the shortErrorStatus object shall also be set.
This object value is changed when any type of pixel test within
pixelTestActivation has completed.

Each row entry of this table, except for the final row, shall contain 400
octets, which is equivalent to 3200 pixels. The last entry of this table
shall use the minimum number of octets possible to encode the remaining
```

```
pixels supported by the sign. Any remaining bits within the final octet of
the last entry of this table shall be zero.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.18.1.3"
::= { pixelStatusEntry 3 }
```

5.11.2.4.5 Number of Pixel Failures from Pixel Test Parameter

```
dmsPixelFailureTestRows  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "rows"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the number of rows in the pixelFailureTable with a
pixelFailureDetectionType of 'pixelTest'.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.19"
::= { statError 19 }
```

5.11.2.4.6 Number of Pixel Failures from Message Display Parameter

```
dmsPixelFailureMessageRows  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "rows"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the number of rows in the pixelFailureTable with a
pixelFailureDetectionType of 'messageDisplay'.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.20"
::= { statError 20 }
```

5.11.2.5 Lamp Status Data

5.11.2.5.1 Stuck On Lamp Failure Parameter

```
lampFailureStuckOn  OBJECT-TYPE
SYNTAX  OCTET STRING (SIZE (0..255))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates whether each lamp within the sign is stuck on as a
bitmap. If a lamp is stuck on, its associated bit is set to one (1). The size
of this object shall always present one bit for each lamp supported by the
DMS, but shall not contain more than seven bits that are not associated with
any lamp. The lamp error bit in shortErrorStatus shall be set if any bit in
lampFailureStuckOff is set.
The size of this object shall always present one bit for each lamp supported
by the DMS, regardless of the failure status of the individual lamps.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeWarning.MLS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.5"
::= { statError 5 }
```

5.11.2.5.2 Stuck Off Lamp Failure Parameter

```
lampFailureStuckOff  OBJECT-TYPE
SYNTAX  OCTET STRING (SIZE (0..255))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates whether each lamp within the sign is stuck off as a
bitmap. If a lamp is stuck off, its associated bit is set to one (1). The
size of this object shall always present one bit for each lamp supported by
the DMS, but shall not contain more than seven bits that are not associated
with any lamp. The lamp error bit in shortErrorStatus shall be set if any bit
in lampFailuresStuckOn is set.
The size of this object shall always present one bit for each lamp supported
by the DMS, regardless of the failure status of the individual lamps.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeWarning.MLS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.6"
::= { statError 6 }
```

5.11.2.5.3 Lamp Test Activation Parameter

```
lampTestActivation  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1), -- deprecated
                 noTest (2),
                 test (3) }
MAX-ACCESS  read-write
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the state of the lamp testing. The actual test
routine can vary among different manufacturers. The results of the lamp
failure test shall be stored appropriately, in the lampFailureStuckOn- and/or
in the lampFailureStuckOff-objects. In v02, the enumerated value of 'other'
was deprecated to improve interoperability.
Setting the value to test starts the test, meaning this test is executed
once. The sign controller shall automatically set the value of this object
back to noTest after completion. Activation of lamp test shall clear the
object lampFailureStuckOn and lampFailureStuckOff, the lamp status table, and
the 'lamp fail' error bit in shortErrorStatus. Results of the lamp test shall
update these two objects, the table and the error bit.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeTestActivation.MLS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Control
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.7"
DEFVAL  {noTest}
::= { statError 7 }
```

5.11.2.5.4 Number of Rows in Lamp Status Table Parameter

```
dmsLampNumRows  OBJECT-TYPE
SYNTAX      Integer32 (0..2040)
UNITS       "rows"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the number of rows in the dmsLampStatusTable. The
```

```
number of rows is equal to the total number of lamps contained in the DMS,
regardless of the failure status of the individual lamps.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeCount.MLS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.23"
::= { statError 23 }
```

5.11.2.5.5 Lamp Status Table Parameter

```
dmsLampStatusTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF DmsLampStatusEntry
MAX-ACCESS  not-accessible
STATUS      deprecated
DESCRIPTION "<Definition> A table containing status information for each lamp
within a DMS.
<Table Type> static
<Superseded by> ISO26048-1-Srsa:fdSrsaPortTable.MLS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.24"
::= {statError 24}

dmsLampStatusEntry OBJECT-TYPE
SYNTAX      DmsLampStatusEntry
MAX-ACCESS  not-accessible
STATUS      deprecated
DESCRIPTION "<Definition> An entry in the lamp status table.<Superseded by>
ISO26048-1-Srsa:fdSrsaPortEntry.MLS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
"
INDEX { dmsLampIndex }
::= {dmsLampStatusTable 1}

DmsLampStatusEntry ::= SEQUENCE {
   dmsLampIndex        Integer32,
   dmsLampDescription  DisplayString,
   dmsLampMfrStatus    DisplayString,
   dmsLampStatus       INTEGER,
   dmsLampPixelTop     Integer32,
   dmsLampPixelLeft    Integer32,
   dmsLampPixelBottom  Integer32,
   dmsLampPixelRight   Integer32}
```

5.11.2.5.5.1 Lamp Index Parameter

```
dmsLampIndex  OBJECT-TYPE
SYNTAX      Integer32 (1..2040)
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Index of the lamp status table. The number of rows in this
table is equal to the value of the dmsLampNumRows object.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortIndex.MLS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
```

```
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.24.1.1"
::= { dmsLampStatusEntry 1 }
```

5.11.2.5.5.2 Lamp Description Parameter

```
dmsLampDescription  OBJECT-TYPE
SYNTAX  DisplayString (SIZE (0..64))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Human-readable description of the lamp. This value should
provide enough information for maintenance personnel to identify the physical
location of the lamp within the DMS.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortDescription.MLS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.24.1.2"
::= { dmsLampStatusEntry 2 }
```

5.11.2.5.5.3 Lamp Manufacturer-defined Status Parameter

```
dmsLampMfrStatus  OBJECT-TYPE
SYNTAX  DisplayString (SIZE (0..64))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the current manufacturer-defined status of the lamp.
This object allows a vendor to provide the operator with additional
information.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortTextStatus.MLS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.24.1.3"
::= { dmsLampStatusEntry 3 }
```

5.11.2.5.5.4 Lamp Status Parameter

```
dmsLampStatus  OBJECT-TYPE
SYNTAX      INTEGER {
                 noError (2),
                 stuckOff (3),
                 stuckOn (4) }
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the current status of the indicated lamp.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortStatus.MLS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.24.1.4"
::= { dmsLampStatusEntry 4 }
```

5.11.2.5.5.5 Lamp - Pixel Mapping Top Parameter

```
dmsLampPixelTop  OBJECT-TYPE
SYNTAX      Integer32 (1..65535)
MAX-ACCESS  read-only
STATUS      deprecated
```

```
DESCRIPTION
"<Definition> Indicates the topmost row of pixels served by this lamp. The
top-most row on the sign face is row 1.
<Superseded by> dmsSrsaPixelsTop.MLS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.24.1.5"
::= { dmsLampStatusEntry 5 }
```

5.11.2.5.5.6 Lamp - Pixel Mapping Left Parameter

```
dmsLampPixelLeft  OBJECT-TYPE
SYNTAX      Integer32 (1..65535)
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the leftmost column of pixels served by this lamp.
The left-most column on the sign face is column 1.
<Superseded by> dmsSrsaPixelsLeft.MLS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.24.1.6"
::= { dmsLampStatusEntry 6 }
```

5.11.2.5.5.7 Lamp - Pixel Mapping Bottom Parameter

```
dmsLampPixelBottom  OBJECT-TYPE
SYNTAX      Integer32 (1..65535)
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the bottommost row of pixels served by this lamp. The
top-most row on the sign face is row 1.
<Superseded by> dmsSrsaPixelsBottom.MLS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.24.1.7"
::= { dmsLampStatusEntry 7 }
```

5.11.2.5.5.8 Lamp - Pixel Mapping Right Parameter

```
dmsLampPixelRight  OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the rightmost column of pixels served by this lamp.
The left-most column on the sign face is column 1.
<Superseded by> dmsSrsaPixelsRight.MLS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.24.1.8"
::= { dmsLampStatusEntry 8 }
```

5.11.2.6 Drum Status Data

5.11.2.6.1 Drum Display Failure Status Map Parameter

```
dmsDrumStatusMap  OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE (0..2))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates whether each drum display subsystem within the sign
has failed. If a subsystem has failed, its associated bit is set to one (1).
The size of this object shall always present one bit for each drum supported
by the DMS, but shall not contain more than seven bits that are not
associated with any drum.
Further information about each failed subsystem may be found in the
dmsDrumStatusTable. If any bit within this object is set, then the 'drum sign
error' bit within the shortErrorStatus object shall also be set.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeWarning.MDS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.25"
::= { statError 25 }
```

5.11.2.6.2 Number of Rows in Drum Status Table Parameter

```
dmsDrumNumRows  OBJECT-TYPE
SYNTAX      Integer32 (0..16)
UNITS       "rows"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the number of rows in the dmsDrumStatusTable.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeCount.MDS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.26"
::= { statError 26 }
```

5.11.2.6.3 Drum Status Table Parameter

```
dmsDrumStatusTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF DmsDrumStatusEntry
MAX-ACCESS  not-accessible
STATUS      deprecated
DESCRIPTION "<Definition> A table containing status information for each drum
display unit within a DMS.
<Table Type> static
<Superseded by> ISO26048-1-Srsa:fdSrsaPortTable.MDS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.27"
::= {statError 27}

dmsDrumStatusEntry OBJECT-TYPE
SYNTAX      DmsDrumStatusEntry
MAX-ACCESS  not-accessible
STATUS      deprecated
DESCRIPTION "<Definition> An entry in the drum status table.<Superseded by>
ISO26048-1-Srsa:fdSrsaPortEntry.MDS
```

```
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object."
INDEX { dmsDrumIndex }
::= {dmsDrumStatusTable 1}

DmsDrumStatusEntry ::= SEQUENCE {
   dmsDrumIndex       Integer32,
   dmsDrumDescription DisplayString,
   dmsDrumMfrStatus   DisplayString,
   dmsDrumStatus      INTEGER}
```

5.11.2.6.3.1 Drum Index Parameter

```
dmsDrumIndex  OBJECT-TYPE
SYNTAX      Integer32 (1..16)
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Index of the drum status table. This index corresponds to the
bit position within the dmsDrumStatusMap bitmap: the row with index 1
corresponds to the low-order bit of the dmsDrumStatusMap, etc.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortIndex.MDS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.27.1.1"
::= { dmsDrumStatusEntry 1 }
```

5.11.2.6.3.2 Drum Description Parameter

```
dmsDrumDescription  OBJECT-TYPE
SYNTAX  DisplayString (SIZE (0..64))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Human-readable description of the drum. This value should
provide enough information for maintenance personnel to identify the physical
location of the drum within the DMS. The description shall include a
meaningful definition of the location where the sensor defined in this row is
located within the DMS.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortDescription.MDS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.27.1.2"
::= { dmsDrumStatusEntry 2 }
```

5.11.2.6.3.3 Drum Manufacturer-defined Status Parameter

```
dmsDrumMfrStatus  OBJECT-TYPE
SYNTAX  DisplayString (SIZE (0..64))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the current manufacturer-defined status of the drum.
This object allows a vendor to provide the operator with additional
information.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortTextStatus.MDS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
```

```
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.27.1.3"
::= { dmsDrumStatusEntry 3 }
```

5.11.2.6.3.4 Drum Status Parameter

```
dmsDrumStatus  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1),
                 noError (2),
                 interlockError (3),
                 stuckError (4),
                 positionError (5),
                 positionUnknownError (6) }
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the current status of the indicated drum.
<Format>
  noError - the drum is working properly.
  interlockError - the drum has failed to lock into a correct display
     position. It is hung up between two adjacent drum faces.
  stuckError - the drum cannot be moved from its present position, due
     to a problem with the drum mechanism.
  positionError - the drum has moved to a position other than the
     position requested by the DMS controller.
  positionUnknownError - the DMS controller cannot determine the
     position of the drum.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortValue.MDS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.27.1.4"
::= { dmsDrumStatusEntry 4 }
```

5.11.2.7 Light Sensor Status Data

5.11.2.7.1 Light Sensor Status Map Parameter

```
dmsLightSensorStatusMap  OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE (0..2))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the operational status of all light sensors. If a
light sensor is failed, the bit corresponding to the light sensor is set to
1; otherwise 0. The size of this object shall always present one bit for each
light sensor supported by the DMS, but shall not contain more than seven bits
that are not associated with any light sensor.
Each bit corresponds to an entry in the dmsLightSensorStatusTable. The low
order bit corresponds to the light sensor with dmsLightSensorIndex = 1.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeWarning.FAL
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.28"
::= { statError 28 }
```

5.11.2.7.2 Number of Rows in Light Sensor Status Table Parameter

```
dmsLightSensorNumRows  OBJECT-TYPE
SYNTAX      Integer32 (0..16)
UNITS       "rows"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the number of rows in the dmsLightSensorStatusTable.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeCount.FAL
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.29"
::= { statError 29 }
```

5.11.2.7.3 Light Sensor Status Table Parameter

```
dmsLightSensorStatusTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF DmsLightSensorStatusEntry
MAX-ACCESS  not-accessible
STATUS      deprecated
DESCRIPTION "<Definition> A table containing status information for each
light sensor within a DMS.
<Table Type> static
<Superseded by> ISO26048-1-Srsa:fdSrsaPortTable.FAL
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.30"
::= {statError 30}

dmsLightSensorStatusEntry OBJECT-TYPE
SYNTAX      DmsLightSensorStatusEntry
MAX-ACCESS  not-accessible
STATUS      deprecated
DESCRIPTION "<Definition> An entry in the light sensor status
table.<Superseded by> ISO26048-1-Srsa:fdSrsaPortEntry.FAL
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object."
INDEX { dmsLightSensorIndex }
::= {dmsLightSensorStatusTable 1}

DmsLightSensorStatusEntry ::= SEQUENCE {
   dmsLightSensorIndex          Integer32,
   dmsLightSensorDescription    DisplayString,
   dmsLightSensorCurrentReading Integer32,
   dmsLightSensorStatus         INTEGER }
```

5.11.2.7.3.1 Light Sensor Index Parameter

```
dmsLightSensorIndex  OBJECT-TYPE
SYNTAX      Integer32 (1..16)
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Index of the light sensor status table. This index corresponds
to the bit position within the dmsLightSensorStatusMap bitmap: the row with
index 1 corresponds to the low-order bit of the dmsLightSensorStatusMap, etc.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortIndex.FAL
```

```
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.30.1.1"
::= { dmsLightSensorStatusEntry 1 }
```

5.11.2.7.3.2 Light Sensor Description Parameter

```
dmsLightSensorDescription  OBJECT-TYPE
SYNTAX      DisplayString (SIZE (0..64))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Human-readable description of the light sensor. This value
should provide enough information for maintenance personnel to identify the
physical location of the light sensor within the DMS. The description shall
include a meaningful definition of the location where the sensor defined in
this row is located within the DMS.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortDescription.FAL
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.30.1.2"
::= { dmsLightSensorStatusEntry 2 }
```

5.11.2.7.3.3 Light Sensor Current Reading Parameter

```
dmsLightSensorCurrentReading  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the current reading of the light sensor. Total
darkness shall cause the current reading to be zero, and full sunlight shall
cause a reading of 65535. The light sensor reading shall be a linear
function; the DMS must perform any required scaling.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortValue.FAL
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.30.1.3"
::= { dmsLightSensorStatusEntry 3 }
```

5.11.2.7.3.4 Light Sensor Status Parameter

```
dmsLightSensorStatus  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1), -- deprecated
                 noError (2),
                 fail (3) }
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the current status of the indicated light sensor.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortStatus.FAL
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.30.1.4"
::= { dmsLightSensorStatusEntry 4 }
```

5.11.2.8 Humidity Data

5.11.2.8.1 Humidity Sensor Status Map Parameter

```
dmsHumiditySensorStatusMap  OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE (0..2))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the operational status of all humidity sensors. If a
humidity sensor is failed, the bit corresponding to the humidity sensor is
set to 1; otherwise 0.  The size of this object shall always present one bit
for each humidity sensor supported by the DMS, but shall not contain more
than seven bits that are not associated with any humidity sensor.
Each bit corresponds to an entry in the dmsHumiditySensorStatusTable. The low
order bit corresponds to the humidity sensor with dmsHumiditySensorIndex = 1.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeWarning.FAH and FEH
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object. The new design distinguishes between ambient
humidity and enclosure humidity.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.31"
::= { statError 31 }
```

5.11.2.8.2 Number of Rows in Humidity Sensor Status Table Parameter

```
dmsHumiditySensorNumRows  OBJECT-TYPE
SYNTAX      Integer32 (0..16)
UNITS       "rows"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the number of rows in the
dmsHumiditySensorStatusTable.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeCount.FAH and FEH
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object. The new design distinguishes between ambient
humidity and enclosure humidity.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.32"
::= { statError 32 }
```

5.11.2.8.3 Humidity Sensor Status Table Parameter

```
dmsHumiditySensorStatusTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF DmsHumiditySensorStatusEntry
MAX-ACCESS  not-accessible
STATUS      deprecated
DESCRIPTION "<Definition> A table containing status information for each
humidity sensor within a DMS.
<Table Type> static
<Superseded by> ISO26048-1-Srsa:fdSrsaPortTable.FAH and FEH
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object. The new design distinguishes between ambient
humidity and enclosure humidity.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.33"
::= {statError 33}

dmsHumiditySensorStatusEntry OBJECT-TYPE
```

```
SYNTAX      DmsHumiditySensorStatusEntry
MAX-ACCESS  not-accessible
STATUS      deprecated
DESCRIPTION "<Definition> An entry in the humidity sensor status
table.<Superseded by> ISO26048-1-Srsa:fdSrsaPortEntry.FAH and FEH
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object. The new design distinguishes between ambient
humidity and enclosure humidity."
INDEX { dmsHumiditySensorIndex }
::= { dmsHumiditySensorStatusTable 1}

DmsHumiditySensorStatusEntry ::= SEQUENCE {
   dmsHumiditySensorIndex          Integer32,
   dmsHumiditySensorDescription    DisplayString,
   dmsHumiditySensorCurrentReading Integer32,
   dmsHumiditySensorStatus         INTEGER }
```

5.11.2.8.3.1 Humidity Sensor Index Parameter

```
dmsHumiditySensorIndex  OBJECT-TYPE
SYNTAX      Integer32 (1..16)
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Index of the humidity sensor status table. This index
corresponds to the bit position within the dmsHumiditySensorStatusMap bitmap:
the row with index 1 corresponds to the low-order bit of the
dmsHumiditySensorStatusMap, etc.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortIndex.FAH and FEH
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object. The new design distinguishes between ambient
humidity and enclosure humidity.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.33.1.1"
::= { dmsHumiditySensorStatusEntry 1 }
```

5.11.2.8.3.2 Humidity Sensor Description Parameter

```
dmsHumiditySensorDescription  OBJECT-TYPE
SYNTAX      DisplayString (SIZE (0..64))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Human-readable description of the humidity sensor. This value
should provide enough information for maintenance personnel to identify the
physical location of the humidity sensor within the DMS. The description
shall include a meaningful definition of the location where the sensor
defined in this row is located within the DMS.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortDescription.FAH and FEH
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object. The new design distinguishes between ambient
humidity and enclosure humidity.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.33.1.2"
::= { dmsHumiditySensorStatusEntry 2 }
```

5.11.2.8.3.3 Humidity Sensor Current Reading Parameter

```
dmsHumiditySensorCurrentReading  OBJECT-TYPE
SYNTAX      Integer32 (0..100)
```

```
UNITS       "percent relative humidity"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the current reading of the humidity sensor, in
percent relative humidity.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortValue.FAH and FEH
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object. The new design distinguishes between ambient
humidity and enclosure humidity.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.33.1.3"
::= { dmsHumiditySensorStatusEntry 3 }
```

5.11.2.8.3.4 Humidity Sensor Status Parameter

```
dmsHumiditySensorStatus  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1), -- deprecated
                 noError (2),
                 fail (3) }
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the current status of the indicated humidity sensor.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortStatus.FAH and FEH
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object. The new design distinguishes between ambient
humidity and enclosure humidity.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.33.1.4"
::= { dmsHumiditySensorStatusEntry 4 }
```

5.11.2.9 Temperature Sensor Data

5.11.2.9.1 Temperature Sensor Status Map Parameter

```
dmsTempSensorStatusMap  OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE (0..2))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the operational status of all temperature sensors. If
a temperature sensor is failed, the bit corresponding to the temperature
sensor is set to 1; otherwise 0. The size of this object shall always present
one bit for each temperature sensor supported by the DMS, but shall not
contain more than seven bits that are not associated with any temperature
sensor.

Each bit corresponds to an entry in the dmsTempSensorStatusTable. The low
order bit corresponds to the temperature sensor with dmsTempSensorIndex = 1.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeWarning.FAT and FET
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object. The new design distinguishes between ambient
temperature and enclosure temperature.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.34"
::= { statError 34 }
```

5.11.2.9.2 Number of Rows in Temperature Sensor Status Table Parameter

```
dmsTempSensorNumRows  OBJECT-TYPE
SYNTAX      Integer32 (0..16)
UNITS       "rows"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the number of rows in the dmsTempSensorStatusTable.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeCount.FAT and FET
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object. The new design distinguishes between ambient
temperature and enclosure temperature.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.35"
::= { statError 35 }
```

5.11.2.9.3 Temperature Sensor Status Table Parameter

```
dmsTempSensorStatusTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF DmsTempSensorStatusEntry
MAX-ACCESS  not-accessible
STATUS      deprecated
DESCRIPTION "<Definition> A table containing status information for each
temperature sensor within a DMS.
<Table Type> static
<Superseded by> ISO26048-1-Srsa:fdSrsaPortTable.FAT and FET
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object. The new design distinguishes between ambient
temperature and enclosure temperature.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.36"
::= {statError 36}

dmsTempSensorStatusEntry OBJECT-TYPE
SYNTAX      DmsTempSensorStatusEntry
MAX-ACCESS  not-accessible
STATUS      deprecated
DESCRIPTION "<Definition> An entry in the temperature sensor status
table.<Superseded by> ISO26048-1-Srsa:fdSrsaPortEntry.FAT and FET
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object. The new design distinguishes between ambient
temperature and enclosure temperature."
INDEX { dmsTempSensorIndex }
::= { dmsTempSensorStatusTable 1}

DmsTempSensorStatusEntry ::= SEQUENCE {
   dmsTempSensorIndex                  Integer32,
   dmsTempSensorDescription            DisplayString,
   dmsTempSensorCurrentReading         ITSInteger8,
   dmsTempSensorHighWarningTemperature  ITSInteger8,
   dmsTempSensorLowWarningTemperature   ITSInteger8,
   dmsTempSensorHighCriticalTemperature ITSInteger8,
   dmsTempSensorLowCriticalTemperature  ITSInteger8,
   dmsTempSensorStatus                 INTEGER }
```

5.11.2.9.3.1 Temperature Sensor Index Parameter

```
dmsTempSensorIndex  OBJECT-TYPE
SYNTAX      Integer32 (1..16)
MAX-ACCESS  read-only
```

```
STATUS      deprecated
DESCRIPTION
"<Definition> Index of the temperature sensor status table. This index
corresponds to the bit position within the dmsTempSensorStatusMap bitmap: the
row with index 1 corresponds to the low-order bit of the
dmsTempSensorStatusMap, etc.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortIndex.FAT and FET
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object. The new design distinguishes between ambient
temperature and enclosure temperature.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.36.1.1"
::= { dmsTempSensorStatusEntry 1 }
```

5.11.2.9.3.2 Temperature Sensor Description Parameter

```
dmsTempSensorDescription  OBJECT-TYPE
SYNTAX      DisplayString (SIZE (0..64))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Human-readable description of the temperature sensor. This
value should provide enough information for maintenance personnel to identify
the physical location of the temperature sensor within the DMS.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortDescription.FAT and FET
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object. The new design distinguishes between ambient
temperature and enclosure temperature.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.36.1.2"
::= { dmsTempSensorStatusEntry 2 }
```

5.11.2.9.3.3 Temperature Sensor Current Reading Parameter

```
dmsTempSensorCurrentReading  OBJECT-TYPE
SYNTAX      ITSInteger8
UNITS       "degrees Celsius"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the current reading of the temperature sensor in full
degrees Celsius.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortValue.FAT and FET
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object. The new design distinguishes between ambient
temperature and enclosure temperature.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.36.1.3"
::= { dmsTempSensorStatusEntry 3 }
```

5.11.2.9.3.4 Temperature Sensor High Warning Temperature Parameter

```
dmsTempSensorHighWarningTemperature  OBJECT-TYPE
SYNTAX      ITSInteger8
UNITS       "degrees Celsius"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the high value of the temperature associated with
this temperature sensor that generates a warning, in full degrees Celsius.
```

```
This value should not be lower than the value of the
dmsTempSensorLowWarningTemperature object.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortMaxThreshold.FAT and FET
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object. The new design distinguishes between ambient
temperature and enclosure temperature.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.36.1.4"
::= { dmsTempSensorStatusEntry 4 }
```

5.11.2.9.3.5 Temperature Sensor Low Warning Temperature Parameter

```
dmsTempSensorLowWarningTemperature  OBJECT-TYPE
SYNTAX      ITSInteger8
UNITS       "degrees Celsius"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the low value of the temperature associated with this
temperature sensor that generates a warning, in full degrees Celsius. This
value should not be higher than the value of the
dmsTempSensorHighWarningTemperature object.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortMinThreshold.FAT and FET
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object. The new design distinguishes between ambient
temperature and enclosure temperature.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.36.1.5"
::= { dmsTempSensorStatusEntry 5 }
```

5.11.2.9.3.6 Temperature Sensor High Critical Temperature Parameter

```
dmsTempSensorHighCriticalTemperature  OBJECT-TYPE
SYNTAX      ITSInteger8
UNITS       "degrees Celsius"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the high value of the critical temperature associated
with this temperature sensor, in full degrees Celsius. This value shall not
be lower than the value of the dmsTempSensorLowCriticalTemperature object.
<Superseded by> ISO26048-1-CondTrigger:fdCondTriggerValue where
fdCondTriggerActionGroup calls an entry that includes
dmsCallableSignDisplayShutdown.
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object. The new design distinguishes between ambient
temperature and enclosure temperature.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.36.1.6"
::= { dmsTempSensorStatusEntry 6 }
```

5.11.2.9.3.7 Temperature Sensor Low Critical Temperature Parameter

```
dmsTempSensorLowCriticalTemperature  OBJECT-TYPE
SYNTAX      ITSInteger8
UNITS       "degrees Celsius"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the low value of the critical temperature associated
```

```
with this temperature sensor, in full degrees Celsius. This value shall not
be higher than the value of the dmsTempSensorHighCriticalTemperature object.
<Superseded by> ISO26048-1-CondTrigger:fdCondTriggerValue2 where
fdCondTriggerActionGroup2 calls an entry that includes
dmsCallableSignDisplayRestart.
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
an entry in the Conditional Trigger Table that enables the functionality.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.36.1.7"
::= { dmsTempSensorStatusEntry 7 }
```

5.11.2.9.3.8 Temperature Sensor Status Parameter

```
dmsTempSensorStatus  OBJECT-TYPE
SYNTAX      INTEGER {
              other (1),
              noError (2),
              fail (3) }
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the current status of the indicated temperature
sensor.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortStatus.FAT and FET
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object. The new design distinguishes between ambient
temperature and enclosure temperature.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.36.1.8"
::= { dmsTempSensorStatusEntry 8 }
```

5.11.2.9.4 Temperature Sensor Highest Critical Temperature Parameter

```
dmsTempSensorHighestCriticalTempThreshold  OBJECT-TYPE
SYNTAX      ITSInteger8
UNITS       "degrees Celsius"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the highest value of the critical temperature
threshold associated with any of the supported temperature sensors in the
DMS, in full degrees Celsius. This value shall not be lower than any of the
high critical values of any of the ISO26048-1-Srsa:fdSrsaPortMaxValue.FET
objects within the ISO26048-1-Srsa:fdSrsaPortTable.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.37"
::= { statError 37 }
```

5.11.2.9.5 Temperature Sensor Lowest Critical Temperature Parameter

```
dmsTempSensorLowestCriticalTempThreshold  OBJECT-TYPE
SYNTAX      ITSInteger8
UNITS       "degrees Celsius"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the lowest value of the critical temperature
threshold associated with any of the supported temperature sensors in the
DMS, in full degrees Celsius. This value shall not be higher than any of the
```

```
low critical values of any of the ISO26048-1-Srsa:fdSrsaPortMinValue.FET
objects within the Srsa:fdSrsaPortTable.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.7.38"
::= { statError 38 }
```

5.11.3 -Power Status Objects

```
statPower  OBJECT-IDENTITY
STATUS  deprecated
DESCRIPTION
"<Definition> This node is an identifier used to group all objects supporting
DMS sign power status monitoring functions that are common to DMS devices.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.8"
::= { dmsStatus 8 }
```

5.11.3.1 Sign Volts Parameter

```
signVolts  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "hundredths of a volt"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> A voltage measurement in units of hundredth (1/100) of a volt.
The maximum value (0xFFFF) corresponds to a voltage of 655.35 volts. This is
an indication of the sign battery voltage.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortValue.FBV
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.8.1"
::= { statPower 1 }
```

5.11.3.2 Low Fuel Threshold Parameter

```
lowFuelThreshold  OBJECT-TYPE
SYNTAX      Integer32 (0..100)
UNITS       "percent"
MAX-ACCESS  read-write
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the low fuel level threshold used to alert the user.
The threshold is indicated as a percent (%) of a full tank. When the level of
fuel is below the threshold, the bit for power alarm (bit 2) in the
shortErrorStatus-object shall be set to one (1).
<Superseded by> ISO26048-1-Srsa:fdSrsaPortMinThreshold.FGF
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.8.2"
::= { statPower 2 }
```

5.11.3.3 Fuel Level Parameter

```
fuelLevel  OBJECT-TYPE
SYNTAX      Integer32 (0..100)
UNITS       "percent"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
```

```
"<Definition> A number indicating the amount of fuel remaining, specified as
a percent (%) of a full tank.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortValue.FGF
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.8.3"
::= { statPower 3 }
```

5.11.3.4 Engine RPM Parameter

```
engineRPM  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "rpm"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the engine rpm in units of 100. This provides a range
from 0 rpm to 25500 rpm.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortValue.FGS
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.8.4"
::= { statPower 4 }
```

5.11.3.5 Line Volts Parameter

```
lineVolts  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "volts"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> The DMS line voltage measurement in (1.0) volts. The range is 0
volts to 255 volts.
<Superseded by> ISO26048-1-Srsa:fdSrsaPortValue.FLV
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal SRSA object.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.8.5"
::= { statPower 5 }
```

5.11.3.6 Power Source Parameter

```
powerSource  OBJECT-TYPE
SYNTAX      INTEGER {
                 other (1),
                 powerShutdown (2),
                 noSignPower (3),
                 acLine (4),
                 generator (5),
                 solar (6),
                 batteryUPS (7) }
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the source of power that is currently utilized by the
sign.
<Format>
```

```
  other: indicates that the sign is powered by a method not listed
     below (see device manual);
  powerShutdown: indicates that there is just enough power to perform
     shutdown activities.
  noSignPower: indicates that the sign controller has power but the
     sign display has no power;
  acLine: indicates that the controller and sign is powered by AC
     power;
  generator: indicates that the sign and the controller are powered by
     a generator;
  solar: indicates that the sign and the controller are powered by solar
     equipment;
  battery-UPS: indicates that the sign and controller are powered by
     battery or UPS with no significant charging occurring.
<Superseded by> ISO26048-1-Srsa: fdPowerSource
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal object from ISO 26048-1.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.8.6"
::= { statPower 6 }
```

5.11.4 Temperature Status Objects

```
statTemp  OBJECT-IDENTITY
STATUS  deprecated
DESCRIPTION
"<Definition> This node is an identifier used to group all objects supporting
DMS sign temperature status monitoring functions that are common to DMS
devices.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.9"
::= { dmsStatus 9 }
```

5.11.4.1 Minimum Temperature of Control Cabinet Parameter

```
tempMinCtrlCabinet  OBJECT-TYPE
SYNTAX      ITSInteger8
UNITS       "degrees Celsius"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the current temperature (single sensor) or the
current minimum temperature (multiple sensors) within the DMS Control Cabinet
in degrees Celsius.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeValueStatsMin.FET
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal object from ISO 26048-1. The new object combines sign
housing and the control cabinet into a single value.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.9.1"
::= { statTemp 1 }
```

5.11.4.2 Maximum Temperature of Control Cabinet Parameter

```
tempMaxCtrlCabinet  OBJECT-TYPE
SYNTAX      ITSInteger8
UNITS       "degrees Celsius"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the current temperature (single sensor) or the
```

```
current maximum temperature (multiple sensors) within the DMS Control Cabinet
in degrees Celsius.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeValueStatsMax.FET
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal object from ISO 26048-1. The new object combines sign
housing and the control cabinet into a single value.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.9.2"
::= { statTemp 2 }
```

5.11.4.3 Minimum Ambient Temperature Parameter

```
tempMinAmbient  OBJECT-TYPE
SYNTAX      ITSInteger8
UNITS       "degrees Celsius"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the current outside ambient temperature (single
sensor) or the current minimum outside ambient temperature (multiple sensors)
in degrees Celsius.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeValueStatsMin.FAT
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal object from ISO 26048-1.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.9.3"
::= { statTemp 3 }
```

5.11.4.4 Maximum Ambient Temperature Parameter

```
tempMaxAmbient  OBJECT-TYPE
SYNTAX      ITSInteger8
UNITS       "degrees Celsius"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the current outside ambient temperature (single
sensor) or the current maximum outside ambient temperature (multiple sensors)
in degrees Celsius.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeValueStatsMax.FAT
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal object from ISO 26048-1.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.9.4"
::= { statTemp 4 }
```

5.11.4.5 Minimum Temperature of Sign Housing Parameter

```
tempMinSignHousing  OBJECT-TYPE
SYNTAX      ITSInteger8
UNITS       "degrees Celsius"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the current temperature (single sensor) or the
current minimum temperature (multiple sensors) in the sign housing in degrees
Celsius.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeValueStatsMin.FET
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal object from ISO 26048-1. The new object combines sign
housing and the control cabinet into a single value.
```

```
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.9.5"
::= { statTemp 5 }
```

5.11.4.6 Maximum Temperature of Sign Housing Parameter

```
tempMaxSignHousing  OBJECT-TYPE
SYNTAX      ITSInteger8
UNITS       "degrees Celsius"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates the current temperature (single sensor) or the
current maximum temperature (multiple sensors) in the sign housing in degrees
Celsius.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeValueStatsMax.FET
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal object from ISO 26048-1. The new object combines sign
housing and the control cabinet into a single value.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.9.6"
::= { statTemp 6 }
```

5.11.4.7 Temperature Sensor Warning Parameter

```
tempSensorWarningMap  OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE (0..2))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates whether each temperature sensor has exceeded a
dmsTempSensorHighWarningTemperature or dmsTempSensorLowWarningTemperature
value. If a temperature sensor has exceeded the defined value, the bit
corresponding to the temperature sensor is set to 1; otherwise 0. The mapping
of bits to individual sensors is manufacturer specific. This bitmap of this
object shall be configured as defined in the dmsTempSensorStatusTable in that
the first bit of this object shall correspond to the first row in that table,
the second bit shall correspond to the second row, and so forth.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeWarning.FET
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal object from ISO 26048-1.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.9.7"
::= { statTemp 7 }
```

5.11.4.8 Critical Temperature Map Parameter

```
tempSensorCriticalTempMap  OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE (0..2))
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> Indicates whether each temperature sensor has exceeded the
dmsTempSensorHighCriticalTemperature or the
dmsTempSensorLowCriticalTemperature threshold. If a temperature sensor has
exceeded the defined value, the bit corresponding to the temperature sensor
is set to 1; otherwise 0. The mapping of bits to individual sensors is
manufacturer specific. This bitmap of this object shall be configured as
defined in the dmsTempSensorStatusTable in that the first bit of this object
```

```
shall correspond to the first row in that table, the second bit shall
correspond to the second row, and so forth.
<Superseded by> ISO26048-1-Srsa:fdSrsaTypeStatus.FET
<Informative> This object was deprecated in NTCIP 1203 v04 and replaced with
the more universal object from ISO 26048-1.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.9.8"
::= { statTemp 8 }
```

5.11.5 SRSA Extension Objects

```
dmsSrsaExtensions  OBJECT-IDENTITY
STATUS  current
DESCRIPTION
"<Definition> This node is an identifier used to group all DMS-specific
extensions to the ISO 26048-1 SRSA table..
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.10"
::= { dmsStatus 10 }
```

5.11.5.1 SRSA Pixels Table Parameter

```
dmsSrsaPixelsTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF DmsSrsaPixelsEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "<Definition> A table that extends the fdSrsaPortTable to
identify the pixels associated with an entry in the SRSA table.
<Informative> For example, if the SRSA entry is for lamp status, this table
will indicate the pixels associated with the lamp. When the lamp experiences
an error, the management station can determine the affected pixels.
One row shall exist in this table for each SRSA port with a type of 'MLS'.
Rows may exist in this table for other items; for example, power supplies
('FPS') can be associated with groups of pixels.
<Table Type> static
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.10.1"
::= { dmsSrsaExtensions 1}

dmsSrsaPixelsEntry  OBJECT-TYPE
SYNTAX      DmsSrsaPixelsEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "<Definition> An entry in the lamp SRSA details table."
INDEX { fdSrsaTypeCode, fdSrsaPortIndex }
::= { dmsSrsaPixelsTable 1}

DmsSrsaPixelsEntry ::= SEQUENCE {
   dmsSrsaPixelsTop     Integer32,
   dmsSrsaPixelsLeft    Integer32,
   dmsSrsaPixelsBottom  Integer32,
   dmsSrsaPixelsRight   Integer32}
```

5.11.5.1.1 SRSA Top Pixel Parameter

```
dmsSrsaPixelsTop  OBJECT-TYPE
SYNTAX      Integer32 (1..65535)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the topmost row of pixels served by this entry. The
top-most row on the sign face is row 1.
```

```
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.10.1.1.1"
::= { dmsSrsaPixelsEntry 1 }
```

5.11.5.1.2 SRSA Left Pixel Parameter

```
dmsSrsaPixelsLeft  OBJECT-TYPE
SYNTAX      Integer32 (1..65535)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the leftmost column of pixels served by this entry.
The left-most column on the sign face is column 1.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.10.1.1.2"
::= { dmsSrsaPixelsEntry 2 }
```

5.11.5.1.3 SRSA Bottom Pixel Parameter

```
dmsSrsaPixelsBottom  OBJECT-TYPE
SYNTAX      Integer32 (1..65535)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the bottommost row of pixels served by this entry.
The top-most row on the sign face is row 1.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.10.1.1.3"
::= { dmsSrsaPixelsEntry 3 }
```

5.11.5.1.4 SRSA Right Pixel Parameter

```
dmsSrsaPixelsRight  OBJECT-TYPE
SYNTAX      Integer32
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the rightmost column of pixels served by this entry.
The left-most column on the sign face is column 1.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.9.10.1.1.4"
::= { dmsSrsaPixelsEntry 4 }
```

5.12 Graphic Definition Objects

```
graphicDefinition  OBJECT-IDENTITY
STATUS  current
DESCRIPTION
"<Definition> This node is an identifier used to group all objects for DMS
graphic configurations that are common to DMS devices.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10"
::= { dms 10 }
```

5.12.1 Maximum Number of Graphics Parameter

```
dmsGraphicMaxEntries  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "graphics"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
```

```
"<Definition> Indicates the maximum number of graphics that the sign can
store.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.1"
::= { graphicDefinition 1 }
```

5.12.2 Number of Graphics Parameter

```
dmsGraphicNumEntries  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "graphics"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the number of graphics currently stored within the
sign.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.2"
::= { graphicDefinition 2 }
```

5.12.3 Maximum Graphic Size Parameter

```
dmsGraphicMaxSize  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "octets"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the maximum size (in bytes) of each graphic the sign
is capable of storing. This value shall be an even multiple of the object
dmsGraphicBlockSize.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.3"
::= { graphicDefinition 3 }
```

5.12.4 Available Graphic Memory Parameter

```
availableGraphicMemory  OBJECT-TYPE
SYNTAX      Counter32
UNITS       "octets"
MAX-ACCESS  read-only
STATUS      deprecated
DESCRIPTION
"<Definition> An indication of the amount of memory left, in bytes, to store
graphics.
<Superseded by> availableGraphicMemoryV4
<Informative> This object was deprecated in NTCIP 1203 v04 because it does
not fulfill the semantics of a 'Counter', which is more rigidly enforced in
SNMPv3.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.4"
::= { graphicDefinition 4 }
```

5.12.5 Graphic Block Size Parameter

```
dmsGraphicBlockSize  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
UNITS       "octets"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
```

```
"<Definition> Indicates the size of each block within each graphic bitmap
image. A graphic bitmap may consist of at most
dmsGraphicMaxSize/dmsGraphicBlockSize number of blocks.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.5"
::= { graphicDefinition 5 }
```

5.12.6 Graphics Table Parameter

```
dmsGraphicTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF DmsGraphicEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "<Definition> A table containing the information needed to
configure/define a particular graphic. The values of a columnar object
(except the dmsGraphicStatus) cannot be changed when the 'dmsGraphicStatus'-
object of that particular row is any value other than 'modifying'.
<Table Type> static
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.6"
::= {graphicDefinition 6}

dmsGraphicEntry OBJECT-TYPE
SYNTAX      DmsGraphicEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "<Definition> Parameters of the Graphic Table."
INDEX {dmsGraphicIndex}
::= {dmsGraphicTable 1}

DmsGraphicEntry ::= SEQUENCE {
      dmsGraphicIndex   Integer32,
      dmsGraphicNumber  Integer32,
      dmsGraphicName    SnmpAdminString,
      dmsGraphicHeight  Integer32,
      dmsGraphicWidth   Integer32,
      dmsGraphicType    INTEGER,
      dmsGraphicID      Integer32,
      dmsGraphicTransparentEnabled  Integer32,
      dmsGraphicTransparentColor    DmsColor,
      dmsGraphicStatus  INTEGER}
```

5.12.6.1 Graphic Index Parameter

```
dmsGraphicIndex  OBJECT-TYPE
SYNTAX      Integer32 (1..255)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the row number of the entry. This index directly
corresponds to dmsGraphicBitmapIndex located in dmsGraphicBitmapTable. The
storage for each graphic of this table is located in dmsGraphicBitmapTable.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.6.1"
::= { dmsGraphicEntry 1 }
```

5.12.6.2 Graphic Number Parameter

```
dmsGraphicNumber  OBJECT-TYPE
SYNTAX      Integer32 (1..255)
MAX-ACCESS  read-write
```

```
STATUS      current
DESCRIPTION
"<Definition> A unique, user-specified number for a particular graphic which
can be different from the value of the dmsGraphicIndex-object. This is the
number referenced by MULTI when specifying a particular graphic. A device
shall return a inconsistentValue error, if this value is not unique.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.6.2"
::= { dmsGraphicEntry 2 }
```

5.12.6.3 Graphic Name Parameter

```
dmsGraphicName  OBJECT-TYPE
SYNTAX      SnmpAdminString (SIZE (0..64))
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the name of the graphic.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.6.3"
::= { dmsGraphicEntry 3 }
```

5.12.6.4 Graphic Height Parameter

```
dmsGraphicHeight  OBJECT-TYPE
SYNTAX      Integer32 (1..255)
UNITS       "pixels"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the height of the graphic in pixels. The value of
this object shall not exceed vmsSignHeightPixels.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.6.4"
::= { dmsGraphicEntry 4 }
```

5.12.6.5 Graphic Width Parameter

```
dmsGraphicWidth  OBJECT-TYPE
SYNTAX      Integer32 (1..65535)
UNITS       "pixels"
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the width of the graphic in pixels. The value of this
object shall not exceed vmsSignWidthPixels.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.6.5"
::= { dmsGraphicEntry 5 }
```

5.12.6.6 Graphic Type Parameter

```
dmsGraphicType  OBJECT-TYPE
SYNTAX      DmsColorScheme
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the type of the graphic stored in this row.
For definitions of the values see the dmsColorScheme object.  All DMS shall
support the monochrome1bit graphic type.
<Parameter Type> Configuration
```

```
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.6.6"
::= { dmsGraphicEntry 6 }
```

5.12.6.7 Graphic ID Parameter

```
dmsGraphicID  OBJECT-TYPE
SYNTAX      Integer32 (0..65535)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Each graphic that has been downloaded to a sign shall have a
relatively unique ID. This ID shall be calculated using the CRC-16 algorithm
defined in ISO 3309 and the associated OER-encoded (as defined in ITU-T
X.696) GraphicInfoList.

The following definitions are used to define the above referenced
GraphicInfoList.

Complete definitions for these referenced objects, including size
information, is contained elsewhere in this document.

GraphicInfoList ::= SEQUENCE {
        number INTEGER(1..255),
          -- dmsGraphicNumber of the subject graphic
        height INTEGER(1..65535),
          -- dmsGraphicHeight of the subject graphic
        width INTEGER(1..65535),
          -- dmsGraphicWidth of the subject graphic
        type INTEGER (1..4),
          -- dmsGraphicType of the subject graphic
        transparentEnabled INTEGER(0..1)
          -- dmsGraphicTransparentEnabled of the graphic
        transparentColor OCTET STRING (SIZE(3))
          -- dmsGraphicTransparentColor of the graphic if dmsGraphicType not
          -- color24bit, first octet is the transparent color and remaining
          -- octets are zero
        bitmap OCTET STRING (SIZE (Z))
          -- the bitmap of the subject graphic
        }

where   Z = ((height * width) + 7) / 8  -- for monochrome1bit
                                        --(remaining bits are set to zero)
        Z = (height * width) -- for monochrome8bit or colorClassic
        Z = (height * width) * 3 -- for color24bit

This gives a predictable byte stream for the GraphicInfoList:
[number]             -- 1 octet
[height]             -- 2 octets, MSB first
[width]              -- 2 octets, MSB first
[type]               -- 1 octet
[transparentEnabled] -- 1 octet
[transparentcolor]   -- 3 octets (last 2 octets set to 0 if not color24bit)
[bitmap]             -- Z octets, according to height, width & color scheme

Examples:
Given the 10x6 bitmap - 84 92 63 08 C2 48 A1 70 =
@OOOO@OO@O
O@OO@OO@@O
```

```
OO@@OOOO@O
OO@@OOOO@O
O@OO@OOO@O
@OOOO@O@@@
OOOO

1)   dmsGraphicNumber = 3
     dmsGraphicHeight = 6
     dmsGraphicWidth = 10
     dmsGraphicType = 1
     dmsGraphicTransparentEnabled = 0
     dmsGraphicTransparentColor = 1
     dmsGraphicBitmap = 84 92 63 08 C2 48 A1 70

GraphicInfoList = 03 00 06 00 0A 01 00 01 00 00 84 92 63 08 C2 48 A1 70
dmsGraphicID = 0xB95A

where                03 = dmsGraphicNumber
                  00 06 = dmsGraphicHeight
                  00 0A = dmsGraphicWidth
                     01 = dmsGraphicType (monochrome1bit)
                     00 = dmsGraphicTransparentEnabled
              01  00 00 = dmsGraphicTransparentColor
84 92 63 08 C2 48 A1 70 = dmsGraphicBitmap

2)   dmsGraphicNumber = 4
     dmsGraphicHeight = 6
     dmsGraphicWidth = 10
     dmsGraphicType = 1
     dmsGraphicTransparentEnabled = 0
     dmsGraphicTransparentColor = 0
     dmsGraphicBitmap = 84 92 63 08 C2 48 A1 70

GraphicInfoList = 04 00 06 00 0A 01 00 00 00 00 84 92 63 08 C2 48 A1 70
dmsGraphicID = 0xBFF5

where                04 = dmsGraphicNumber
                  00 06 = dmsGraphicHeight
                  00 0A = dmsGraphicWidth
                     01 = dmsGraphicType (monochrome1bit)
                     00 = dmsGraphicTransparentEnabled
               00 00 00 = dmsGraphicTransparentColor
84 92 63 08 C2 48 A1 70 = dmsGraphicBitmap

3) A 4x4 pixel color classic graphic of the letter 'Z' in red

     dmsGraphicNumber = 5
     dmsGraphicHeight = 4
     dmsGraphicWidth = 4
     dmsGraphicType = 3
     dmsGraphicTransparentEnabled = 1
     dmsGraphicTransparentColor = 7
     dmsGraphicBitmap = 01 01 01 01 07 07 01 07 07 01 07 07 01 01 01 01

GraphicInfoList = 05 00 04 00 04 03 01 07 00 00 01 01 01 01 07 07 01 07 07 01
07 07 01 01 01 01
dmsGraphicID = 0x8FE0
```

```
where                05 = dmsGraphicNumber
                     00 04 = dmsGraphicHeight
                     00 04 = dmsGraphicWidth
                     03 = dmsGraphicType (colorClassic)
                     01 = dmsGraphicTransparentEnabled
                     07 00 00 = dmsGraphicTransparentColor (white)
01 01 01 01 07 07 01 07 07 01 07 07 01 01 01 01 = dmsGraphicBitmap

4) -- a 2x2 pixel square RGB graphic with green as the transparent color in
the lower left pixel
     dmsGraphicNumber = 7
     dmsGraphicHeight = 2
     dmsGraphicWidth = 2
     dmsGraphicType = 4
     dmsGraphicTransparentEnabled = 1
     dmsGraphicTransparentColor = 00 FF 00
     dmsGraphicBitmap = FFFFFF FF00FF 00FF00 FF00FF

GraphicInfoList = 07 00 02 00 02 04 01 00 FF 00 FFFFFF FF00FF 00FF00 FF00FF
dmsGraphicID = 0x078D

where                    07 = dmsGraphicNumber
                      00 02 = dmsGraphicHeight
                      00 02 = dmsGraphicWidth
                         04 = dmsGraphicType (color24bit)
                         01 = dmsGraphicTransparentEnabled
                   00 FF 00 = dmsGraphicTransparentColor (green)
FFFFFF FF00FF 00FF00 FF00FF = dmsGraphicBitmap

<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.6.7"
::= { dmsGraphicEntry 7 }
```

5.12.6.8 Graphic Transparent Enabled Parameter

```
dmsGraphicTransparentEnabled  OBJECT-TYPE
SYNTAX      Integer32 (0..1)
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates whether the graphic contains a color that is
considered transparent. A value of 0 means there is no transparent color.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.6.8"
::= { dmsGraphicEntry 8 }
```

5.12.6.9 Graphic Transparent Color Parameter

```
dmsGraphicTransparentColor  OBJECT-TYPE
SYNTAX      DmsColor
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> If dmsGraphicTransparentEnabled indicates that the graphic
contains a transparent color, this object specifies the color. All pixels in
the graphic that exactly match this color shall be considered transparent
such that when the graphic is displayed on the sign, those transparent pixels
will be left at whatever color exists on the message beneath the graphic (or
before the graphic is 'painted' onto the sign). The format of this color is
defined by the DmsColor textual convention.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.6.9"
::= { dmsGraphicEntry 9 }
```

5.12.6.10 Graphic Status Parameter

```
dmsGraphicStatus  OBJECT-TYPE
SYNTAX      INTEGER {
                 notUsed (1),
                 modifying (2),
                 calculatingID (3),
                 readyForUse (4),
                 inUse (5),
                 permanent (6),
                 modifyReq (7),
                 readyForUseReq (8),
                 notUsedReq (9) }
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> Indicates the current state of the graphic. This state-machine
allows for defining a graphic, readying a graphic (making it usable by a
message), and preventing its modification. See Section 4.3.2 for additional
state-machine requirements.

If dmsGraphicStatus is set to a value of notUsedReq (9), as this state-
machine transitions to the state of notUsed (1) the device shall release all
memory space allocated to that graphic bitmap.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.6.10"
::= { dmsGraphicEntry 10 }
```

5.12.7 Graphics Bitmap Table Parameter

```
dmsGraphicBitmapTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF DmsGraphicBitmapEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "<Definition> A table containing the bitmap information for the
graphic entries within the dmsGraphicTable. The values of a columnar object
in this table cannot be changed when the 'dmsGraphicStatus' object of the
corresponding row (row with the same index) in the dmsGraphicTable is any
value other than 'modifying'.

The dmsGraphicBitmapTable permits a complete bitmap to be downloaded to a
sign in pieces small enough to fit within a single SNMP SetRequest. Each
dmsGraphicBlockSize-sized chunk within the bitmap is represented by a
distinct row of the bitmap data Table; that is, the table is indexed by (1)
the dmsGraphicIndex, and (2) the dmsGraphicBlockNumber of a particular data
block within the graphic's data. Note that this mechanism is purely a
piecewise transfer method layered atop the bitmap data; when all the blocks
of a graphic image are downloaded, they must, as a group, conform to the
format described below. For a particular value of dmsGraphicIndex, the bitmap
defining the graphic image can be constructed in the following manner:
Concatenate the rows of the dmsGraphicBitmapTable with the matching
dmsGraphicIndex, in order of the dmsGraphicBlockNumber value, until the
resultant OCTET STRING is large enough to contain the entire image as defined
```

by the dmsGraphicHeight and dmsGraphicWidth values for the dmsGraphicIndex in question.

The format of a complete bitmap is described here:
A complete bitmap image is defined by the rows in dmsGraphicBitmapTable that share a common dmsGraphicIndex, as defined above. A bitmap image denotes the color of each pixel within a rectangular region. The size of the rectangular region is defined by the dmsGraphicHeight and dmsGraphicWidth objects.

If dmsGraphicType is a value of 'monochrome1bit', each bit within the bitmap data corresponds to a pixel on the DMS display. Starting with the first byte of the bitmap data, the most significant bit defines the state of the pixel in the upper left corner of the rectangular region. Byte 1 through byte N of the bitmap data corresponds to the rectangular region by rows, left to right, then top to bottom. If the rectangular region is not divisible by 8, the remaining bits shall be 0 and contained in the lower bits of the last byte of the bitmap data. In this case, the total size of the bitmap image in bytes is given by this formula:

```
  B = ((dmsGraphicWidth * dmsGraphicHeight) + 7)/8
```

The first term computes the approximate size of the bitmap in bytes, +/- one byte. The second term computes whether the size of the bitmap in bits is divisible by 8; if not, an extra byte is required to hold the remaining few bits.

If the dmsColorScheme is monochrome8bit, colorClassic, or color24bit, and the graphic is defined with dmsGraphicType monochrome1bit, then pixels are displayed with the foreground color (bit = 1) or black (bit = 0). Note that if dmsGraphicTransparentEnabled is set to 1 and dmsGraphicTransparentColor is set to 0 then a monochrome1bit graphic can be displayed in the same manner as a font character.

If dmsGraphicType is a value of 'monochrome8bit', each byte within the bitmap data corresponds to a pixel on the DMS display. The first byte of the bitmap data defines the state of the pixel in the upper left corner of the rectangular region. Byte 1 through byte N of the bitmap data correspond to the rectangular region by rows, left to right, then top to bottom. Each byte is one of 255 shades of the monochrome color. In this case, the formula for the total size of the bitmap in bytes is given by this formula:

```
  B = (dmsGraphicWidth*dmsGraphicHeight)
```

If dmsGraphicType is a value of 'colorClassic', each byte within the bitmap data corresponds to a pixel on the DMS display. The first byte of the bitmap data defines the state of the pixel in the upper left corner of the rectangular region. Byte 1 through byte N of the bitmap data correspond to the rectangular region by rows, left to right, then top to bottom. The data in each byte shall be one of the values indicated by dmsColorScheme under the colorClassic type. In this case, the formula for the total size of the bitmap in bytes is given by this formula:

```
  B = (dmsGraphicWidth*dmsGraphicHeight)
```

If dmsGraphicType is a value of 'color24bit, sets of three bytes within the bitmap data correspond to a pixel on the DMS display. The first three bytes of the bitmap data define the state of the pixel in the upper left corner of the rectangular region. Byte 1 through byte N of the bitmap data corresponds to the rectangular region by rows, left to right, then top to bottom. The first byte of the bitmap data shall be the value of blue for the upper left pixel. The second byte of the bitmap data shall be the value of green for the

```
upper left pixel. The third byte of the bitmap data shall be the value of red
for the upper left pixel. In this case, the formula for the total size of the
bitmap in bytes is given by this formula:
  B = (dmsGraphicWidth*dmsGraphicHeight)*3

All rows of the bitmap data Table must always logically exist (that is, under
no circumstances shall a controller produce a SuchInstance value when asked
for a row of the dmsGraphicBitmapTable where
dmsGraphicIndex<=dmsGraphicsMaxEntries and
dmsGraphicBlockNumber<=dmsGraphicMaxSize/dmsGraphicBlockSize). If a GET
request is received for a block for which no corresponding SET request has
been accepted, then the controller shall return a block of length
dmsGraphicBlockSize, each octet of which has the value 0 (zero). Similarly,
when displaying a bitmap, the contents of any block within the bitmap image
that has not been defined by a SET operation shall be assumed to be a
sequence of octets with the value 0 (zero) and length dmsGraphicBlockSize.
--Data Examples
-- 'monochrome1bit' Example 1
--
--  Example 1 shows a graphic of an arrow as it would be shown on the
--  DMS display.
--  Since the graphic size is 24x7 pixels (which is divisible by 8),
--  the bitwise layout below represents how it appears on the display.
--
--               765432107654321076543210
--
--  BYTE1..3    000000000000110000000000
--  BYTE4..6    000000000000011100000000
--  BYTE7..9    000000000000000111100000
--  BYTE10..12  001111111111111111111000
--  BYTE13..15  000000000000000111100000
--  BYTE16..18  000000000000011100000000
--  BYTE19..21  000000000000110000000000
--
--  The following represents the byte stream of the graphic above. The
--  24 by 7 pixel graphic takes 21 bytes to define.
--  00 0C 00 00 07 00 00 01 E0 3F FF F8 00 01 E0 00 07 00 00 0C 00
--
--
-- 'monochrome1bit' Example 2
--   The following pattern is what would be displayed. The graphic is
--   10 pixels wide by 6 pixels high. The general appearance of the
--   sample graphic is an X followed by a 1.
--
--  1000010010
--  0100100110
--  0011000010
--  0011000010
--  0100100010
--  1000010111
--
--  Byte stream of example 2
--
--   Example 2 graphic is 8 bytes in length. Only 60 of the 64 bits
--   make up the graphic. The last 4 bits are buffer
--             7654321076543210
--
```

```
--  BYTE1..2  1000010010010010
--  BYTE3..4  0110001100001000
--  BYTE5..6  1100001001001000
--  BYTE7..8  1010000101110000
--
--  The following represents the byte stream of the graphic above. The
--  10 by 6 pixel graphic takes 8 bytes to define.
--  84 92 63 08 C2 48 A1 70
--
--
--  'color24bit' Example
--
--   Using the same graphic of an multi-colored arrow as the first example,
--   below is how it would
--   appear on the display. A legend is listed below for color reference.
--
--  0-black, R-red, W-white, B-blue, G-green, P-purple
--
--  0000RW00000000
--  00000RRW000000
--  0000000RRRW000
--  GGGGGGGPPPPPW0
--  0000000PPPB000
--  00000PPB000000
--  0000PB00000000
--
--  The following represents the byte stream of the graphic above. The
--  14 by 7 pixel graphic takes 294 bytes to define. The bytes below
--  are and red, green, blue representation of the pixels to be displayed.
--  In hexadecimal each grouping below represents a pixel.
--    BBGGRR where BB represents a byte of blue in hexadecimal
--       and GG represents a byte of green in hexadecimal
--       and RR represents a byte of red in hexadecimal
--
-- 000000 000000 000000 000000 0000FF FFFFFF 000000 000000 000000 000000
-- 000000 000000 000000 000000 000000 000000 000000 000000 000000 0000FF
-- 0000FF FFFFFF 000000 000000 000000 000000 000000 000000 000000 000000
-- 000000 000000 000000 000000 000000 0000FF 0000FF 0000FF FFFFFF 000000
-- 000000 000000 00FF00 00FF00 00FF00 00FF00 00FF00 00FF00 00FF00 FF33CC
-- FF33CC FF33CC FF33CC FF33CC FFFFFF 000000 000000 000000 000000 000000
-- 000000 000000 000000 FF33CC FF33CC FF33CC FF0000 000000 000000 000000
-- 000000 000000 000000 000000 000000 FF33CC FF33CC FF0000 000000 000000
-- 000000 000000 000000 000000 000000 000000 000000 000000 FF33CC FF0000
-- 000000 000000 000000 000000 000000 000000 000000 000000

<Table Type> static
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.7"
::= {graphicDefinition 7}

dmsGraphicBitmapEntry OBJECT-TYPE
SYNTAX      DmsGraphicBitmapEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "<Definition> Parameters of the Graphic Bitmap Table."
INDEX {dmsGraphicBitmapIndex, dmsGraphicBlockNumber}
::= {dmsGraphicBitmapTable 1}
```

```
DmsGraphicBitmapEntry::= SEQUENCE {
      dmsGraphicBitmapIndex   Integer32,
      dmsGraphicBlockNumber   Integer32,
      dmsGraphicBlockBitmap   OCTET STRING}
```

5.12.7.1 Graphic Index Parameter

```
dmsGraphicBitmapIndex  OBJECT-TYPE
SYNTAX      Integer32 (1..255)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the row number of the entry. This index directly
corresponds to dmsGraphicIndex, the index of dmsGraphicTable.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.7.1"
::= { dmsGraphicBitmapEntry 1 }
```

5.12.7.2 Graphic Block Number Parameter

```
dmsGraphicBlockNumber  OBJECT-TYPE
SYNTAX      Integer32 (1..255)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the offset of the corresponding
dmsGraphicBlockBitmap's data within the graphic image, in
dmsGraphicBlockSize-sized chunks.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.7.2"
::= { dmsGraphicBitmapEntry 2 }
```

5.12.7.3 Graphic Block Bitmap Parameter

```
dmsGraphicBlockBitmap  OBJECT-TYPE
SYNTAX      OCTET STRING
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
"<Definition> The contents of the given block of the bitmap of the graphic
image. Each dmsGraphicBlockBitmap value is a sequence of dmsGraphicBlockSize
octets. If a SET request for dmsGraphicBlockBitmap contains less than
dmsGraphicBlockSize octets, then the supplied data shall be loaded into the
beginning of the block, and the remainder of the block shall be filled with
octets with the value 0 (zero). If a SET request for dmsGraphicBlockBitmap
contains more than dmsGraphicBlockSize octets, the device shall return a SNMP
wrongValue error. If a GET request is received for a dmsGraphicBlockBitmap
entry for which no SET request has been accepted, then the controller shall
respond with a successful GET reply, and the value returned to the central
system shall be an OCTET STRING of dmsGraphicBlockSize octets, all of which
have the value 0 (zero).
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.7.3"
::= { dmsGraphicBitmapEntry 3 }
```

5.12.8 Available Graphic Memory Version 4 Parameter

```
dmsAvailableGraphicMemoryV4  OBJECT-TYPE
SYNTAX      Unsigned32
UNITS       "octets"
MAX-ACCESS  read-only
```

```
STATUS      current
DESCRIPTION
"<Definition> An indication of the amount of memory left, in octets (i.e.,
eight-bit bytes), to store graphics.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.10.4"
::= { graphicDefinition 8 }
```

5.13 Callable Commands

```
dmsCallable  OBJECT-IDENTITY
STATUS  current
DESCRIPTION
"<Definition> This node is an identifier used to group all objects for
commands that can be called in the DMS.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.11"
::= { dms 11 }
```

5.13.1 Number of Callable Messages

```
dmsNumCallableMessages  OBJECT-TYPE
SYNTAX      Integer32 (0..255)
UNITS       "entries"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
"<Definition> Indicates the number of rows currently stored in the
dmsCallableMessagesTable. The messages are not required to be sequential.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.11.1"
::= { dmsCallable 1 }
```

5.13.2 Callable Command Table

```
dmsCallableMessageTable  OBJECT-TYPE
SYNTAX      SEQUENCE OF DmsCallableMessageEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "<Definition> A table containing a list of message activation
codes that can be called by other routines, such as the fdActionPointer (as
defined in ISO 26048-1), which can be called by the fdCondTriggerTable.
Implementations that support this table and the fdActionTable shall allow any
instance of fdActionPointer to reference any row of this table.
<Informative> Whereas the dmsActionTable uses the MessageIDCode to define the
contents of the schedule row of the message table, this table uses the
MessageActivateCode to immediately activate a message on the sign with a
defined duration and priority (i.e., if successful the MultiString is stored
in dmsMessageMultiString.5.1)
<Table Type> dynamic
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.11.2"
::= { dmsCallable 2 }

dmsCallableMessageEntry  OBJECT-TYPE
SYNTAX      DmsCallableMessageEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION "<Definition> Parameters of the dmsCallableMessageTable.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.11.2.1"
INDEX {dmsCallableMessageIndex}
::= { dmsCallableMessageTable 1}
```

```
DmsCallableMessageEntry ::= SEQUENCE {
   dmsCallableMessageIndex        Integer32,
   dmsCallableMessageCode         MessageActivationCodeV4,
   dmsCallableMessageStorageType  StorageType,
   dmsCallableMessageRowStatus    RowStatus }
```

5.13.2.1 Callable Message Index Parameter

```
dmsCallableMessageIndex  OBJECT-TYPE
SYNTAX      Integer32 (1..255)
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"<Definition> Index of the row entry within the CallableMessageTable.
<Parameter Type> Status
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.11.2.1.1"
::= { dmsCallableMessageEntry 1 }
```

5.13.2.2 Callable Message Activation Code Parameter

```
dmsCallableMessageCode  OBJECT-TYPE
SYNTAX      MessageActivationCodeV4
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
"<Definition> The message activation code to be called when this row is
called by an outside source, such as the fdActionPointer.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.11.2.1.2"
::= { dmsCallableMessageEntry 2 }
```

5.13.2.3 Callable Message Storage Type Parameter

```
dmsCallableMessageStorageType  OBJECT-TYPE
SYNTAX      StorageType
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
"<Definition> This object defines whether this callable message is kept in
volatile storage and lost upon reboot or if this row survives a reboot by
being stored in non-volatile or permanent storage. Rows having the value
'permanent' shall allow write access to its associated
fdCallableCommandRowStatus to disable the callable command.
<Parameter Type> Configuration
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.11.2.1.3"
::= { dmsCallableMessageEntry 3 }
```

5.13.2.4 Callable Message Row Status Parameter

```
dmsCallableMessageRowStatus  OBJECT-TYPE
SYNTAX      RowStatus
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
"<Definition> The object that indicates the status of this callable message
and allows entries to be added and removed from this table. All other read-
create objects in this entry can only be modified when the value of this
object is not 'active'. Implementations shall support both createAndGo and
createAndWait.
<Parameter Type> Configuration
```

```
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.11.2.1.4"
::= { dmsCallableMessageEntry 4 }
```

5.13.3 Sign Display Shutdown

```
dmsCallableSignDisplayShutdown OBJECT-IDENTITY
STATUS current
DESCRIPTION
"<Definition> This object can be called by the fdActionTable to force the
sign display to shutdown. This can be useful to prevent damage to components
(e.g., due to overheating).
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.11.3"
::= { dmsCallable 3 }
```

5.13.4 Sign Display Restart

```
dmsCallableSignDisplayRestart OBJECT-IDENTITY
STATUS current
DESCRIPTION
"<Definition> This object can be called by the fdActionTable to force the
sign display to restart after a shutdown.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.11.4"
::= { dmsCallable 4 }
```

5.14 Object Groups

```
dmsConformance OBJECT-IDENTITY
STATUS current
DESCRIPTION
"<Definition> This node is an identifier used to manage the conformance for
DMS.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127"
::= { dms 127 }

dmsGroups OBJECT-IDENTITY
STATUS     current
DESCRIPTION
"<Definition> A node for group definitions related to the DMS MIB.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2"
::= { dmsConformance 2 }
```

5.14.1 DMS Core Group

```
dmsCoreGroupV1 OBJECT-GROUP
OBJECTS { dmsSignType,
          dmsSignTechnology,
          dmsMessageMultiString,
          dmsMessageOwner,
          dmsMessageBeacon,
          dmsMessagePixelService,
          dmsMessageRunTimePriority,
          dmsMessageStatus,
          dmsControlMode,
          dmsActivateMsgError,
          dmsMultiSyntaxError,
          dmsMultiSyntaxErrorPosition,
          dmsMultiOtherErrorDescription,
          dmsActivateMessageV4,
          dmsActivateErrorMsgCodeV4,
          dmsMessageTimeRemaining,
          shortErrorStatus,
          dmsOtherShortErrorDescr }
STATUS  current
DESCRIPTION
"<Definition> The core set of DMS objects required for all implementations.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.1"
::= { dmsGroups 1 }
```

5.14.2 VMS Core Group

```
vmsCoreGroupV1 OBJECT-GROUP
OBJECTS { dmsBeaconType,
          monochromeColor,
          defaultFlashOn,
          defaultFlashOnActivate,
          defaultFlashOff,
          defaultFlashOffActivate,
          defaultFont,
          defaultFontActivate,
          defaultJustificationLine,
          defaultJustificationLineActivate,
          defaultJustificationPage,
          defaultJustificationPageActivate,
          defaultPageOnTime,
          defaultPageOnTimeActivate,
          defaultPageOffTime,
          defaultPageOffTimeActivate,
          defaultBackgroundRGB,
          defaultBackgroundRGBActivate,
          defaultForegroundRGB,
          defaultForegroundRGBActivate,
          defaultCharacterSet,
          dmsColorScheme,
          dmsSupportedMultiTags,
          dmsMaxNumberPages,
          dmsMaxMultiStringLength,
          dmsNumPermanentMsg,
          dmsNumChangeableMsg,
          dmsMaxChangeableMsg,
          dmsFreeChangeableMemory,
          dmsNumVolatileMsg,
          dmsMaxVolatileMsg,
          dmsFreeVolatileMemory,
          dmsMessageCRC,
          dmsValidateMessageError,
          dmsMemoryMgmt }
STATUS  current
DESCRIPTION
"<Definition> The core set of VMS objects required for all variable message
sign implementations.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.2"
::= { dmsGroups 2 }
```

5.14.3 Matrix Group

```
dmsMatrixGroupV1 OBJECT-GROUP
OBJECTS { vmsCharacterHeightPixels,
          vmsCharacterWidthPixels,
          vmsSignHeightPixels,
          vmsSignWidthPixels,
          vmsHorizontalPitch,
          vmsVerticalPitch,
          pixelFailureTableNumRows,
          pixelFailureXLocation,
          pixelFailureYLocation,
          pixelFailureStatus,
          pixelTestActivation,
          dmsPixelStatusV4,
          dmsPixelFailureTestRows,
          dmsPixelFailureMessageRows }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required for all matrix signs.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.3"
::= { dmsGroups 3 }
```

5.14.4 Beacon Group

```
dmsBeaconGroupV1 OBJECT-GROUP
OBJECTS { dmsBeaconCount,
          dmsBeaconXLocation,
          dmsBeaconYLocation,
          dmsBeaconWidth,
          dmsBeaconHeight,
          dmsBeaconColorScheme,
          dmsBeaconFlashResolution,
          dmsBeaconPlanDescr,
          dmsBeaconPlanCycle,
          dmsBeaconPlanStorageType,
          dmsBeaconPlanRowStatus,
          dmsBeaconPlanDetailCycle,
          dmsBeaconPlanDetailOffset,
          dmsBeaconPlanDetailFlash,
          dmsBeaconPlanDetailColor }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required to manage the DMS beacons.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.4"
::= { dmsGroups 4 }
```

5.14.5 Display Capabilities Group

```
dmsDisplayCapabilitiesGroupV1 OBJECT-GROUP
OBJECTS { dmsSignAccess,
          dmsSignHeight,
          dmsSignWidth,
          dmsHorizontalBorder,
          dmsVerticalBorder,
          dmsLegend }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required to determine the display
capabilities.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.5"
::= { dmsGroups 5 }
```

5.14.6 Display Location Group

```
dmsDisplayLocationGroupV1 OBJECT-GROUP
OBJECTS { dmsDisplayLatitude,
          dmsDisplayLongitude,
          dmsDisplayDirection,
          dmsDisplayOffsetLatitude,
          dmsDisplayOffsetLongitude }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required to identify the location of the
display.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.6"
::= { dmsGroups 6 }
```

5.14.7 Current Message Group

```
dmsCurrentMessageGroupV1 OBJECT-GROUP
OBJECTS { dmsMsgTableSource,
          dmsMsgSourceMode,
          dmsMsgRequesterAddressType,
          dmsMsgRequesterAddress,
          dmsMsgRequesterSecurityName }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required to monitor the current message.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.7"
::= { dmsGroups 7 }
```

5.14.8 DMS Font Group

```
dmsFontGroupV1 OBJECT-GROUP
OBJECTS { numFonts,
          fontNumber,
          fontName,
          fontHeight,
          fontCharSpacing,
          fontLineSpacing,
          fontVersionID,
          fontStatus,
          maxFontCharacters,
          characterWidth,
          characterBitmap,
          fontMaxCharacterSize }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required to support fonts.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.8"
::= { dmsGroups 8 }
```

5.14.9 DMS Graphics Group

```
dmsGraphicsGroupV1 OBJECT-GROUP
OBJECTS { dmsGraphicMaxEntries,
          dmsGraphicNumEntries,
          dmsGraphicMaxSize,
          dmsGraphicBlockSize,
          dmsGraphicNumber,
          dmsGraphicName,
          dmsGraphicHeight,
          dmsGraphicWidth,
          dmsGraphicType,
          dmsGraphicID,
          dmsGraphicTransparentEnabled,
          dmsGraphicTransparentColor,
          dmsGraphicStatus,
          dmsGraphicBlockNumber,
          dmsGraphicBlockBitmap,
          dmsAvailableGraphicMemoryV4 }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required to support graphics.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.9"
::= { dmsGroups 9 }
```

5.14.10 Drum Group

```
dmsDrumGroupV1 OBJECT-GROUP
OBJECTS { dmsActivateMessageState }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required to manage drum signs.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.10"
::= { dmsGroups 10 }
```

5.14.11 Pixel Service Group

```
dmsPixelServiceGroupV1 OBJECT-GROUP
OBJECTS { vmsPixelServiceFrequency,
          vmsPixelServiceTime,
          vmsPixelServiceDurationV4 }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required to manage pixel service.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.11"
::= { dmsGroups 11 }
```

5.14.12 Illumination Group

```
dmsIlluminationGroupV1 OBJECT-GROUP
OBJECTS { dmsIllumControl,
          dmsIllumNumBrightLevels,
          dmsIllumBrightLevelStatus,
          dmsIllumManLevel,
          dmsIllumLightOutputStatus }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required to manually control
illumination.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.12"
::= { dmsGroups 12 }
```

5.14.13 Automatic Brightness Group

```
dmsAutoBrightGroupV1 OBJECT-GROUP
OBJECTS { dmsIllumMaxPhotocellLevel,
          dmsIllumPhotocellLevelStatus,
          dmsIllumBrightnessValues,
          dmsIllumBrightnessValuesError }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required to manage automatic brightness
of the sign display.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.13"
::= { dmsGroups 13 }
```

5.14.14 DMS Schedule Group

```
dmsScheduleGroupV1 OBJECT-GROUP
OBJECTS { numActionTableEntries,
          dmsActionMsgCode }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required to implement scheduled messages
(i.e., row 6 of the message table).
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.14"
::= { dmsGroups 14 }
```

5.14.15 Callable Message Group

```
dmsCallableMessageGroupV1 OBJECT-GROUP
OBJECTS { dmsNumCallableMessages,
          dmsCallableMessageCode,
          dmsCallableMessageStorageType,
          dmsCallableMessageRowStatus }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required to implement messages based on
internal configurable logic (e.g., ISO 26048-1 conditional triggers).
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.15"
::= { dmsGroups 15 }
```

5.14.16 SRSA Pixel Group

```
dmsSrsaPixelGroupV1 OBJECT-GROUP
OBJECTS { dmsSrsaPixelsTop,
          dmsSrsaPixelsLeft,
          dmsSrsaPixelsBottom,
          dmsSrsaPixelsRight }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required to determine the pixels
associated with an SRSA entry (see ISO 26048-1).
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.16"
::= { dmsGroups 16 }
```

5.14.17 MULTI Fields Group

```
dmsMultiFieldsGroupV1 OBJECT-GROUP
OBJECTS { statMultiFieldRows,
          statMultiFieldCode,
          statMultiCurrentFieldValue }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required to monitor MULTI field values.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.17"
::= { dmsGroups 17 }
```

5.14.18 Speed Limit Group

```
dmsSpeedLimitGroupV1 OBJECT-GROUP
OBJECTS { dmsCurrentSpeedLimit }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required to manage the speed limit to be
associated with the sign.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.18"
```

```
::= { dmsGroups 18 }
```

5.14.19 Short Power Recovery Group

```
dmsShortPowerRecoveryGroupV1 OBJECT-GROUP
OBJECTS { dmsShortPowerRecoveryMessage }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required to manage the message to be
displayed after a short power recovery.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.19"
::= { dmsGroups 19 }
```

5.14.20 Long Power Recovery Group

```
dmsLongPowerRecoveryGroupV1 OBJECT-GROUP
OBJECTS { dmsShortPowerLossTime,
          dmsLongPowerRecoveryMessage }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required to manage the message to be
displayed after a long power recovery.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.20"
::= { dmsGroups 20 }
```

5.14.21 Power Loss Group

```
dmsPowerLossGroupV1 OBJECT-GROUP
OBJECTS { dmsPowerLossMessage }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required to manage the message to be
displayed during a power loss.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.21"
::= { dmsGroups 21 }
```

5.14.22 Reset Message Group

```
dmsResetMessageGroupV1 OBJECT-GROUP
OBJECTS { dmsResetMessage }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required to manage the message to be
displayed after a reset.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.22"
::= { dmsGroups 22 }
```

5.14.23 Communicaions Loss Group

```
dmsCommLossGroupV1 OBJECT-GROUP
OBJECTS { dmsCommunicationsLossMessage,
          dmsTimeCommLoss }
STATUS  current
DESCRIPTION
"<Definition> The set of DMS objects required to manage the message to be
displayed during a prolonged period without receiving any communications.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.23"
::= { dmsGroups 23 }
```

5.14.24 End Duration Group

```
dmsEndDurationGroupV1 OBJECT-GROUP
OBJECTS { dmsEndDurationMessage }
STATUS  current
```

```
DESCRIPTION
"<Definition> The set of DMS objects required to manage the message to be
displayed at the end of the previously displayed message's duration.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.127.2.24"
::= { dmsGroups 24 }

END
```

Section 6
Markup Language for Transportation Information (MULTI) [Normative]

6.1 Scope

The scope of this section includes the identification and description of the “Markup Language for Transportation Information” (MULTI). MULTI is a language (not an object) used to convey a Message (Text and Message Attributes) between two entities. An object that makes use of MULTI has a syntax of octet string. The octet string shall conform to the MULTI language.

6.2 MULTI - Setup and Definition

6.2.1 Definition

The Markup Language for Transportation Information (MULTI) is similar to HTML where text is transmitted, and tags define how the text appears (is displayed). Tags are enclosed within delimiters, contain an ID (one or more characters), and any optional parameters necessary for the tag.

MULTI uses the encoding defined by the value of defaultCharacterSet, which can be set to either eightBit or utf8Style encoding. The null character (0x00) is not allowed within MULTI strings for either encoding.

'eightBit' encoding allows for the direct encoding of character numbers 1-255; character codes above 255 must use the hexadecimal MULTI tag. 'utf8Style' encoding allows for the direct encoding of character numbers 1-65535, which supports all characters that can be defined in the NTCIP character table. Within utf8Style encoding, characters numbers below 128 are encoded in a single octet (i.e., resulting in an identical encoding as eightBit for these values) and character numbers above 127 are encoded in multiple octets.

All characters used in MULTI tags are translated into character numbers according to their standardized ASCII code points, which are identical to their code points in Unicode and always have values less than 128. As a result, all MULTI tag characters are always encoded into MULTI strings as 8-bit characters with the most significant bit set to 0, regardless of which encoding scheme is used. Within this section, the MULTI tags are sometimes described as being encoded with "ASCII characters"; it should be understood that this is equivalent to a Unicode character and both encoding schemes represent the character with identical one-octet encodings.

Each MULTI tag begins with a left bracket ([), and ends with a right bracket (]). The tag ID appears after the left bracket ([), and is one or more case-insensitive letters. If the tag has any parameters, they immediately follow the tag ID and are case-insensitive (except where specified). No space or other separating character shall appear between the tag ID and the parameters.

Some tags may operate in pairs, in which case the standard tag notation is defined as the opening tag. The opening tag defines where the tag's functionality begins. The closing tag defines where the functionality of the tag ends, and is defined as an opening tag with a forward slash preceding the tag ID e.g., the opening flash tag is “[fl],” and the closing flash tag is “[/fl].”

A tag does not need to have all or any of its parameters specified. When this occurs, the Sign Controller uses stored default values to determine the complete attributes of the Message. The default parameter values are determined when the message is activated, not when it is stored. Thus, a message could change if the default attributes it uses are changed between the time the message is stored and the time it is activated. Changes to the default MULTI attributes do not affect the currently displayed message. However, the currently displayed message could be reactivated to reflect the changes.

The left bracket ([ ) and right bracket ( ]) are restricted for tag delimiters. To display either of these symbols, two brackets, e.g., "[[" or "]]," must appear together and is interpreted as a single bracket that shall be displayed. If a single right bracket is encountered, the device shall return a syntax error with a value of "unsupportedTag". If a single left bracket without a valid MULTI tag attribute is encountered, the device shall return a syntax error with a value of "unsupportedTag".

MULTI allows tags within the text field of another tag (e.g. flashing within moving text), however there are limitations as to the number of tags, use of tags, and complexity of a Message due to the Display Technology of the Sign and the sign manufacturer.

6.3 Rules to apply attribute tags

a) When a closing tag is defined, a closing tag shall turn off that attribute.
b) A closing tag shall not be required to switch from one non-default state to a second non-default state of the same attribute. In other words, nesting of the same attribute tag shall not be allowed.
c) An opening tag shall apply until the end of the message or until it is changed, meaning that the attribute traverses new line breaks and new page breaks.
d) A message implicitly begins with all attribute tags set to their default states.
e) Any tag transmitted without the parameter value shall use the default parameter value for that tag.
f) Activation of stored messages shall use the values of the MULTI default objects at the time the message is activated and not at the time the message is stored.
   1) Any tag may be placed between the opening and the closing tag of any attribute.

6.4 Defined Tags

Tags are used to describe how the Message shall appear (be displayed). Table 6summarizes the tags defined in MULTI. The conformance column indicates the associated supplemental requirement from which the tag is derived; if the requirement has been selected in the PRL, the associated tag shall be supported.

Table 6 MULTI Tags

Attribute Tag (opening)	Closing Tag (if existing)	Description	Conformance
cbx		Color–background **(deprecated; use pb or cr instead)** The background color for a message	3.6.6.2.5
pbz or pbr,g,b		Color–page background The page background color for a message	3.6.6.2.5
cfx or cfr,g,b		Color–foreground The foreground color for a message	3.6.6.2.5
crx,y,w,h,r,g,b or crx,y,w,h,z		Color Rectangle Color for a rectangular area of the current page of a message	3.6.6.2.5
fx,y		Field The information to embed within a message that is based on data from some device, e.g., clock calendar, temperature sensor, detector, etc.	3.6.6.2.13 (see 6.4.3 for additional details)
fltxoy or floytx	/fl	Flash Activate flashing of the text, define the flash on and off times, and the order of flashing (on/off or off/on)	3.6.6.2.10 3.6.6.2.11

Attribute Tag (opening)	Closing Tag (if existing)	Description	Conformance
fox or fox,cccc		Font Select a font number (as specified within the font table) for the message display. Optional cccc indicates the fontVersionID.	3.6.6.2.6
gn or gn,x,y or gn,x,y,cccc		Graphic Select a graphic image to insert into the message. A graphic image is treated as a single displayable character. It may require a few pixels, or the whole sign to display it. The optional cccc indicates the graphicID for the image.	3.6.6.2.14
hcx		Hexadecimal Character The hexadecimal value of the character to display. Value of a character for display	3.6.6.2.12
jlx		Justification–Line Specify line justification: left, center, right, or full	3.6.6.2.4
jpx		Justification–Page Specify page justification: top, middle, or bottom	3.6.6.2.2
msx,y	/msx,y	Manufacturer Specific Tag(s) Specifies a manufacturer specific tag	
mvtdw,s,r, text		Moving Text Specify the parameters of a horizontal moving (scrolling) text	3.6.6.2.7
nlx		New Line Specify the start of a new line	3.6.6.2.3
np		New Page Specify the start of a new page	3.6.6.2.1
ptxoy		Page Time Specify the page times (t = on , o = off)	3.6.6.2.9
scx	/sc	Spacing Character Specify the spacing between characters	3.6.6.2.8
trx,y,w,h		Text Rectangle Specify the placement of a text window on the display	3.6.6.2.15

6.4.1 Color Background

Note: The function of this tag is deprecated and replaced by the Page Background Color and ColorRectangle MULTI tags. This object is defined for backwards compatibility.

Tag format: [cbx]
where x is an octet string up to three characters in length. The string shall be a numeric value between 0 and 999 selecting a color.

This tag indicates the background color of the Message. This is the color of the “closed” or “off” pixels. The color for the background color code is defined by the enumerated listing of colors in the DmsClassicColor textual convention. The default background color is specified by the defaultBackgroundRGB object.

NTCIP 1203 v03 does not require the sign to be able to change the background color; however the

Controller must recognize the tag. If the controller can change the background color, but does not support the selected color scheme (as defined in the dmsColorScheme object), then a syntaxMULTI error shall be generated with a dmsMultiSyntaxError value of unsupportedTagValue. If the controller cannot change the background color at all, then a syntaxMULTI error shall be generated with a dmsMultiSyntaxError value of unsupportedTag.

6.4.1.1 EXAMPLES

To display the Message “THIS IS A TEST WITH COLOR CHANGE” where the first two words are displayed in the default background color (black, code 0), the next two words have a background color is green (code 3), and the remaining words use the default color, the MULTI string could read:

“THIS IS [cb3]A TEST [cb]WITH COLOR CHANGE”
“THIS IS [cb3]A TEST [cb0]WITH COLOR CHANGE”
“[cb]THIS IS [cb3]A TEST [cb0]WITH COLOR CHANGE”

6.4.2 Page Background Color

Tag format: [pbz] or [pbr,g,b]

where z is an octet string up to three characters in length. The string shall be a numeric value between 0 and 9 when using the colorClassic color scheme (with meanings as defined by DmsClassicColor), between 0 and 1 when using the monochrome1Bit color scheme, or between 0 and 255 when using the monochrome8Bit color scheme (see 5.5.22 Color Scheme Parameter for definitions).

where r is an octet string up to three characters in length. The string shall be a numeric value between 0 and 255 selecting a shade of the color red as defined within section 5.5.22 Color Scheme Parameter.

where g is an octet string up to three characters in length. The string shall be a numeric value between 0 and 255 selecting a shade of the color green as defined within section 5.5.22 Color Scheme Parameter.

where b is an octet string up to three characters in length. The string shall be a numeric value between 0 and 255 selecting a shade of the color blue as defined within section 5.5.22 Color Scheme Parameter.

This tag indicates the page background color of the message before the addition of any text, graphics, or color rectangles. It also specifies the color for bits with a value of zero in 'monochrome1Bit' graphics. The default page background color is specified by the defaultBackgroundRGB object. See the dmsColorScheme object for the definition of color codes (section 5.5.22).

If there is more than one Page Background Color tag on a page, the last one specified will take precedence. The effect of the Page Background Color tag continues across message pages. All Page Background Color tags within a page must appear before any text, graphics or color rectangle specifications for that page. If a Page Background Color tag is specified after text, graphics or color rectangles appear on a page, the controller shall return a syntaxMULTI/tagConflict error.

NTCIP 1203 v03 does not require the sign to be able to change the page background color; however the Controller must recognize the tag. If the controller can change the page background color, but does not support the selected color scheme (as defined in the dmsColorScheme object), then a syntaxMULTI error shall be generated with a dmsMultiSyntaxError value of unsupportedTagValue. If the controller cannot change the page background color at all, then a syntaxMULTI error shall be generated with a dmsMultiSyntaxError value of unsupportedTag

6.4.2.1 Examples

To display a two-page message with a green (code 3) background on the first page and a black (code 0) background on the second page (and assuming a dmsColorScheme value of 'colorClassic (3)'), the

MULTI string could read:

“[pb3]GREEN BACKGROUND[np][pb0]BLACK BACKGROUND”

If the defaultBackgroundRGB object specified black, the previous example can be write as:

“[pb3]GREEN BACKGROUND[np][pb]BLACK BACKGROUND”

6.4.3 Color Foreground

Tag format: [cfx] (version 1 and 2) or [cfr,g,b] (version 2 only)

where x is an octet string up to three characters in length. The string shall be a numeric value between 0 and 9 when using the colorClassic color scheme (with meanings as defined by DmsClassicColor), between 0 and 1 when using the monochrome1Bit color scheme, or between 0 and 255 when using the monochrome8Bit color scheme (see 5.5.22 Color Scheme Parameter for definitions)..

where r is an octet string up to three characters in length. The string shall be a numeric value between 0 and 255 selecting a shade of the color red as defined within section 5.5.22 Color Scheme Parameter.

where g is an octet string up to three characters in length. The string shall be a numeric value between 0 and 255 selecting a shade of the color green as defined within section 5.5.22 Color Scheme Parameter.

where b is an octet string up to three characters in length. The string shall be a numeric value between 0 and 255 selecting a shade of the color blue as defined within section 5.5.22 Color Scheme Parameter.

This tag indicates the foreground color of font characters or 'monochrome1Bit' graphics in the message. This is the color of the pixels with a '1' bit in the characterBitmap object. The default foreground color is specified by the defaultForegroundRGB object. See the DmsColor textual convention for the definition of color codes (Section 5.1).

NTCIP 1203 v03 does not require the sign to be able to change the foreground color, however the Controller must recognize the tag. If the controller can change the foreground color, but does not support the selected color scheme (as defined in the DmsColorScheme textual convention), then a syntaxMULTI error shall be generated with a dmsMultiSyntaxError value of unsupportedTagValue. If the controller cannot change the foreground color at all, then a syntaxMULTI error shall be generated with a dmsMultiSyntaxError value of unsupportedTag.

6.4.3.1 Examples

To display the Message “THIS IS A TEST WITH COLOR CHANGE” where the first two words are displayed in the in white (code 7), the next two words use the default foreground color (Amber, code 9), and the remaining words use the color green (code 3), the MULTI string could read:

“[cf7]THIS IS [cf]A TEST [cf3]WITH COLOR CHANGE”
“[cf7]THIS IS [cf9]A TEST [cf3]WITH COLOR CHANGE”

6.4.4 Color Rectangle

Tag format: [crx,y,w,h,r,g,b]
or
[crx,y,w,h,z]

where x is the left pixel coordinate (beginning with 1) of the upper left corner of a rectangle to receive the color specified in the "rgb" parameters.

where y is the top pixel coordinate (beginning with 1) of the upper left corner of a rectangle to receive the color specified in the "rgb" parameters.

where w is the width in pixels of a rectangle to receive the color specified in the "rgb" parameters. A value of zero (0) specifies that the width is all the pixels from the specified x coordinate to the right edge of the sign. The value range is zero (0) to the width of the sign (as defined in 5.3.4 Sign Width in Pixels parameter.

where h is the height in pixels of a rectangle to receive the color specified in the "rgb" parameters. A value of zero (0) specifies that the height is all the pixels from the specified y coordinate to the bottom edge of the sign. The value range is zero (0) to the height of the sign (as defined in 5.3.3. Sign Height in Pixels parameter).

where r is an octet string up to three characters in length. The string shall be a numeric value between 0 and 255 selecting a shade of the color red as defined within section 5.5.22 Color Scheme Parameter.

where g is an octet string up to three characters in length. The string shall be a numeric value between 0 and 255 selecting a shade of the color green as defined within section 5.5.22 Color Scheme Parameter.

where b is an octet string up to three characters in length. The string shall be a numeric value between 0 and 255 selecting a shade of the color blue as defined within section 5.5.22 Color Scheme Parameter.

where z is an octet string up to three characters in length. The string shall be a numeric value between 0 and 9 when using the colorClassic color scheme, between 0 and 1 when using the monochrome1Bit color scheme, or between 0 and 255 when using the monochrome8Bit color scheme (see 5.5.22 Color Scheme Parameter for definitions).

This tag indicates a background color for a rectangular area of the current page of a message. There can be multiple instances of this tag on a single page. This tag applies only to the current page of the message, it has no further effect after a new page tag. For rules on the order to display overlapping graphics, text rectangles and color rectangles see "Overlaying Graphics, Text Rectangles and Color Rectangles" in the "Text Rectangle" tag description. If a specified rectangle does not fully fit on the sign or if the specified color is not supported by the sign, the controller shall return a syntaxMULTI/unsupportedTagValue error.

6.4.4.1 Examples

Assume a full-matrix 100 by 27 pixel sign. To display a message with the left half of the sign background being red (code 1) and the right half background being blue (code 5), the MULTI string could read:

"[cr1,1,50,27,1][cr51,1,50,27,5]TWO COLORS SHOWING"

Assume a full-matrix 100 by 27 pixel sign. To create a message with a green (code 3) background for the entire sign along with a white (code 7) rectangle in the middle with black (code 0) text on the white rectangle, the MULTI string could read:

"[cb3][cr10,10,65,11,7][tr10,10,65,11][cf0]EXIT NOW"

6.4.5 Fields

Tag format: [fx,y]

where x is an octet string up to two characters in length, indicating the field ID.

where y is an octet string up to two characters in length, indicating the number of characters used to display the data.

Both strings, x and y, shall be a numeric value between 1 and 99.

An operator or user of a DMS may want to display information based on data received from a device that has a direct interface with the DMS Controller. This is accomplished via the field tag, where the Message being displayed changes based on the data (typically real-time) from the other device. The device could be a clock calendar, a weather station, a speed station, etc.

There are two parameters for the field tag, x and y.

The first parameter, x, is an ID to indicate the type of information. Table 7 shows the information to be displayed for each field ID.

Table 7 Field Descriptions

ID	Default Field Width	Allowable Widths	Fill Character	Justification	Overflow Fill	Example	Description	Section
1	5	5	space	right	n/a	'_9:00'	Local time, 12 hour format (no AM/PM indicator present) as defined by controller-localTime	3.6.6.2.13.1
2	5	5	0	right	n/a	'09:00'	Local time, 24 hour format as defined by controller-localTime	3.6.6.2.13.1
3	3	2, 3	space	right	space	'-10' or '_10'	Ambient (Outside) Temperature, degrees Celsius (no plus sign)	3.6.6.2.13.4
4	3	2, 3	space	right	space	'-10' or '_10'	Ambient (Outside) Temperature, degrees Fahrenheit (no plus sign)	3.6.6.2.13.4
5	3	2, 3	space	right	'-'	' 90'	Speed, km/h, as defined by dmsCurrentSpeed	3.6.6.2.13.5
6	2	2, 3	space	right	'-'	' 55'	Speed, mph, as defined by dmsCurrentSpeed	3.6.6.2.13.5
7	3	3	n/a	n/a	n/a	'MON'	Day of week, as defined by controller-localTime Shall be one of (SUN, MON, TUE, WED, THU, FRI, SAT)	3.6.6.2.13.6
		4-9	manufacturer specific Note: Use of these manufacturer specific codes inhibits interoperability.					
8	2	2	0	right	n/a	'05'	Date of month (number), as defined by controller-localTime	3.6.6.2.13.7
9	2	2	0	right	n/a	'04'	Month of year (number), as defined by controller-localTime	3.6.6.2.13.8
10	2	2	0	right	n/a	'00'	Year, 2 digits, as defined by controller-localTime	3.6.6.2.13.9

ID	Default Field Width	Allowable Widths	Fill Character	Justification	Overflow Fill	Example	Description	Section
11	4	4	0	right	n/a	'2000'	Year, 4 digits, as defined by controller-localTime	3.6.6.2.13.9
12	8	8	space	right	n/a	'_9:00_AM' '11:00_PM'	Local time, 12 hour format (with capital AM/PM indicator present) as defined by controller-localTime	3.6.6.2.13.2
13	8	8	space	right	n/a	'_9:00_am' '11:00_pm'	Local time, 12 hour format (with lower-case am/pm indicator present) as defined by controller-localTime	3.6.6.2.13.3
14 - 49							Reserved for future assignment	
50 - 99							User-definable	3.6.6.2.13.10

The second parameter, y, is optional and, if included, must be in the range of 'Allowable Widths'. This defines the width, or number of characters used to display the data.

If the Default Field Width parameter is supplied, and the data requires fewer characters than specified, the data is right justified and the indicated Fill Character (4th column in Table above) will be used to make up the missing characters (e.g., if the local time, 12 hour format is 8:00 and the Default Field Width for Field Tag ID 1 is '5', then the value of this field value would be expressed as '_8:00'). If the Default Field Width parameter is supplied, and the data requires more characters than specified, the indicated Overflow Fill character (6th column in Table above) is used for all characters (e.g., if the speed is larger than 100 km/h (assume 114 km/h), but the Default Field Width for Field Tag ID is '2', then the field value would be expressed as '--').

If the width parameter is not supplied and there are more than one allowable widths, only the characters actually present in the data will be used. Overflow Fill characters will be used if the data is larger than the variable range, for example –100 degrees for field tag 3 or 4.

Specifying widths other than the default width should only be used when you wish to force the fill character to be used, or when you wish to limit the range of displayed data, e.g. to display detected speeds of only 99 miles or kilometers per hour or less.

There is no default object for the field tag. No fields exist in a Message unless explicitly defined in the MULTI Message.

NTCIP 1203 v03 does not require the sign to be able to implement all field IDs, however the Controller must recognize the tag and take appropriate action by implementing the associated functionality or by generating a dmsMultiSyntaxError with a value of unsupportedTag.

If a character immediately precedes or follows a field tag, the current character spacing shall be inserted between the character and the field.

6.4.5.1 Examples

To display the Message “YOUR SPEED IS aa MPH,” where aa is filled with the real-time speed (limited to a maximum of 99 mph) from a local device, the MULTI string could read:

“YOUR SPEED IS [f6,2] MPH”

To display the Message “TIME IS aa:aa TEMPERATURE IS bbb F,” aa:aa is filled with the current time and bbb is filled with the current temperature (using no fill characters), the MULTI string could read:

“TIME IS [f1] TEMPERATURE IS [f4] F”

6.4.6 Flash Time

Tag format: [fltxoy]
or
[floytx]

where t is a fixed parameter code to indicate following number is the flash on time.

where x follows the parameter code “t” and is an octet string up to two characters in length, and indicating the flash on time in tenths (1/10ths) of a second.

where o is a fixed parameter code to indicate following number is the flash off time.

where y follows the parameter code “o” and is an octet string up to two characters in length, and indicating the flash off time in tenths (1/10ths) of a second.

Both strings, x and y, shall be a numeric value between 0 and 99. If either value is zero (0), flashing is turned off.

This tag controls the flashing of a Message or part of a Message. The default of this tag is its non-existence, meaning that each time a message or parts of it are to be flashed, this tag needs to be indicated.

The flashing order, on then off or off then on, is performed in the order the tx and oy parameters appear. In the absence of the t and o parameters (no “t” and “o” codes within the tag), the default flashing order is always on then off with their respective default values. If the order of sequence is to be changed, then the parameters t and o can appear without any time values, in which case the default times (specified by the defaultFlashOn- and defaultFlashOff objectss) are used. If time parameters are indicated, the associated “t” and/or “o” code must appear.

NTCIP 1203 v03 does not require what, if anything, a sign can or cannot flash: a specific character, word, line, or page. However, the Controller must recognize the tag and take appropriate action by implementing functionality or by generating a dmsMultiSyntaxError with a value of unsupportedTag.

A graphic shall be flashing using the specified parameters (or the default flashing parameters) when MULTI graphic tag occurs within a flashing MULTI tag. Also, see the rules and limitations defined under 'Overlaying Graphics, Text Rectangles and Color Rectangles'.

6.4.6.1 Examples

To display the Message “THIS IS A TEST,” where “IS A” is flashing with an on-time of 1.0 seconds and then an off-time of 0.5 seconds (defaults of on- and off-times are set to 0), the MULTI string could read:

“THIS [flt10o5]IS A [/fl]TEST”
“THIS [flt10o5]IS A [/fl]TEST”

To display the Message “THIS IS A TEST,” where “THIS IS A TEST” is flashing with an on-time of 1.0 seconds and then an off-time of 0.5 seconds (defaults: on-time = 1.0; off-time = 0.5), the MULTI string could read:

“[fl]THIS IS A TEST [/fl]”

“[flt10o05]THIS IS A TEST[/fl]”
“[fl]THIS IS A TEST”
“[flt10o05]THIS IS A TEST”

To display the Message “THIS IS A TEST,” where “THIS” is flashing, on 1.0 seconds, then off 1.0 seconds, “TEST” is flashing, off 1.0 seconds, then on 1.0 seconds, with the default on and off times set to 0, the MULTI string could read:

“[flt10o10]THIS [/fl]IS A [flo10t10]TEST[/fl]”
“[flt10o10]THIS [/fl]IS A [flo10t10]TEST”

6.4.7 Font

Tag format: [fox] or [fox,cccc]

where x is an octet string up to three characters in length, and indicates the fontNumber. “X” shall be a numeric value between 1 and 255.

where cccc is an optional 4-digit hexadecimal number indicating the fontVersionID from the font table. Each 'c' in “cccc” shall be an ASCII character in the range from 0-9 or A-F.

This tag controls the selection of the font used to display a message. The font is selected using the fontNumber, not the fontIndex object from the fontTable. The default font is indicated in the defaultFont-object. When fonts of different heights are displayed on the same line, the bottom-most pixel of each font shall be aligned.

The optional “cccc” is used to compare its value with that of the fontVersionID from the fontTable while dmsMessageStatus is in the validating state, and when the message is activated for display. The “cccc” from the tag and the fontVersionID from the fontTable must match for a successful operation. The fontVersionID from the message table is ignored during these operations when the “cccc” is not included in the tag.

NTCIP 1203 v03 does not require how many fonts are to be supported. If a non-existing font is selected, either the dmsValidateMessageError or dmsActivateMsgError object (depending on whether dmsMessageStatus or dmsActivateMessageV4 is set) must be set to a value of 'syntaxMULTI' and the dmsMultiSyntaxError object must be set to a value of 'fontNotDefined'. If the “cccc” field is included, and its value does not match the value of fontVersionID for that font, then either dmsValidateMessageError or dmsActivateMsgError must be set to a value of 'syntaxMULTI' and the dmsMultiSyntaxError object must be set to a value of 'fontVersionID'. Understanding and acting upon the “cccc” field is required for all devices, but the field does not have to be included in the tag.

6.4.7.1 Examples

To display the Message “THIS IS A TEST,” where “IS A” uses the user font number 2, with the default font set to 1, the MULTI string could read:

“THIS [fo2]IS A [fo]TEST”
“THIS [fo2,E19C]IS A [fo,8AC7] TEST”
“THIS [fo2]IS A [fo1]TEST”
“THIS [fo2,E19C]IS A [fo1,8AC7]TEST”
“[fo1]THIS [fo2]IS A [fo]TEST”
“[fo1,8AC7]THIS [fo2,E19C]IS A [fo,8AC7]TEST”

6.4.8 Graphic

Tag format: [gn] or [gn,x,y] or [gn,x,y,cccc]

where n is an octet string up to three characters in length indicating the dmsGraphicNumber from the graphic table (not the dmsGraphicIndex). “n” shall be a numeric value between 1 and 255.

where x specifies the horizontal displacement in pixels of the graphic image from the left edge of the sign. A value of 1 specifies that the left edge of the graphic image is in the left-most pixel column of the sign. “x” shall be a numeric value ranging from 1 to the width of the sign minus the width of the graphic plus 1.

where y specifies the vertical displacement in pixels of the graphic image from the top edge of the sign. A value of 1 specifies that the top edge of the graphic image is in the top-most pixel row of the sign. “y” shall be a numeric value ranging from 1 to the height of the sign minus the height of the graphic plus 1.

where cccc is an optional 4-digit hexadecimal number indicating the graphicVersionID from the graphic table. Each 'c' in “cccc” shall be an ASCII character in the range from 0-9 or A-F.

If the tag format [gn] is used, it is assume that the graphic will start at location 1.1 (upper left hand corner).

This tag controls the selection of a graphic image to insert into a message. The image is selected from the dmsGraphicTable using the dmsGraphicNumber, not the dmsGraphicIndex object.

The “cccc” is compared to the dmsGraphicID from the dmsGraphicTable while dmsMessageStatus is in the validating state, and when the message is activated for display. The “cccc” from the tag and the dmsGraphicID from the dmsGraphicTable must match for a successful operation. If this match is incorrect, the device shall return a syntaxMULTI / graphicID error. The dmsGraphicID from the dmsGraphicTable is ignored during these operations when the “cccc” is not included in the tag.

NTCIP 1203 v03 does not require how many graphic images are to be supported.

For rules on the order to display overlapping graphics, text rectangles and color rectangles see “Overlaying Graphics, Text Rectangles and Color Rectangles” in the “Text Rectangle” tag description.

If a nonexistent image is selected by defining a message and validating via the dmsMessageStatus object, dmsValidateMessageError must be set to a value of 'syntaxMULTI' and dmsMultiSyntaxError object must be set to a value of 'graphicNotDefined'.

If a nonexistent image is selected by activating a message via the dmsActivateMessageV4 object, dmsActivateMsgError must be set to a value of 'syntaxMULTI' and dmsMultiSyntaxError object must be set to a value of 'graphicNotDefined'.

If the optional “cccc” field is included, and its value does not match the value of dmsGraphicID for that image, then the dmsValidateMessageError or dmsActivateMsgError object (depending whether the message is stored in the message table => dmsValidateMessageError or activated => dmsActivateMsgError) must be set to a value of 'graphicID' and the dmsMultiSyntaxError object must be set to a value of 'graphicID'.

6.4.8.1 Examples

To display a message with a graphic (assume graphic #5) on the left and the word “DETOUR” in the middle (assume default line justification is center for the first example), the MULTI string could read:

“[g5,1,1]DETOUR”
“[jl3][g5,1,1]DETOUR”
“[jl3][g5,1,1,E19C]DETOUR”

If the graphic were to be placed on the right side of the sign (assume a sign width of 105 pixels and a graphic width of 25 pixels), the MULTI string could read:

> “[g5,81,1]DETOUR”
> “DETOUR[g5,81,1,E19C]”

6.4.9 Hexadecimal Character

Tag format: [hcx]

> where x is an octet string up to four characters in length, and indicates the character from the current font using the hexadecimal value of the character code to be displayed. “X” shall be a hexadecimal (0-9, A-F) value between 1 and FFFF.

This tag is intended as a method to select a character from a font that cannot be typed on a keyboard (characters 0 through 31 and 128 through 65535.

If this tag is not supported, but is encountered during a message validation, then dmsMultiSyntaxError shall be set to 'unsupportedTag'. If this tag is supported, but it contains an unrecognized character, then dmsMultiSyntaxError shall be set to 'characterNotDefined'.

6.4.9.1 Examples

To display the Message “THIS IS * A TEST,” where “*“ is the hexadecimal code 8A to have all pixels of the character ON, the MULTI string could read:

> “THIS IS [hc8A] A TEST”

6.4.10 Justification—Line

Tag format: [jlx]

> where x is a single octet character, and indicates the type of line justification. “X” shall be a have value between 1 and 5, inclusive.

This tag allows the selection of line justification for the text or portion of the text selected. The value of x shall define the justification according to Table 8.

Table 8 Line Justification Codes

Justification Code	Line Justification
1	other
2	left
3	center
4	right
5	full

The centering of text shall be positioned to have the extra space AFTER the text, when exact centering is not possible because of an odd number of remaining spaces. For example, to center NEMA on a seven (7) character sign, the result would be “_ NEMA _ _”, one space before the word NEMA and two spaces after the word NEMA.

The default value for this tag is indicated in the defaultJustificationLine- object.

The line justification tag must be used in logical order (from left, center, right), otherwise dmsMultiSyntaxError will be set to “tagConflict”. Overlapping of text results in a “textTooBig” value for dmsMultiSyntaxError. No other justification tag may be used in conjunction with full justification on the same line.

If an unsupported justification code is selected, the Controller must recognize the tag and take appropriate action by generating a dmsMultiSyntaxError with a value of unsupportedTag Value.

6.4.10.1 Examples

To display the Message “THIS IS A TEST”, left justified with the default line justification being center, the MULTI string could read:

“[jl2]THIS IS A TEST”

To display the Message “THIS IS A TEST”, with “THIS IS” left justified and “A TEST” right justified and the default line justification being left, the MULTI string could read:

“THIS IS [jl4]A TEST”
“[jl]THIS IS [jl4]A TEST”
“[jl2]THIS IS [jl4]A TEST”

To display the Message “THIS IS A TEST”, with “THIS IS” left justified and “A TEST” right justified and the default line justification being right, the MULTI string could read:

“[jl2]THIS IS [jl]A TEST”
“[jl2]THIS IS [jl4]A TEST”

To display the Message “THIS IS A TEST”, with center justified and the default line justification being center, the MULTI string could read:

“THIS IS A TEST”
“[jl3]THIS IS A TEST”
“[jl]THIS IS A TEST”

To display the message “LFT _ _ _ _ CNTR _ _ RIGHT” on an 18 character sign (or text rectangle), with left, center, and right justified text, the MULTI string can read:

“[jl2]LFT[jl3]CNTR[jl4]RIGHT]”

Note that the center-justified text is centered in the sign (or text rectangle), not in the remaining space between the left and right justified text.

6.4.11 Justification—Page

Tag format: [jpx]

where x is a single octet character, and indicates the type of line justification. “X” shall be a have value between 1 and 4, inclusive.

This tag allows the selection of page justification for the text or portion of the text selected. The value of x shall define the justification according to the Table 9.

Table 9 Page Justification Codes

Justification Code	Page Justification
1	other
2	top
3	middle
4	bottom

The centering of text shall be positioned to have the extra line BELOW the text, when exact centering is not possible because of odd number of unused lines. For example, to center

NTCIP
BY NEMA

on a five (5) line sign, the result would be

NTCIP
BY NEMA

One line would be above the word NTCIP and two lines would be below the words BY NEMA.

The default value for this tag is indicated in the defaultJustificationPage- object.

For multiple page justification tags, the tags must be used in logical order (from top, middle, bottom), otherwise dmsMultiSyntaxError will be set to “tagConflict”. Overlapping of text results in a “textTooBig” value for dmsMultiSyntaxError.

Any page justification tag on a text line must appear before any plain text, hex character tag, or field tag specifications on that line. If a page justification tag appears on a text line after any plain text, hex character tag, or field tag specifications on that line, the controller shall return a dmsMultiSyntaxError=tagConflict error.

When an unsupported justification code is selected, the controller must generate a dmsMultiSyntaxError with a value of unsupportedTagValue.

6.4.11.1 Examples

To display the Message “THIS IS[nl]A TEST”, top justified with the default page justification being middle, the MULTI string could read:

“[jp2]THIS IS[nl]A TEST”

To display the Message “THIS IS[nl]A TEST”, middle justified with the default page justification being middle, the MULTI string could read:

“[jp3]THIS IS[nl]A TEST”
“THIS IS[nl]A TEST”
“[jp]THIS IS[nl]A TEST”

To display the following message on an 8-line sign,

```
TOP
LINE 2
LINE 3
   -
MIDDLE
   -
   -
   -
BOTTOM
```

the MULTI string could read:

“[jp2]TOP[nl]LINE 2[nl]LINE 3[jp3]MIDDLE[jp4]BOTTOM”

Note that the middle-justified line is placed on the middle line of the sign (or text rectangle), not in the middle of the empty lines between the top and bottom justified lines.

6.4.12 Manufacturer Specific Tag

Tag format: [msx,y]

where x is an ASCII number, and indicates the number assigned by NEMA to a specific manufacturer.

where y is a manufacturer specific string. See the manufacturer's manual for explanations. This string, if present, must be preceded by a comma.

This tag allows manufacturers to implement proprietary or experimental functions.

6.4.13 Moving Text Tag

Tag format: [mvtdw,s,r,text]

where t is a character(s) indicating the type of the moving tag. Two types are available:
c = circular,
lx = linear with “x” optionally indicating the delay in tenths of a second between the end of linear motion and the restarting of the linear motion from the initial state. If x is not present, there shall be no delay.

where dis a character indicating the direction in which the text is moving with the following possibilities:
l = left moving text
r = right moving text

where w is a number indicating the width, in pixels, of the window in which the 'text' is to be moved/scrolled.

where sis a number indicating the number of pixels that the text shall move at the defined rate 'r.'

where r is a number indicating the time, in tenths of a second, between two steps 's.'

where text is the array of characters that is to be moved/scrolled. The text shall be case-sensitive.

This tag allows the moving (or scrolling) of the text indicated within the brackets. The different parameters indicate different functions that can be associated with the moving/scrolling of text.

For left moving/scrolling, the window shall be initialized with the first character of the text aligned with the left edge of the window.

For right moving/scrolling, the window shall be initialized with the last character of the text aligned with the right edge of the window.

Circular moving/scrolling is the continuous display of the indicated text, including all spaces shown within the text. In this case, the text will appear moving across the window as though multiple copies of the text were appended to itself. The character spacing is applied between the apparent multiple copies of text.

Linear moving/scrolling is the intermittent display of the indicated text, including all spaces shown within the text. In this case, the window initialized with beginning of the text appearing in the window, then moving across the window until all characters have been displayed. The process will repeat again after the indicated delay time defined by the value x when the t-parameter is lx.

The text can only be moved over one line. If text is supposed to be moved over more than one line, then this tag needs to be indicated for each line.

If this tag is unsupported, or if the display would appear incorrect for the selected parameters (e.g., using a value of 's' equals one (1) on a character matrix sign), the sign should report an unsupportedTagValue error.

If a character immediately precedes or follows a moving text tag, the current character spacing shall be inserted between the character and the moving text window.

If necessary, the number of pixel columns in the final shift of a linear move (before repeating) will be reduced such that the last column of the moving text will appear in the rightmost column of the window for left moves or in the leftmost column of the window for right moves.

6.4.13.1 Examples

Although the printed examples show the text moving by whole character positions, the text displayed on the sign will shift by the amount in the MULTI tag (in the following examples, 1 pixel at a time).

To display the moving text “THIS IS A TEST” which moves circularly to the right within a window of 50 pixels and a rate of 1 pixel per 3 tenths of a second, the MULTI string could read:

“[mvcr50,1,3,THIS IS A TEST]”

Circular right scrolling:

```
[ IS A TEST]
[S IS A TES]
[IS IS A TE]
[HIS IS A T]
[THIS IS A ]
[TTHIS IS A]
[STTHIS IS ]
[ESTTHIS IS]
[TESTTHIS I]
[ TESTTHIS ]
[A TESTTHIS]
```

and so on..

> Note: The text string does not include any spaces (character 0x20) between the words THIS and TEST; however, the display will show inter-character spacing between TEST and THIS.

To display the moving text “THIS IS A TEST” (no spaces before and after the text) which moves linearly (with a delay of 0.5 seconds) to the left within a window of 50 pixels and a rate of 1 pixel per 3 tenths of a second, the MULTI string could read:

“[mvl5l50,1,3,THIS IS A TEST]”

Linear left scrolling:

```
[THIS IS A ]
[HIS IS A T]
[IS IS A TE]
[S IS A TES]
[ IS A TEST]
<0.5 sec delay occurs here>
[THIS IS A ]
```

[HIS IS A T]

and so on..

The following is an example of what occurs when the text field is smaller than the specified window for a left moving linear effect. Because all of the text in the MULTI string has been displayed, the text does not move.

“[mvl5l50,1,3,TEST]”

Linear left scrolling:

[TEST   ]
<0.5 sec delay occurs here>
[TEST   ]

The following is an example of what occurs when the text field is smaller than the specified window for a right moving linear effect.

“[mvl5r50,1,3,TEST]”

Linear right scrolling:

[TEST]
<0.5 sec delay occurs here>
[TEST]

The following is an example of how to move text where a small string of characters is to be scrolled through a larger window. For readability in this example an asterisk represents the space character.

“[mvl5r50,1,3,********TEST****]”
Linear right scrolling:
[TEST****]
[*TEST***]
[**TEST**]
[***TEST*]
[****TEST]
[*****TES]
[******TE]
[*******T]
[********]
<0.5 sec delay occurs here>
[TEST****]
[*TEST***]

and so on…

6.4.14 New Line
Tag format: [nlx]

where x is an ASCII number, and indicates the spacing, in pixels, between two lines. If “x” is not present, the spacing shall be defined by the result of the font line spacing algorithm.

This tag defines the end of one line of Text and the start of a new line of Text. It can optionally allow the default line spacing to be changed (valid only for this line break). All Text that appears after the [nlx] tag appears on the next line of the sign. There is no closing tag for new line tag.

NTCIP 1203 v03 currently does not allow word wrapping. Each line of Text must be limited to the allowable space for the line. The Controller must recognize the tag and return an error should too many characters appear on a line.

6.4.14.1 Examples

To display the Message “THIS IS A TEST,” with “THIS IS” on the top line and “A TEST” on the next line, the MULTI string could read:

“THIS IS[nl]A TEST”

To display the Message “THIS IS A TEST,” with “THIS IS” on the second line and “A TEST” on the next line, the MULTI string could read:

“[nl]THIS IS[nl]A TEST”
“[nl]THIS IS[nl]A TEST”

To display another example utilizing different line spacing (assuming that the default is 3 pixels, and the selected new line spacing should be 5 pixels), the MULTI string could read:

“[nl]THIS IS[nl5]A TEST”

To use another example with 3 lines and different line spacing between the first and the second (spacing of 5 pixels), and the second and the third line (default spacing of 3 pixels), the MULTI string could read:

“THIS IS[nl5]A TEST[nl]ON THREE LINES”

6.4.15 New Page

Tag format: [np]

This tag defines the end of one Page of Text and the start of a new Page of Text. All text that appears after the [np] tag appears on the next Page of the Message. There is no closing tag for new page tag.

If the number of pages used exceeds the number of pages identified by the dmsMaxNumberPages object the Controller must recognize the tag and set dmsMultiSyntaxError object to a value of tooManyPages(12).

6.4.15.1 Examples

To display the Message “THIS IS A TEST,” with “THIS IS” on the first page and “A TEST” on the next page, the MULTI string could read:

“THIS IS[np]A TEST”

To display the Message “THIS IS A TEST,” with “THIS IS” on the second page and “A TEST” on the next page, the MULTI string could read:

“[np]THIS IS[np]A TEST”
“ [np]THIS IS[np]A TEST”
“   [np]THIS IS[np]A TEST”

6.4.16 Page Time

Tag format: [ptxoy]

where t is a fixed parameter code to indicate following number is the page on time.
where x follows the parameter code “t” and is an octet string up to three characters in length, and indicating the page on time in tenths (1/10ths) of a second. This shall be a numeric

value from 1 to 255. The non-existence of a value indicates that the on-time is the default value.
where o is a fixed parameter code to indicate following number is the page off time.
where y follows the parameter code “o” and is an octet string up to three characters in length, and indicating the page off time in tenths (1/10ths) of a second. This shall be a numeric value from 0 to 255. The non-existence of a value indicates that the on-time is the default value.

This tag controls the amount of time each Page of Text is displayed and turned off before switching to the next Page of Text. The t and/or o parameters can appear without any time values, when the default page times (specified by the defaultPageOnTime and defaultPageOffTime objects) are to be used. If time parameters are indicated, the associated “t” and/or “o” code must appear.

If multiple page on and off times are sent for one page, the value of the last indication shall be used.

NTCIP 1203 v03 does not limit page time values, however, the Controller must recognize the tag and take appropriate action by implementing functionality or by generating a dmsMultiSyntaxError with a value of unsupportedTag.

6.4.16.1 Examples

To display the Message “THIS IS A TEST,” where “THIS IS” is on a page with an on-time of 3.0 seconds and an off-time of 0.5 seconds, “A TEST” is on a second page with an on-time of 2.0 seconds and an off-time of 1.0 seconds, with the default page on-time set to 3.0 seconds and page off-time set to 1.0, the MULTI string could read:

“[pt30o5]THIS IS[np][pt20o10]A TEST”
“[pto5]THIS IS[np][pt20o]A TEST”

To display the Message “THIS IS A TEST,” where “THIS IS” is on a page with an on-time of 3.0 seconds and an off-time of 0.5 seconds, “A TEST” is on a second page with a page on-time of 2.0 seconds and an off-time of 1.0 seconds, with the default page on-time set to 3.0 seconds and page off-time set to 0.5, the MULTI string could read:

“[pto]THIS IS[np][pt20o10]A TEST”

To display the Message “THIS IS A TEST,” where “THIS IS” is on a page with an on-time of 3.0 seconds and an of- time of 0.5 seconds, “A TEST” is on a second page with a page on-time of 2.0 seconds and an off-time of 1.0 seconds, with the default page on-time set to 2.0 seconds and page off-time set to 1.0, the MULTI string could read:

“[pt30o5]THIS IS[np][pto]A TEST”

6.4.17 Spacing – Character

Tag format: [scx]

where x is an octet string up to two characters in length, and indicates the number of pixels between the characters. “x” is a mandatory parameter and shall be a numeric value between 0 and 99.

This tag controls the spacing between any two adjacent characters. The tag will override the character spacing defined by the result of the font character spacing algorithm.

The default spacing for a character is the default spacing of that character's font. A closing tag shall be required to return to the character's font spacing.

The space indicated shall apply to the space between the last character of the previous spacing and the first character of the new spacing.

NTCIP 1203 v03 does not require support of spacing character values. However, the Controller must recognize the tag and take appropriate action by implementing functionality or by generating a dmsMultiSyntaxError with a value of unsupportedTag.

6.4.17.1 Examples

To display the Message “THIS IS A TEST,” where “IS A” uses a different spacing between each character, with an assumed font character spacing of 1 pixel, the MULTI string could read:

“THIS_[sc2]IS_A_[/sc]TEST”

the display would then show:

“T*H*I*S*_**I**S**_**A**_*T*E*S*T”

where an “*” indicates a space of one pixel, and

where a "_" indicates a space character

6.4.18 Text Rectangle

Tag format: [trx,y,w,h]

where x is the left pixel coordinate (beginning with 1) of the upper left corner of a rectangle to receive text (font characters).

where y is the top pixel coordinate (beginning with 1) of the upper left corner of a rectangle to receive text (font characters).

where w is the width in pixels of a rectangle to receive text (font characters). A value of zero (0) specifies that the width is all the pixels from the specified x coordinate to the right edge of the sign. The value range is zero (0) to the width of the sign (as defined in 5.3.4 Sign Width in Pixels parameter).

where h is the height in pixels of a rectangle to receive text (font characters). A value of zero (0) specifies that the height is all the pixels from the specified y coordinate to the bottom edge of the sign. The value range is zero (0) to the height of the sign (as defined Section 5.3.3. Sign Height in Pixels parameter).

When text (comprised of font characters) is drawn on a message page, it is drawn within a predefined rectangle of pixel coordinates. By default, that rectangle is the entire face of the sign. This tag allows specification of a different rectangle. Any line justification or page justification tags justify the text relative to the currently specified rectangle.

If specified text does not fully fit within the text rectangle, the controller shall return a syntaxMULTI/textTooBig error.

If the specified text rectangle does not match character boundaries (of a character matrix signs) or line boundaries (of a line matrix sign), the controller shall return a syntaxMULTI/unsupportedTagValue error.

Multiple text rectangles can be specified within a message page. All text following a text rectangle tag is drawn within that rectangle. Text rectangles only apply to the message page in which they are specified. In other words, when a new page tag is encountered, any current text rectangle specification is discarded and the new page begins with the default rectangle of the entire sign face.

If a specified text rectangle does not fully fit on the sign, the controller shall return a syntaxMULTI/ unsupportedTagValue error.

Overlaying Graphics, Text Rectangles and Color Rectangles

Because graphics, text rectangles and color rectangles can be placed at any specified location on a message page, specific rules must be applied if any of these items are supposed to overlap. The order for adding these items to a message is as follows:

a) The default background color or the color specified by a color background tag is applied to all pixels of the message page (sign face).
b) As graphics and color rectangles are encountered, they are added to the message page. Color rectangles overlay or fill the specified color into the entire specified rectangle without regard to what has been previously drawn there. Graphics also overlay any previously drawn pixels unless a portion of the graphic is transparent (see the dmsGraphicTransparentEnabled and dmsGraphicTransparentColor objects).
c) Text shall not be drawn into the message, until a text rectangle (the whole sign, if no rectangle is defined) is ended by one of the following:
    1) End of a page;
    2) New text rectangle tag;
    3) End of a message.

    This is done by coloring the foreground of the text with the foreground color while leaving the background unchanged.
d) At the end of a page, go back to step a). At the end of a text rectangle, go back to step b).

    Note: Graphics and color rectangles are not placed within the area of a text rectangle—they are placed relative to the entire sign face, not a text rectangle.

No portion of a region of text or graphic that flashes or moves (MULTI flash or moving tag) shall be overlaid by other graphics, text rectangles or color rectangles; otherwise the controller shall return a syntaxMULTI/tagConflict error. This flashing or moving region may be only a part of a text rectangle that contains the flashing or moving text.

6.4.18.1 Examples

Assume a full-matrix 105 by 27 pixel sign. To display a 27 by 27 graphic (assume it is graphic #4) on the left side of the sign and place three lines of centered text in the area to the right of the graphic, the MULTI string could read:

```
“[g4,1,1][tr28,1,78,27]LEFT[nl]EXIT[nl]AHEAD”
“[g4,1,1][tr28,1,0,0][jl3]LEFT[nl]EXIT[nl]AHEAD”
```

To put the same graphic on the right side of the sign with the text area to its left, the MULTI string could read:

```
“[g4,79,1][tr1,1,78,27]LEFT[nl]EXIT[nl]AHEAD”
“[tr1,1,78,27][g4,79,1]LEFT[nl]EXIT[nl]AHEAD”
“[tr1,1,78,27]LEFT[nl]EXIT[nl]AHEAD[g4,79,1]”
```

To create a one-page message with the graphic placed in the middle of the sign with a text area on either side of it, the MULTI string could read:

```
“[g4,40,1][tr1,1,39,27]LEFT[nl]SIDE[tr67,1,39,27]RIGHT[nl]SIDE”
```

To place the graphic on the right side of the sign and place the text “65” (5x7 font) over the top and center of the graphic, the MULTI string could read:

```
“[g4,79,1][tr79,11,27,7]65”
“[g4,79,1][tr79,1,27,27][jl3][jp3]65”
```

Annex A
Requirements Traceability Matrix (RTM) [Normative]

The Requirements Traceability Matrix (RTM) links the Functional Requirements as presented in Section 3 with the corresponding Dialogs (Section 4.2) on the same (gray) line. Each Functional Requirement/Dialog relates/uses one or more groups of Objects. The Objects (also known as Data Elements) are listed to the side; the formal definition of each object is contained within Section 5. Using this table, each Functional Requirement can thus be traced in a standardized way.

> Note: The INDEX objects into any of the tables are not explicitly exchanged but are used as index values for other objects that are exchanged.

The audience for this table is implementers (vendors and central system developers) and conformance testers. Additionally, other interested parties might use this table to determine how particular functions are to be implemented using the standardized dialogs, interfaces, and object definitions.

To conform to a Functional Requirement, a DMS shall implement all Objects and Dialogs traced from that Functional Requirement; a Management Station shall implement all Dialogs traced from the Functional Requirement. to be consistent with a Functional Requirement, a Management Station shall be able to fulfill the Functional Requirement using only Objects and Dialogs that a conforming DMS is required to support.

Section 3 defines Supplemental Requirements, which are refining other functional requirements. These functional requirements in turn are generally traced to design elements (e.g., rather than being directly traced to design elements). The Supplemental RTM in Section A.4 below identifies and traces the implied relationships between supplemental requirements and direct requirements.

Section 6 defines the 'Mark-Up Language for Transportation Information' (MULTI), which defines tags that can be used within the text of a DMS message to define its display on the face of the DMS display. The Multi Field Traceability Matrix in Section A.5 below identifies and traces the implied relationships between MULTI tags and direct requirements.

A.1 Notation [Informative]

A.1.1 Functional Requirement Columns

The functional requirements are defined within Section 3 and the RTM is based upon the requirements within that Section. The section number and the functional requirement name are indicated within these columns.

A.1.2 Dialog Column

The standardized dialogs are defined within Section 4 and the RTM references the traces from requirements to this dialog. The section number of the dialog is indicated within this column.

A.1.3 Object Columns

The objects are defined within Section 5 of NTCIP 1203 v3 and Section 2 of NTCIP 1201. The RTM references the data objects that are referenced by the dialog. The section number and object name are indicated within these columns.

A.1.4 Additional Specifications

The "Additional Specifications" column may (and should) be used to provide additional notes and requirements about the dialog or may be used by an implementer to provide any additional details about the implementation.

A.2 Instructions for Completing the RTM [Informative]

To find the standardized design content for a functional requirement, search for the requirement identification number and functional requirement under the functional requirements columns. Next to the functional requirements column will be a dialog identification number, identifying either a generic dialog (found in Section G.3) or a specified dialog (found in Section 4.2) to be used to fulfill that requirement. To the right of the dialog identification number are the identification number and name of the data objects that are referenced or used by the dialog to fulfill the functional requirement. Object definitions specific to NTCIP 1203 v3 can be found in Section 5. If an object is defined in a different standard, that standard shall be listed first, followed by the section number where the object definition can be found. The "Additional Specifications" column will provide additional notes or details about the design content.

A.3 Requirements Traceability Matrix (RTM) Table

Requirement	Object Reference	Object	Dialogue	Notes
3.5.1.1.1 : Determine Sign Type and Technology			26048-1§9.2.1 : Get elemental data	
	5.2.2	dmsSignType		
	5.2.9	dmsSignTechnology		
3.5.1.2.1.1 : Determine the Size of the Sign Face			26048-1§9.2.1 : Get elemental data	
	5.2.3	dmsSignHeight		
	5.2.4	dmsSignWidth		
3.5.1.2.1.2 : Determine the Size of the Sign Border			26048-1§9.2.1 : Get elemental data	
	5.2.5	dmsHorizontalBorder		
	5.2.6	dmsVerticalBorder		
3.5.1.2.1.3 : Determine Sign Access and Legend			26048-1§9.2.1 : Get elemental data	
	5.2.1	dmsSignAccess		
	5.2.7	dmsLegend		

Requirement	Object Reference	Object	Dialogue	Notes
3.5.1.2.2.1 : Determine Sign Face Size in Pixels			26048-1§9.2.1 : Get elemental data	
	5.3.3	vmsSignHeightPixels		
	5.3.4	vmsSignWidthPixels		
3.5.1.2.2.2 : Determine Character Size in Pixels			26048-1§9.2.1 : Get elemental data	
	5.3.1	vmsCharacterHeightPixels		
	5.3.2	vmsCharacterWidthPixels		
3.5.1.2.2.3 : Determine Pixel Spacing			26048-1§9.2.1 : Get elemental data	
	5.3.5	vmsHorizontalPitch		
	5.3.6	vmsVerticalPitch		
3.5.1.2.3.1 : Determine Maximum Number of Pages			26048-1§9.2.1 : Get elemental data	
	5.5.24	dmsMaxNumberPages		
3.5.1.2.3.2 : Determine Maximum Message Length			26048-1§9.2.1 : Get elemental data	
	5.5.25	dmsMaxMultiStringLength		
3.5.1.2.3.3 : Determine Supported Color Schemes			26048-1§9.2.1 : Get elemental data	
	5.3.7	monochromeColor		
	5.5.22	dmsColorScheme		
3.5.1.2.3.4 : Determine Message Display Capabilities			26048-1§9.2.1 : Get elemental data	
	5.5.23	dmsSupportedMultiTags		
3.5.1.2.4 : Delete All Messages of a Message Type with One Command			26048-1§9.2.2 : Set elemental data	
	5.7.16	dmsMemoryMgmt		
3.5.1.3.1 : Determine Maximum Number of Fonts Supported			26048-1§9.2.1 : Get elemental data	
	5.4.1	numFonts		
3.5.1.3.2 : Determine Maximum Character Size			26048-1§9.2.1 : Get elemental data	

Requirement	Object Reference	Object	Dialogue	Notes
	5.4.5	fontMaxCharacterSize		
3.5.1.3.3 : Determine Maximum Number of Characters per Font			26048-1§9.2.1 : Get elemental data	
	5.4.3	maxFontCharacters		
3.5.1.3.4 : Retrieve a Font Definition			4.2.2.1 : Retrieving a Font Definition	
	5.4.2.2	fontNumber		
	5.4.2.3	fontName		
	5.4.2.4	fontHeight		
	5.4.2.5	fontCharSpacing		
	5.4.2.6	fontLineSpacing		
	5.4.2.7	fontVersionID		
	5.4.2.8	fontStatus		
	5.4.4.2	characterWidth		
	5.4.4.3	characterBitmap		
3.5.1.3.5 : Configure a Font			4.2.2.2 : Configuring a Font	
	5.4.2.2	fontNumber		
	5.4.2.3	fontName		
	5.4.2.4	fontHeight		
	5.4.2.5	fontCharSpacing		
	5.4.2.6	fontLineSpacing		
	5.4.2.8	fontStatus		
	5.4.4.2	characterWidth		
	5.4.4.3	characterBitmap		
3.5.1.3.6 : Delete a Font			4.2.2.3 : Deleting a Font	

Requirement	Object Reference	Object	Dialogue	Notes
	5.4.2.8	fontStatus		
3.5.1.3.7 : Validate a Font			4.2.2.4 : Validating a Font	
	5.4.2.7	fontVersionID		
	5.4.2.8	fontStatus		
3.5.1.4.1 : Determine Number of Graphics			26048-1§9.2.1 : Get elemental data	
	5.12.1	dmsGraphicMaxEntries		
	5.12.2	dmsGraphicNumEntries		
3.5.1.4.2 : Determine Maximum Graphic Size			26048-1§9.2.1 : Get elemental data	
	5.12.3	dmsGraphicMaxSize		
	5.12.5	dmsGraphicBlockSize		
3.5.1.4.3 : Determine Available Graphics Memory			26048-1§9.2.1 : Get elemental data	
	5.12.2	dmsGraphicNumEntries		
	5.12.8	availableGraphicMemoryV2		
3.5.1.4.4 : Retrieve a Graphic Definition			4.2.2.5 : Retrieving a Graphic Definition	
	5.12.6.2	dmsGraphicNumber		
	5.12.6.3	dmsGraphicName		
	5.12.6.4	dmsGraphicHeight		
	5.12.6.5	dmsGraphicWidth		
	5.12.6.6	dmsGraphicType		
	5.12.6.8	dmsGraphicTransparentEnabled		
	5.12.6.9	dmsGraphicTransparentColor		
	5.12.6.10	dmsGraphicStatus		
	5.12.7.2	dmsGraphicBlockNumber		
	5.12.7.3	dmsGraphicBlockBitmap		

Requirement	Object Reference	Object	Dialogue	Notes
3.5.1.4.5 : Store a Graphic Definition			4.2.2.6 : Storing a Graphic Definition	
	5.12.6.2	dmsGraphicNumber		
	5.12.6.3	dmsGraphicName		
	5.12.6.4	dmsGraphicHeight		
	5.12.6.5	dmsGraphicWidth		
	5.12.6.6	dmsGraphicType		
	5.12.6.8	dmsGraphicTransparentEnabled		
	5.12.6.9	dmsGraphicTransparentColor		
	5.12.6.10	dmsGraphicStatus		
	5.12.7.2	dmsGraphicBlockNumber		
	5.12.7.3	dmsGraphicBlockBitmap		
3.5.1.4.6 : Delete a Graphic			4.2.2.7 : Deleting a Graphic	
	5.12.6.10	dmsGraphicStatus		
3.5.1.4.7 : Validate a Graphic			4.2.2.8 : Validating a Graphic	
	5.12.6.7	dmsGraphicID		
	5.12.6.10	dmsGraphicStatus		
3.5.1.5.1 : Determine Maximum Number of Light Sensor Levels			26048-1§9.2.5 : Get tabular data	
	Srsa-MIB§14	fdSrsaPortMaxValue		type = FAL
3.5.1.5.2 : Configure Light Output Algorithm			4.2.2.9 : Configuring Light Output Algorithm	
	5.8.7	dmsIllumBrightnessValues		
	5.8.8	dmsIllumBrightnessValuesError		
3.5.1.5.3 : Determine Current Light Output Algorithm			26048-1§9.2.1 : Get elemental data	
	5.8.7	dmsIllumBrightnessValues		

Requirement	Object Reference	Object	Dialogue	Notes
	5.8.8	dmsIllumBrightnessValuesError		
3.5.1.6 : Configure Current Speed Limit			26048-1§9.2.2 : Set elemental data	
	5.11.1.4	dmsCurrentSpeedLimit		
3.5.1.7 : Determine Current Speed Limit Configuration			26048-1§9.2.1 : Get elemental data	
	5.11.1.4	dmsCurrentSpeedLimit		
3.5.1.8.1 : Determine the Sign Display Location			26048-1§9.2.1 : Get elemental data	
	5.2.10	dmsDisplayLatitude		
	5.2.11	dmsDisplayLongitude		
	5.2.12	dmsDisplayDirection		
3.5.1.8.2 : Configure the Sign Display Offset Location			26048-1§9.2.2 : Set elemental data	
	5.2.12	dmsDisplayDirection		
	5.2.13	dmsDisplayOffsetLatitude		
	5.2.14	dmsDisplayOffsetLongitude		
3.5.1.8.3 : Determine Sign Display Offset Location			26048-1§9.2.1 : Get elemental data	
	5.2.12	dmsDisplayDirection		
	5.2.13	dmsDisplayOffsetLatitude		
	5.2.14	dmsDisplayOffsetLongitude		
3.5.1.9.1 : Determine Number of Beacons			26048-1§9.2.1 : Get elemental data	
	5.2.15.1	dmsBeaconCount		
3.5.1.9.2 : Determine Beacon Properties			26048-1§9.2.5 : Get tabular data	
	5.2.15.2.2	dmsBeaconXLocation		
	5.2.15.2.3	dmsBeaconYLocation		
	5.2.15.2.4	dmsBeaconWidth		
	5.2.15.2.5	dmsBeaconHeight		

Requirement	Object Reference	Object	Dialogue	Notes
	5.2.15.2.6	dmsBeaconColorScheme		
	5.2.15.2.7	dmsBeaconFlashResolution		
3.5.1.9.3 : Configure Beacon Operational Strategies			26048-1§9.2.11 : Configure entry of a dynamic table	
	5.2.15.3.2	dmsBeaconPlanDescr		
	5.2.15.3.3	dmsBeaconPlanCycle		
	5.2.15.3.4	dmsBeaconPlanStorageType		
	5.2.15.3.5	dmsBeaconPlanRowStatus		
3.5.1.9.4 : Retrieve Beacon Operational Strategies			26048-1§9.2.9 : Get data from dynamic table entry	
	5.2.15.3.2	dmsBeaconPlanDescr		
	5.2.15.3.3	dmsBeaconPlanCycle		
	5.2.15.3.4	dmsBeaconPlanStorageType		
	5.2.15.3.5	dmsBeaconPlanRowStatus		
3.5.1.9.5 : Configure Beacon Operation within a Strategy			26048-1§9.2.11 : Configure entry of a dynamic table	
	5.2.15.4.1	dmsBeaconPlanDetailCycle		
	5.2.15.4.2	dmsBeaconPlanDetailOffset		
	5.2.15.4.3	dmsBeaconPlanDetailFlash		
	5.2.15.4.4	dmsBeaconPlanDetailColor		
3.5.1.9.6 : Retrieve Beacon Operation within a Strategy			26048-1§9.2.9 : Get data from dynamic table entry	
	5.2.15.4.1	dmsBeaconPlanDetailCycle		
	5.2.15.4.2	dmsBeaconPlanDetailOffset		
	5.2.15.4.3	dmsBeaconPlanDetailFlash		
	5.2.15.4.4	dmsBeaconPlanDetailColor		
3.5.2.1 : Manage Control Source			26048-1§9.2.2 : Set elemental data	
	5.7.1	dmsControlMode		

Requirement	Object Reference	Object	Dialogue	Notes
3.5.2.2.1 : Activate a Message			4.2.3.1 : Activating a Message	
	5.7.17	dmsActivateMsgError		
	5.7.18	dmsMultiSyntaxError		
	5.7.19	dmsMultiSyntaxErrorPosition		
	5.7.20	dmsMultiOtherErrorDescription		
	5.7.26	dmsActivateMessageV4		
	5.7.27	dmsActivateErrorMsgCodeV4		
	5.11.2.1.1	shortErrorStatus		
	5.11.2.1.3	dmsOtherShortErrorDescr		
3.5.2.2.2.1 : Determine Default Message Display Parameters			26048-1§9.2.1 : Get elemental data	
	5.5.3	defaultFlashOn		
	5.5.4	defaultFlashOnActivate		
	5.5.5	defaultFlashOff		
	5.5.6	defaultFlashOffActivate		
	5.5.7	defaultFont		
	5.5.8	defaultFontActivate		
	5.5.9	defaultJustificationLine		
	5.5.10	defaultJustificationLineActivate		
	5.5.11	defaultJustificationPage		
	5.5.12	defaultJustificationPageActivate		
	5.5.13	defaultPageOnTime		
	5.5.14	defaultPageOnTimeActivate		
	5.5.15	defaultPageOffTime		
	5.5.16	defaultPageOffTimeActivate		

Requirement	Object Reference	Object	Dialogue	Notes
	5.5.17	defaultBackgroundRGB		
	5.5.18	defaultBackgroundRGBActivate		
	5.5.19	defaultForegroundRGB		
	5.5.20	defaultForegroundRGBActivate		
	5.5.21	defaultCharacterSet		
3.5.2.2.2.2 : Configure Default Background and Foreground Color			26048-1§9.2.2 : Set elemental data	
	5.5.17	defaultBackgroundRGB		
	5.5.19	defaultForegroundRGB		
3.5.2.2.2.3 : Configure Default Flash-On and Flash-Off Times			26048-1§9.2.2 : Set elemental data	
	5.5.3	defaultFlashOn		
	5.5.5	defaultFlashOff		
3.5.2.2.2.4 : Configure Default Font			26048-1§9.2.2 : Set elemental data	
	5.5.7	defaultFont		
3.5.2.2.2.5 : Configure Default Line Justification			26048-1§9.2.2 : Set elemental data	
	5.5.9	defaultJustificationLine		
3.5.2.2.2.6 : Configure Default Page Justification			26048-1§9.2.2 : Set elemental data	
	5.5.11	defaultJustificationPage		
3.5.2.2.2.7 : Configure Default Page On-Time and Page Off-Time			26048-1§9.2.2 : Set elemental data	
	5.5.13	defaultPageOnTime		
	5.5.15	defaultPageOffTime		
3.5.2.2.2.8 : Configure Default Character Set			26048-1§9.2.2 : Set elemental data	
	5.5.21	defaultCharacterSet		
3.5.2.2.3.1 : Determine Available Message Types			26048-1§9.2.1 : Get elemental data	
	5.6.1	dmsNumPermanentMsg		

Requirement	Object Reference	Object	Dialogue	Notes
	5.6.3	dmsMaxChangeableMsg		
	5.6.6	dmsMaxVolatileMsg		
3.5.2.2.3.2 : Determine Available Message Space			26048-1§9.2.1 : Get elemental data	
	5.6.2	dmsNumChangeableMsg		
	5.6.4	dmsFreeChangeableMemory		
	5.6.5	dmsNumVolatileMsg		
	5.6.7	dmsFreeVolatileMemory		
3.5.2.2.3.3 : Define a Message			4.2.3.2 : Defining a Message	
	5.6.8.3	dmsMessageMultiString		
	5.6.8.4	dmsMessageOwner		
	5.6.8.6	dmsMessageBeacon		
	5.6.8.7	dmsMessagePixelService		
	5.6.8.8	dmsMessageRunTimePriority		
	5.6.8.9	dmsMessageStatus		
	5.6.9	dmsValidateMessageError		
	5.7.18	dmsMultiSyntaxError		
	5.7.19	dmsMultiSyntaxErrorPosition		
	5.7.20	dmsMultiOtherErrorDescription		
3.5.2.2.3.4 : Verify Message Contents			26048-1§9.2.5 : Get tabular data	
	5.6.8.5	dmsMessageCRC		
3.5.2.2.3.5 : Retrieve Message			4.2.3.3 : Retrieving a Message	
	5.6.8.3	dmsMessageMultiString		
	5.6.8.4	dmsMessageOwner		
	5.6.8.6	dmsMessageBeacon		

Requirement	Object Reference	Object	Dialogue	Notes
	5.6.8.7	dmsMessagePixelService		
	5.6.8.8	dmsMessageRunTimePriority		
	5.6.8.9	dmsMessageStatus		
3.5.2.2.4.1 : Configure a DMS Message Action			26048-1§9.2.6 : Set tabular data	
	5.9.2.2	dmsActionMsgCode		
3.5.2.2.4.2 : Configure a Callable Message			26048-1§9.2.11 : Configure entry of a dynamic table	
	5.13.2.2	dmsCallableMessageCode		
	5.13.2.3	dmsCallableMessageStorageType		
	5.13.2.4	dmsCallableMessageRowStatus		
3.5.2.2.5.1.1 : Configure Message for Short Power Loss Recovery Event			26048-1§9.2.2 : Set elemental data	
	5.7.8	dmsShortPowerRecoveryMessage		
3.5.2.2.5.1.2 : Configure Message for Long Power Loss Recovery Event			26048-1§9.2.2 : Set elemental data	
	5.7.9	dmsLongPowerRecoveryMessage		
	5.7.10	dmsShortPowerLossTime		
3.5.2.2.5.1.3 : Configure Message for Power Loss Event			26048-1§9.2.2 : Set elemental data	
	5.7.14	dmsPowerLossMessage		
3.5.2.2.5.1.4 : Configure Message for Controller Reset Event			26048-1§9.2.2 : Set elemental data	
	5.7.11	dmsResetMessage		
3.5.2.2.5.1.5 : Configure Message for Communications Loss Event			26048-1§9.2.2 : Set elemental data	
	5.7.12	dmsCommunicationsLossMessage		
	5.7.13	dmsTimeCommLoss		

Requirement	Object Reference	Object	Dialogue	Notes
3.5.2.2.5.1.6 : Configure Message for End Message Display Duration Event			26048-1§9.2.2 : Set elemental data	
	5.7.15	dmsEndDurationMessage		
3.5.2.2.6 : Activate a Message with Status			4.2.3.6 : Activating a Message with Status	
	5.7.17	dmsActivateMsgError		
	5.7.18	dmsMultiSyntaxError		
	5.7.19	dmsMultiSyntaxErrorPosition		
	5.7.20	dmsMultiOtherErrorDescription		
	5.7.25	dmsActivateMessageState		
	5.7.26	dmsActivateMessageV4		
	5.7.27	dmsActivateErrorMsgCodeV4		
	5.11.2.1.1	shortErrorStatus		
	5.11.2.1.3	dmsOtherShortErrorDescr		
3.5.2.2.7 : Extend Message Activation			26048-1§9.2.2 : Set elemental data	
	5.7.4	dmsMessageTimeRemaining		
3.5.2.3.1 : Determine Number of Brightness Levels			26048-1§9.2.1 : Get elemental data	
	5.8.4	dmsIllumNumBrightLevels		
3.5.2.3.2 : Determine Current Composite Photocell Value			26048-1§9.2.1 : Get elemental data	
	Srsa-MIB§24.4	fdSrsaTypeValueStatsMfrValue		Where fdSrsaTypeCode equals "FAL"; See 3.6.3.4.
3.5.2.3.3 : Manually Direct-Control Brightness			4.2.3.4 : Manually Controlling Sign Brightness	
	5.8.6	dmsIllumManLevel		
	5.8.1	dmsIllumControl		Set to 'manualDirect'
3.5.2.3.4 : Manually Index-Control Brightness			4.2.3.4 : Manually Controlling Sign Brightness	

Requirement	Object Reference	Object	Dialogue	Notes
	5.8.6	dmsIllumManLevel		
	5.8.1	dmsIllumControl		Set to 'manualIndexed'
3.5.2.3.5 : Switch Brightness Control Modes			26048-1§9.2.2 : Set elemental data	
	5.8.1	dmsIllumControl		
3.5.2.4.1 : Manage the Exercise of Pixels			4.2.3.5 : Manage the Exercise of Pixels	
	5.7.22	vmsPixelServiceFrequency		
	5.7.23	vmsPixelServiceTime		
	5.7.31	vmsPixelServiceDurationV4		
3.5.2.4.2 : Determine Pixel Service Settings			26048-1§9.2.1 : Get elemental data	
	5.7.22	vmsPixelServiceFrequency		
	5.7.23	vmsPixelServiceTime		
	5.7.31	vmsPixelServiceDurationV4		
3.5.3.1.1 : Provide General DMS Error Status Information			26048-1§9.2.1 : Get elemental data	
	5.11.2.1.1	shortErrorStatus		
	5.11.2.1.3	dmsOtherShortErrorDescr		
3.5.3.1.2.1 : Activate Pixel Testing			4.2.4.1 : Activating Pixel Testing	
	5.11.2.4.3	pixelTestActivation		
3.5.3.1.2.2 : Monitor Pixel Error Count			26048-1§9.2.1 : Get elemental data	
	5.11.2.4.1	pixelFailureTableNumRows		
	5.11.2.4.5	dmsPixelFailureTestRows		
	5.11.2.4.6	dmsPixelFailureMessageRows		
3.5.3.1.2.3 : Monitor Pixel Error Map			26048-1§9.2.7 : Get data column	
	5.11.2.4.4.3	dmsPixelStatusV4		
3.5.3.1.2.4 : Monitor Pixel Error Details			4.2.4.2 : Monitoring Pixel Error Details	

Requirement	Object Reference	Object	Dialogue	Notes
	5.11.2.4.2.3	pixelFailureXLocation		
	5.11.2.4.2.4	pixelFailureYLocation		
	5.11.2.4.2.5	pixelFailureStatus		
3.5.3.1.3 : Monitor Lamp Error Details			26048-1§9.2.5 : Get tabular data	
	5.11.5.1.1	dmsSrsaPixelsTop		
	5.11.5.1.2	dmsSrsaPixelsLeft		
	5.11.5.1.3	dmsSrsaPixelsBottom		
	5.11.5.1.4	dmsSrsaPixelsRight		
3.5.3.1.4 : Monitor Message Activation Error Details			4.2.4.3 : Monitoring Message Activation Error Details	
	5.7.17	dmsActivateMsgError		
	5.7.18	dmsMultiSyntaxError		
	5.7.19	dmsMultiSyntaxErrorPosition		
	5.7.20	dmsMultiOtherErrorDescription		
	5.7.27	dmsActivateErrorMsgCodeV4		
3.5.3.1.5 : Monitor the Sign's Control Source			26048-1§9.2.1 : Get elemental data	
	5.7.1	dmsControlMode		
3.5.3.1.6 : Determine Critical Temperature			26048-1§9.2.6 : Set tabular data	
	Srsa-MIB§24.1	fdSrsaTypeValueStatsMin		
	Srsa-MIB§24.2	fdSrsaTypeValueStatsMax		
3.5.3.2.1 : Monitor Information about the Currently Displayed Message			4.2.4.4 : Monitoring the Current Message	
	5.6.8.3	dmsMessageMultiString		
	5.6.8.4	dmsMessageOwner		
	5.6.8.6	dmsMessageBeacon		

Requirement	Object Reference	Object	Dialogue	Notes
	5.6.8.7	dmsMessagePixelService		
	5.6.8.8	dmsMessageRunTimePriority		
	5.7.4	dmsMessageTimeRemaining		
	5.7.5	dmsMsgTableSource		
	5.7.7	dmsMsgSourceMode		
	5.7.28	dmsMsgRequesterAddressType		
	5.7.29	dmsMsgRequesterAddress		
	5.7.30	dmsMsgRequesterSecurityName		
	5.8.5	dmsIllumBrightLevelStatus		
	5.8.9	dmsIllumLightOutputStatus		
3.5.3.2.2 : Monitor Dynamic Field Values			4.2.4.5 : Monitoring Dynamic Field Values	
	5.11.1.1	statMultiFieldRows		
	5.11.1.2.2	statMultiFieldCode		
	5.11.1.2.3	statMultiCurrentFieldValue		
3.5.3.3.1 : Monitor a Callable Message			26048-1§9.2.9 : Get data from dynamic table entry	
	5.13.2.2	dmsCallableMessageCode		
	5.13.2.3	dmsCallableMessageStorageType		
	5.13.2.4	dmsCallableMessageRowStatus		
3.5.3.3.2 : Monitor Short Power Recovery Message			26048-1§9.2.1 : Get elemental data	
	5.7.8	dmsShortPowerRecoveryMessage		
3.5.3.3.3 : Monitor Long Power Recovery Message			26048-1§9.2.1 : Get elemental data	
	5.7.9	dmsLongPowerRecoveryMessage		
	5.7.10	dmsShortPowerLossTime		
3.5.3.3.4 : Monitor Power Loss Message			26048-1§9.2.1 : Get elemental data	

Requirement	Object Reference	Object	Dialogue	Notes
	5.7.14	dmsPowerLossMessage		
3.5.3.3.5 : Monitor Reset Message			26048-1§9.2.1 : Get elemental data	
	5.7.11	dmsResetMessage		
3.5.3.3.6 : Monitor Communications Loss Message			26048-1§9.2.1 : Get elemental data	
	5.7.12	dmsCommunicationsLossMessage		
	5.7.13	dmsTimeCommLoss		
3.5.3.3.7 : Monitor End Duration Message			26048-1§9.2.1 : Get elemental data	
	5.7.15	dmsEndDurationMessage		
3.5.3.3.8 : Retrieve a DMS Message Action			26048-1§9.2.5 : Get tabular data	
	5.9.2.2	dmsActionMsgCode		
3.5.3.3.9 : Determine Number of DMS Message Actions			26048-1§9.2.1 : Get elemental data	
	5.9.1	numActionTableEntries		
3.5.3.3.10 : Determine Number of Callable Messages			26048-1§9.2.1 : Get elemental data	
	5.13.1	dmsNumCallableMessages		

A.4 MULTI Field Traceability Matrix

The following table provides an implementer / tester with the traceability of requirements to particular MULTI Tags (defined in Section 6).

MULTI Field Traceability Matrix				
Requirement ID	**Requirement**	**MULTI Tag ID**	**MULTI Tag Name**	**MULTI Tag**
3.6.6.2.1	Support Multi-Page Messages			
		6.4.15	New Page	[np]
3.6.6.2.2	Support Page Justification			
3.6.6.2.2.1	Support for One Page Justification within a Message			

MULTI Field Traceability Matrix				
Requirement ID	Requirement	MULTI Tag ID	MULTI Tag Name	MULTI Tag
		6.4.11	Justification - Page	[jpx]
3.6.6.2.2.2	Support for Multiple Page Justifications within a Message			
		6.4.11	Justification - Page	[jpx]
3.6.6.2.3	Support Multiple Line Messages			
		6.4.14	New Line	[nlx]
3.6.6.2.4	Support Line Justification			
		6.4.10	Justification - Line	[jlx]
3.6.6.2.2.1	Support for a Single Line Justification within a Message			
		6.4.10	Justification - Line	[jlx]
3.6.6.2.4.2	Support Line Justification on a Page-by-Page Basis			
		6.4.10	Justification - Line	[jlx]
3.6.6.2.4.3	Support Line Justification on a Line-by-Line Basis			
		6.4.10	Justification - Line	[jlx]
3.6.6.2.5	Support Color			
3.6.6.2.5.1	Support a Single Color Combination per Message			
		6.4.2	Page Background Color	[pbz] or [pbr,g,b]
		6.4.3	Color Foreground	[cfx]
3.6.6.2.5.2	Support a Color Combination for each Page			
		6.4.2	Page Background Color	[pbz] or [pbr,g,b]
		6.4.3	Color Foreground	[cfx]
3.6.6.2.5.3	Support a Color Combination for each Character within a Message			
		6.4.2	Page Background Color	[pbz] or [pbr,g,b]
		6.4.3	Color Foreground	[cfx]
3.6.6.2.5.4	Color for each Pixel within a Message			

MULTI Field Traceability Matrix				
Requirement ID	**Requirement**	**MULTI Tag ID**	**MULTI Tag Name**	**MULTI Tag**
		6.4.2	Page Background Color	[pbz] or [pbr,g,b]
		6.4.3	Color Foreground	[cfx]
		6.4.4	Color Rectangle	[crx,y,w,h,r,g,b] or [crx,y,w,h,z]
3.6.6.2.6	Support Font Commands			
		6.4.7	Font	[fox]
3.6.6.2.7	Support Moving Text			
		6.4.13	Moving Text	[mvtdw,s,r,text]
3.6.6.2.8	Support Character Spacing			
		6.4.17	Spacing - Character	[scx]
3.6.6.2.9	Support Customizable Page Display Times in a Message			
		6.4.16	Page Time	[ptxoy]
3.6.6.2.10	Support Flashing			
		6.4.6 Flash Time	[fltxoy]	
3.6.6.2.11	Support Customizable Flashing Times within a Message			
		6.4.6	Flash Time	[fltxoy]
3.6.6.2.12	Support Hexadecimal Character			
		6.4.9	Hexadecimal Character	[hcx]
3.6.6.2.13	Support Message Data Fields			
3.6.6.2.13.1	Support Current Time Field without AM/PM Field			
		6.4.5	Local Time 12 Hour	[f1,y]
		6.4.5	Local Time 24 Hour	[f2,y]
3.6.6.2.13.2	Support Current Time with uppercase AM/PM Field			
		6.4.5	Local time, 12 hour format with capital AM/PM indicator present	[f12,y]

MULTI Field Traceability Matrix				
Requirement ID	**Requirement**	**MULTI Tag ID**	**MULTI Tag Name**	**MULTI Tag**
3.6.6.2.13.3	Support Current Time with lowercase am/pm			
		6.4.5	Local time, 12 hour format with lowercase am/pm indicator present	[f13,y]
3.6.6.2.13.4	Support Current Temperature Field			
		6.4.5	Ambient Temperature Celsius	[f3,y]
		6.4.5	Ambient Temperature Fahrenheit	[f4,y]
3.6.6.2.13.5	Support Detected Vehicle Speed Field			
		6.4.5	Speed km/h	[f5,y]
		6.4.5	Speed mph	[f6,y]
3.6.6.2.13.6	Support Current Day of Week Field			
		6.4.5	Day of Week	[f7,y]
3.6.6.2.13.7	Support Current Day of Month Field			
		6.4.5	Date of Month	[f8,y]
3.6.6.2.13.8	Support Current Month of Year Field			
		6.4.5	Month of Year	[f9,y]
3.6.6.2.13.9	Support Current Year Field			
		6.4.5	Year 2 Digit	[f10,y]
		6.4.5	Year 4 Digit	[f11,y]
3.6.6.2.13.10	Support User-Definable Field			
		6.4.5	User-Definable Field	[f50,y] to [f99,y]
3.6.6.2.13.11	Data Field Refresh Rate			
		6.4.5	Fields	[fx,y]
3.6.6.2.14	Support of Graphics			
		6.4.8	Graphic	[gn] or [gn,x,y] or [gn,x,y,cccc]
3.6.6.2.15	Specify Location of Message Display			
		6.4.18	Text Rectangle	[trx,y,w,h]
		6.4.2	Page Background Color	[pbz] or [pbr,g,b]

MULTI Field Traceability Matrix				
Requirement ID	**Requirement**	**MULTI Tag ID**	**MULTI Tag Name**	**MULTI Tag**
		6.4.3	Color Foreground	[cfx]
		6.4.4	Color Rectangle	[crx,y,w,h,r,g,b] or [crx,y,w,h,z]
3.6.6.2.17	Support of Manufacturer Specfiic Message Definitions			
		6.4.12	Manufacturer Specfiic Tag	[msx,y]
3.6.8	Supplemental Requirements for Color Scheme			
3.6.8.2	Support Classic NTCIP Scheme			
		6.4.2	Page Background Color	[pbz] or [pbr,g,b]
		6.4.3	Color Foreground	[cfx]
		6.4.4	Color Rectangle	[crx,y,w,h,r,g,b] or [crx,y,w,h,z]
3.6.8.3	Support 24-Bit Color Scheme			
		6.4.2	Page Background Color	[pbz] or [pbr,g,b]
		6.4.3	Color Foreground	[cfx]
		6.4.4	Color Rectangle	[crx,y,w,h,r,g,b] or [crx,y,w,h,z]
3.6.8.4	Support Single Color			
		6.4.2	Page Background Color	[pbz] or [pbr,g,b]
		6.4.3	Color Foreground	[cfx]
3.6.12	Supplemental Requirements for Page Justification			
3.6.12.1	Support Top Page Justification			
		6.4.11	Top Justification	[jp2]
3.6.12.2	Support Middle Page Justification			
		6.4.11	Middle Justification	[jp3]
3.6.12.3	Support Bottom Page Justification			
		6.4.11	Bottom Justification	[jp4]
3.6.13	Supplemental Requirements for Line			

MULTI Field Traceability Matrix				
Requirement ID	**Requirement**	**MULTI Tag ID**	**MULTI Tag Name**	**MULTI Tag**
	Justification			
3.6.13.1	Support Left Line Justification			
		6.4.10	Left Justification	[jl2]
3.6.13.2	Support Center Line Justification			
		6.4.10	Center Justification	[jl3]
3.6.13.3	Support Right Line Justification			
		6.4.10	Right Justification	[jl4]
3.6.13.4	Support Full Line Justification			
		6.4.10	Full Justification	[jl5]

A.5 Version Conformance

Per RFC 2580, MIB modules can include MODULE-COMPLIANCE macros to define:

- When each object-group must be supported.
- When object-types can be sub-ranged within an AGENT-CAPABILITIES statement,
- To what extent access to objects can be restricted within implementations,
- Which object types were introduced and deprecated within each version of the standard,
- When refinements were made to the valid syntax of objects (e.g., new or deprecated enumerations),

IETF MIB modules typically contain one MODULE-COMPLIANCE statement for each revision of the module.

NTCIP standards define when each object-type must be supported using the PRL and RTM traceability tables. These tables allow the reader to better understand the user needs behind each object-type. Some of the other details mentioned above are often discussed within object descriptions, requirements, or other locations within the standard. However, some issues, such as when objects were added or ranges changed are seldom discussed in other portions of the document.

Rather than using MODULE-COMPLIANCE macros within each MIB module, the NTCIP standards include a version conformance table and an object refinement table to provide identical information in a more concise manner that does not complicate the MIB modules. These tables are provided to ensure that these implementation issues are documented within the standard and to provide a single summary area for this information. It is recognized that these tables duplicate and extend requirements contained within object-type descriptions, the PRL, and the RTM. In case of any conflict between these sources the object-type descriptions, PRL, and RTM take precedence over the version conformance table and object refinement table.

A.1.1 Version Conformance

Table 10 indicates which object groups and capabilities are required for each version of NTCIP 1203 for the objects contained within this MIB, starting with NTCIP 1203 v04, the first version of this document to include object group definitions. The predicates used in this table are defined internal to this table.

Table 10: Version Conformance for NTCIP1203-Dms MIB

Group	MIB	v04 (current)
fdActionGroupV1 (action)	ISO26048-1-Action	condTrigger OR dayPlan OR schedTrigger:M
fdGroupV1	ISO26048-1-FieldDevice	M
fdGnssGroupV1	ISO26048-1-FieldDevice	O
fdUtcClockGroupV1	ISO26048-1-Clock	M
fdLocalClockGroupV1	ISO26048-1-Clock	dayPlanTrigger OR schedTrigger OR Time:M
fdDstClockGroupV1	ISO26048-1-Clock	dayPlanTrigger OR schedTrigger OR Time:O
fdCommandGroupV1	ISO26048-1-Command	O
fdCondTriggerGroupV1 (condTrigger)	ISO26048-1-CondTrigger	O
fdControllerGroupV1	ISO26048-1-Controller	M
fdControllerStartupGroupV1	ISO26048-1-Controller	O
fdDayPlanSchedulerGroupV1 (dayPlanTrigger)	ISO26048-1-DayPlan	O
fdDynObjBasicGroupV1 (dynObj)	ISO26048-1-DynObj	O

Group	MIB	v04 (current)
fdDynObjTwoStepGroupV1	ISO26048-1-DynObj	dynObj:O
fdDynObjSettableGroupV1	ISO26048-1-DynObj	dynObj:O
fdLogConfigGroupV1	ISO26048-1-Log	O.1 (1..*)
fdNotifyGroupV1 (notify)	ISO26048-1-Notification	O.1 (1..*)
fdNotificationIndependentGroupV1	ISO26048-1-Notification	notify AND independent:M
fdOwnerGroupV1	ISO26048-1-Owner	action: M
fdRecordingGroupV1	ISO26048-1-Recording	O
fdSchedTriggerGroupV1 (schedTrigger)	ISO26048-1-SchedTrigger	O
fdSrsaBasicGroupV1 (srsa)	ISO26048-1-Srsa	autoBright:M; O
fdSrsaOutputGroupV1	ISO26048-1-Srsa	srsa:O
fdSrsaInputGroupV1	ISO26048-1-Srsa	autoBright:M; srsa:O
fdSrsaActiveTestGroupV1	ISO26048-1-Srsa	srsa:O
fdSrsaTypeValueStatsGroupV1	ISO26048-1-Srsa	autoBright; srsa:O
fdSyslogGroupV1	ISO26048-1-Syslog	M
fdSyslogQueryGroupV1	ISO26048-1-Syslog	O
fdTransactionGroupV1	ISO26048-1-Transaction	X
dmsCoreGroupV1	NTCIP1203-Dms	M
vmsCoreGroupV1	NTCIP1203-Dms	O
dmsMatrixGroupV1	NTCIP1203-Dms	O
dmsBeaconGroupV1	NTCIP1203-Dms	O
dmsDisplayCapabilitiesGroupV1	NTCIP1203-Dms	O
dmsDisplayLocationGroupV1	NTCIP1203-Dms	O
dmsCurrentMessageGroupV1	NTCIP1203-Dms	O
dmsFontGroupV1	NTCIP1203-Dms	O
dmsGraphicsGroupV1	NTCIP1203-Dms	O
dmsDrumGroupV1	NTCIP1203-Dms	O
dmsPixelServiceGroupV1	NTCIP1203-Dms	O
dmsIlluminationGroupV1	NTCIP1203-Dms	O
dmsAutoBrightGroupV1 (autoBright)	NTCIP1203-Dms	O
dmsScheduleGroupV1	NTCIP1203-Dms	dayPlanTrigger:M
dmsCallableMessageGroupV1	NTCIP1203-Dms	condTrigger OR schedTrigger: M
dmsPixelGroupGroupV1	NTCIP1203-Dms	O
dmsMultiFieldsGroupV1	NTCIP1203-Dms	O
dmsSpeedLimitGroupV1	NTCIP1203-Dms	O
dmsShortPowerRecoveryGroupV1	NTCIP1203-Dms	O
dmsLongPowerRecoveryGroupV1	NTCIP1203-Dms	O
dmsPowerLossGroupV1	NTCIP1203-Dms	O
dmsResetMessageGroupV1	NTCIP1203-Dms	O
dmsCommLossGroupV1	NTCIP1203-Dms	O
dmsEndDurationGroupV1	NTCIP1203-Dms	O

A.6 Object Refinements

To conform to this MIB, implementations shall:

- support the defined MAX-ACCESS of each supported object, unless the object refinement table indicates a different minimum access;
- support at least the access defined when the object refinement table indicates a refined access for the object type;
- support the full range of values defined by the SYNTAX for each supported object, unless the

object refinement table indicates a different SYNTAX;

- support at least the range of values defined when the object refinement table indicates a refined syntax for the object type.

NOTE 1— The refined access or syntax includes any text in the DESCRIPTION field within the Table 11.

NOTE 2— Information in the table regarding versions prior to v04 are informative interpretations of the previous versions and not normative.

Table 11 indicates the object refinements for the object types used by NTCIP 1203. Refinements 1-3 are defined as follows:

1. `MIN-ACCESS read-only`
   `DESCRIPTION "Values within permanent rows shall be read-only; values in all other rows shall allow full access."`
2. `DESCRIPTION "Implementations are limited to NVT-ASCII characters."`
3. `DESCRIPTION "Implementations shall support all NVT-ASCII characters and any other UTF-8 characters identified in the PICS."`

NOTE—Full access equates to either read-write or read-create, as appropriate.

Table 11: Compliance Statement for NTCIP1203-Dms MIB

Obj	Versions	Refinement			
		1	2	3	Additional Refinements
dmsBeaconPlanIndex					
	v04				SYNTAX INTEGER (1) DESCRIPTION "Implementations that support beacons shall support indexes starting at 1 and incrementing by one for each beacon. The required number of supported beacons is defined in the PICS with a minimum value of 0."
dmsBeaconPlanDescr					
	v04	x			
dmsBeaconPlanCycle					
	v04	x			
dmsBeaconPlanStorageType					
	v04	x			
dmsBeaconPlanRowStatus					
	v04	x			
dmsBeaconPlanDetailCycle					
	v04	x			DESCRIPTION " An implementation may constrain the value to be equal to the dmsBeaconPlanCycle or an divisor thereof."
dmsBeaconPlanDetailOffset					
	v04	x			
dmsBeaconPlanDetailFlash					
	v04	x			
dmsBeaconPlanDetailColor					
	v04	x			

Obj	Versions	Refinement			
		1	2	3	Additional Refinements
fontIndex					
	v01, v01A1, v02, v03, v04				SYNTAX INTEGER (1) DESCRIPTION "Implementations that support fonts shall support font indexes starting at 1 and incrementing by one for each font. The required number of supported fonts is defined in the PICS with a minimum value of 1."
fontNumber					
	v02, v03, v04	x			
fontName					
	v01, v01A1		x		
	v02, v03	x	x		
	v04	x		x	
fontHeight					
	v01, v01A1				MIN-ACCESS read-only DESCRIPTION "Implementations shall subrange this object as defined in the object-type description."
	v02, v03, v04	x			DESCRIPTION " Implementations shall subrange this object as defined in the object-type description."
fontCharSpacing					
	v01, v01A1				MIN-ACCESS read-only DESCRIPTION "Implementations shall subrange this object as defined in the object-type description."
	v02, v03, v04	x			DESCRIPTION " Implementations shall subrange this object as defined in the object-type description."
fontLineSpacing					
	v01, v01A1				MIN-ACCESS read-only DESCRIPTION "Implementations shall subrange this object as defined in the object-type description."
	v02, v03, v04	x			DESCRIPTION "Implementations shall subrange this object as defined in the object-type description."
fontStatus					
	v02, v03	x			MIN-ACCESS read-only SYNTAX { notUsed (1), modifying (2), calculatingID (3), readyForUse (4), inUse (5), permanent (6), unmanaged (11) } WRITE-SYNTAX INTEGER { modifyReq (7), readyForUseReq (8), notUsedReq (9),

Obj	Versions	Refinement			
		1	2	3	Additional Refinements
					unmanagedReq (10) } DESCRIPTION " Seven values are status values (read-only) and four values are command values (write-only). The values unmanagedReq (10) and unmanaged (11) are defined to provide backwards compatibility with v01 devices. "
	v04	x			MIN-ACCESS read-only SYNTAX { notUsed (1), modifying (2), calculatingID (3), readyForUse (4), inUse (5), permanent (6) } WRITE-SYNTAX INTEGER { modifyReq (7), readyForUseReq (8), notUsedReq (9) } DESCRIPTION " Seven values are status values (read-only) and four values are command values (write-only). The values unmanagedReq (10) and unmanaged (11) were defined to provide backwards compatibility with v01 devices and are considered obsolete when interfacing via SNMPv3. "
characterNumber					
	v01, v01A1, v02, v03, v04				DESCRIPTION "Implementations that support fonts shall support the character numbers identified in the PICS. Supported character numbers are not required to be sequential."
characterWidth					
	v01, v01A1				DESCRIPTION "Implementations shall subrange this object as defined in the object-type description."
	v02, v03, v04	x			MIN-ACCESS read-only DESCRIPTION "Implementations shall subrange this object as defined in the object-type description."
characterBitmap					
	v01, v01A1				DESCRIPTION "Implementations shall subrange this object as defined in the object-type description."
	v02, v03, v04	x			MIN-ACCESS read-only DESCRIPTION "Implementations shall subrange this object as defined in the object-type description."
defaultBackgroundColor					
	v01, v01A1, v02, v03				DESCRIPTION "Implementations shall support color codes as identified by the PICS. Implementations shall subrange this object to only those colors supported. Implementations may make this object read-only."
defaultForegroundColor					
	v01, v01A1, v02, v03				DESCRIPTION " Implementations shall support color codes as identified by the PICS. Implementations shall subrange this object to only those colors supported. Implementations may make this object read-only."
defaultFlashOn					

Obj	Versions	Refinement			
		1	2	3	Additional Refinements
	v01, v01A1, v02, v03, v04				MIN-ACCESS read-only DESCRIPTION "Implementations may make this object read-only. Implementations that are writable shall support flash-on times as identified in the PICS. Supported times are not required to be sequential. An implementation that supports this object but does not support flashing can subrange this object to the single value of zero (0)."
defaultFlashOff					
	v01, v01A1, v02, v03, v04				MIN-ACCESS read-only DESCRIPTION "Implementations may make this object read-only. Implementations that are writable shall support flash-off times as identified in the PICS. Supported times are not required to be sequential. An implementation that supports this object but does not support flashing can subrange this object to the single value of zero (0)."
defaultFont					
	v01, v01A1, v02, v03, v04				MIN-ACCESS read-only DESCRIPTION "Implementations may make this object read-only. If the DMS supports a single font, the object shall be read-only and reference the fontNumber of the font."
defaultJustificationLine					
	v01, v01A1, v02, v03, v04				MIN-ACCESS read-only DESCRIPTION "Implementations may make this object read-only. Implementations that are writable shall support the line justification modes identified in the PICS."
defaultJustificationPage					
	v01, v01A1, v02, v03, v04				MIN-ACCESS read-only DESCRIPTION "Implementations may make this object read-only. Implementations that are writable shall support the page justification modes identified in the PICS."
defaultPageOnTime					
	v01, v01A1, v02, v03, v04				DESCRIPTION "Implementations may make this object read-only. Implementations that are writable shall support page-on times as identified in the PICS. Supported times are not required to be sequential."
defaultPageOffTime					
	v01, v01A1, v02, v03, v04				DESCRIPTION "Implementations may make this object read-only. Implementations that are writable shall support page-off times as identified in the PICS. Supported times are not required to be sequential."
defaultBackgroundRGB					
	v02, v03, v04				DESCRIPTION "Implementations may make this object read-only. Implementations that are writable shall support color codes as identified by the PICS. Implementations shall subrange this object

Obj	Versions	Refinement			
		1	2	3	Additional Refinements
					to only those colors supported."
defaultForegroundRGB					
	v02, v03, v04				DESCRIPTION "Implementations may make this object read-only. Implementations that are writable shall support color codes as identified by the PICS. Implementations shall subrange this object to only those colors supported."
defaultCharacterSet					

	v01,
	v01A1,
	v02,
	v03
					SYNTAX { other (1),
					         eightBit (2) }
					WRITE-SYNTAX { eightBit (2) }
					DESCRIPTION "The value 'other' is not writeable but an implementation may transition to 'other' to indicate that another characterSet has been invoked by a means not defined by this object. Prior to v04, 'utf8Style' was not defined."

	v04				SYNTAX { other (1), eightBit (2) , utf8Style (3) } WRITE-SYNTAX { eightBit (2), utf8Style (3) } DESCRIPTION "The value 'other' is not writeable but an implementation may transition to 'other' to indicate that another characterSet has been invoked by a means not defined by this object."
dmsMessageMemoryType					

	v01A1,
	v02,
	v03,
	v04
					SYNTAX INTEGER {
					                 currentBuffer (5),
					                 blank (7) }
					DESCRIPTION "The value of 'other' is reserved for manufacturer specific use and is not expected to be widely used. Implementations shall support at least one of 'permanent', 'changeable', or 'volatile' in addition to 'currentBuffer' and 'blank'. Support for 'schedule' is conditional based on support for dmsScheduleGroupV1."

dmsMessageNumber					
	v01				DESCRIPTION "Range is limited based on memory type as follows: permanent: per PICS changeable: per PICS volatile: per PICS currentBuffer: (1) schedule: (1)"
	v01A1, v02, v03, v04				DESCRIPTION "Range is limited based on memory type as follows: permanent: per PICS changeable: per PICS volatile: per PICS currentBuffer: (1) schedule: (1) blank: (1..255) "
dmsMessageMultiString					
	v01				MIN-ACCESS read-only DESCRIPTION "When dmsMessageMemoryType is 'permanent', 'currentBuffer', or 'schedule', the access of this object shall be

Obj	Versions	Refinement			
		1	2	3	Additional Refinements
					read-only; for all other cases, the access of this object shall be read-write."
	v01A1, v02, v03, v04				MIN-ACCESS read-only DESCRIPTION "When dmsMessageMemoryType is 'permanent', 'currentBuffer', 'schedule', or 'blank', the access of this object shall be read-only; for all other cases, the access of this object shall be read-write."
dmsMessageOwner					
	v01,		x		MIN-ACCESS read-only DESCRIPTION "When dmsMessageMemoryType is 'permanent', 'currentBuffer', or 'schedule', the access of this object shall be read-only; for all other cases, the access of this object shall be read-write."
	v01A1, v02, v03		x		MIN-ACCESS read-only DESCRIPTION "When dmsMessageMemoryType is 'permanent', 'currentBuffer', 'schedule', or 'blank', the access of this object shall be read-only; for all other cases, the access of this object shall be read-write."
	v04			x	MIN-ACCESS read-only DESCRIPTION "When dmsMessageMemoryType is 'permanent', 'currentBuffer', 'schedule', or 'blank', the access of this object shall be read-only; for all other cases, the access of this object shall be read-write."
dmsMessageBeacon					
	v01				SYNTAX Integer32 (0..1) MIN-ACCESS read-only DESCRIPTION "When dmsMessageMemoryType is 'permanent', 'currentBuffer', or 'schedule', the access of this object shall be read-only; for all other cases, the access of this object shall be read-write."
	v01A1, v02, v03				SYNTAX Integer32 (0..1) MIN-ACCESS read-only DESCRIPTION "When dmsMessageMemoryType is 'permanent', 'currentBuffer', 'schedule', or 'blank', the access of this object shall be read-only; for all other cases, the access of this object shall be read-write."
	v04				SYNTAX Integer32 (0) MIN-ACCESS read-only DESCRIPTION "When dmsMessageMemoryType is 'permanent', 'currentBuffer', 'schedule', or 'blank', the access of this object shall be read-only; for all other cases, the access of this object shall be read-write. Support for this object is required, but it shall be subranged to only support the beacon plans that are available, which may limit the value to 0."
dmsMessagePixelService					
	v01				MIN-ACCESS read-only DESCRIPTION "When dmsMessageMemoryType is 'permanent', 'currentBuffer', or 'schedule', the access of this object shall be read-only; for all other cases, the access of this object shall be read-write."
	v01A1,				MIN-ACCESS read-only

Obj	Versions	Refinement			
		1	2	3	Additional Refinements
	v02, v03				DESCRIPTION "When dmsMessageMemoryType is 'permanent', 'currentBuffer', 'schedule', or 'blank', the access of this object shall be read-only; for all other cases, the access of this object shall be read-write."
	v04				SYNTAX Integer32 (0) MIN-ACCESS read-only DESCRIPTION "When dmsMessageMemoryType is 'permanent', 'currentBuffer', 'schedule', or 'blank', the access of this object shall be read-only; for all other cases, the access of this object shall be read-write. Support for this object is required, but implementations that do not support pixel service shall restrict the value to 0."
dmsMessageRunTimePriority					
	v01				MIN-ACCESS read-only DESCRIPTION "When dmsMessageMemoryType is 'currentBuffer', the access of this object shall be read-only; for all other cases, the access of this object shall be read-write."
	v01A1, v02, v03				MIN-ACCESS read-only DESCRIPTION "When dmsMessageMemoryType is 'permanent', 'currentBuffer' or 'blank', the access of this object shall be read-only; for all other cases, the access of this object shall be read-write."
	v04				MIN-ACCESS read-only DESCRIPTION "When dmsMessageMemoryType is 'currentBuffer', 'blank', or 'permanent', the access of this object shall be read-only; for all other cases, the access of this object shall be read-write."
dmsMessageStatus					
	v01				SYNTAX { notUsed (1), modifying (2), validating (3), valid (4), error (5) } WRITE-SYNTAX INTEGER { modifyReq (7), validateReq (8), notUsedReq (9) } DESCRIPTION "Five values are status values (read-only) and three values are command values (write-only). When dmsMessageMemoryType is 'permanent' or 'currentBuffer', the access of this object shall be read-only; for all other cases, the access of this object shall be read-write."
	v01A1, v02, v03, v04				SYNTAX { notUsed (1), modifying (2), validating (3), valid (4), error (5) } WRITE-SYNTAX INTEGER { modifyReq (7), validateReq (8), notUsedReq (9) } DESCRIPTION "Five values are status values (read-only) and three values are command values (write-only). When dmsMessageMemoryType is 'permanent', 'currentBuffer', or 'blank', the access of this object shall be read-only; for all other cases, the access of this

Obj	Versions	Refinement			
		1	2	3	Additional Refinements
					object shall be read-write."
	v01A1, v02, v03, v04				SYNTAX { notUsed (1), modifying (2), validating (3), valid (4), error (5) } WRITE-SYNTAX INTEGER { modifyReq (7), validateReq (8), notUsedReq (9) } DESCRIPTION "Five values are status values (read-only) and three values are command values (write-only). When dmsMessageMemoryType is 'currentBuffer' or 'blank', the access of this object shall be read-only; for all other cases (including 'permanent'), the access of this object shall be read-write (e.g., to allow for modifying run-time priority as a minimum."
dmsControlMode					
	v01, v01A1				SYNTAX { other (1), local (2), external (3), central (4), centralOverride (5), simulation (6) } WRITE-SYNTAX INTEGER { local (2), external (3), central (4), centralOverride (5), simulation (6) } DESCRIPTION "The value 'other' is not writeable but an implementation may transition to 'other' to indicate that another mode has been invoked by a means not defined by this object."
	v02, v03, v04				SYNTAX { local (2), central (4), centralOverride (5) } DESCRIPTION "The values 'other', 'external', and 'simulation' were deprecated in v02."
dmsMsgSourceMode					
	v02, v03, v04				SYNTAX { other (1), local (2), external (3), central (8), timebasedScheduler (9), powerRecovery (10), reset (11), commLoss (12), powerLoss (13), endDuration (14) } DESCRIPTION "The values for 'otherCom1', 'otherCom2', 'otherCom3', and 'otherCom4' were deprecated in v02."
dmsMemoryMgmt					
	v02, v03,				SYNTAX { normal (2), clearChanhgeableMessages (3),

Obj	Versions	Refinement			
		1	2	3	Additional Refinements
	v04				clearVolatileMessages (4) } DESCRIPTION "The value for 'other' was deprecated in v02."
dmsActivateMsgError					
	v01, v01A1				SYNTAX { other (1), none (2), priority (3), messageStatus (4), messageMemoryType (5), messageNumber (6), messageCRC (7), syntaxMULTI (8), localMode (9) } DESCRIPTION "The values for 'centralMode' and 'centralOverrideMode' were added in v02."
dmsMultiSyntaxError					
	v01, v01A1				SYNTAX { other (1), none (2), unsupportedTag (3), unsupportedTagValue (4), textTooBig (5), fontNotDefined (6), characterNotDefined (7), fieldDeviceNotExist (8), fieldDeviceError (9), flashRegionError (10), tagConflict (11), tooManyPages (12) } DESCRIPTION "The values for 'fontVersionID', 'graphicID', and 'graphicNotDefined' were added in v02. The value for 'invalidCharacterEncoding' was added in v04."
	v02, v03				SYNTAX { other (1), none (2), unsupportedTag (3), unsupportedTagValue (4), textTooBig (5), fontNotDefined (6), characterNotDefined (7), fieldDeviceNotExist (8), fieldDeviceError (9), flashRegionError (10), tagConflict (11), tooManyPages (12) , fontVersionID (13), graphicID (14), graphicNotDefined (15) } DESCRIPTION "The value for 'invalidCharacterEncoding' was added in v04."
dmsMultiOtherErrorDescription					
	v01, v01A1, v02, v03		x		

Obj	Versions	Refinement			
		1	2	3	Additional Refinements
dmsMsgRequesterAddressType					
	v04				DESCRIPTION "The implementation shall support the address types identified in the PICS."
dmsIllumControl					
	v01, v01A1				SYNTAX { other (1), photocell (2), timer (3), manual (4) } DESCRIPTION "The values for 'manualDirect' and 'manualIndexed' were added v02."
	v02, v03, v04				SYNTAX { other (1), photocell (2), timer (3), manualDirect (5), manualIndexed (6) } DESCRIPTION "The value for 'manual' was deprecated v02."
dmsIllumBrightLevelStatus					
	v01, v01A1, v02, v03, v04				DESCRIPTION "The value is subranged with an upper limit as defined by dmsIllumNumBrightLevels."
dmsIllumManLevel					
	v01, v01A1, v02, v03, v04				DESCRIPTION "The value is subranged with an upper limit as defined by dmsIllumNumBrightLevels."
dmsActionIndex					
	v01, v01A1, v02, v03, v04				DESCRIPTION " Implementations shall support action indexes starting at 1 and incrementing by one for each action. The required number of supported actions is defined in the PICS with a minimum value of 1."
pixelFailureDetectionType					
	v02, v03, v04				SYNTAX { pixelTest (2), messageDisplay (3) } DESCRIPTION "The value for 'other' was deprecated in v02."
pixelTestActivation					
	v02, v03, v04				SYNTAX { noTest (2), test (3), clearTable (4) } WRITE-SYNTAX { test (3), clearTable (4) } DESCRIPTION "The value for 'other' was deprecated in v02. the value for 'noTest' is not writeable."
dmsGraphicIndex					
	v02, v03,				DESCRIPTION " Implementations shall support graphic indexes starting at 1 and incrementing by one for each graphic. The

Obj	Versions	Refinement			
		1	2	3	Additional Refinements
	v04				required number of supported graphics is defined in the PICS with a minimum value of 1."
dmsGraphicNumber					
	v02, v03, v04	x			
dmsGraphicName					
	v02, v03	x	x		
	v04	x		x	
dmsGraphicHeight					
	v02, v03, v04	x			DESCRIPTION "Implementations shall subrange this object as defined in the object-type description."
dmsGraphicWidth					
	v02, v03, v04	x			DESCRIPTION "Implementations shall subrange this object as defined in the object-type description."
dmsGraphicType					
	v02, v03, v04	x			DESCRIPTION "Implementations shall support the color schemes identified in the PICS."
dmsGraphicTransparentColor					
	v02, v03, v04	x			DESCRIPTION "Implementations shall subrange this object as defined in the object-type description."
dmsGraphicStatus					
	v02, v03, v04	x			SYNTAX { notUsed (1), modifying (2), calculatingID (3), readyForUse (4), inUse (5), permanent (6) } WRITE-SYNTAX { modifyReq (7), readyForUseReq (8), notUsedReq (9) } DESCRIPTION "Six values are status values (read-only) and three values are command values (write-only)."
dmsGraphicBlockBitmap					
	v02, v03, v04	x			
dmsCallableMessageCode					
	v04	x			
dmsCallableMessageStorageType					
	v04	x			

Annex B
Object Tree [Informative]

Table 12 provides a concise listing of the objects defined in this document, identifying how object definitions are combined under specific nodes, their status, and their accessibility.

Table 12: Listing of MIB Objects

Object	OID	Status	Access
dmsSignAccess	1.3.6.1.4.1.1206.4.2.3.1.1	current	read-only
dmsSignType	1.3.6.1.4.1.1206.4.2.3.1.2	current	read-only
dmsSignHeight	1.3.6.1.4.1.1206.4.2.3.1.3	current	read-only
dmsSignWidth	1.3.6.1.4.1.1206.4.2.3.1.4	current	read-only
dmsHorizontalBorder	1.3.6.1.4.1.1206.4.2.3.1.5	current	read-only
dmsVerticalBorder	1.3.6.1.4.1.1206.4.2.3.1.6	current	read-only
dmsLegend	1.3.6.1.4.1.1206.4.2.3.1.7	current	read-only
dmsBeaconType	1.3.6.1.4.1.1206.4.2.3.1.8	deprecated	read-only
dmsSignTechnology	1.3.6.1.4.1.1206.4.2.3.1.9	current	read-only
dmsDisplayLatitude	1.3.6.1.4.1.1206.4.2.3.1.10	current	read-only
dmsDisplayLongitude	1.3.6.1.4.1.1206.4.2.3.1.11	current	read-only
dmsDisplayDirection	1.3.6.1.4.1.1206.4.2.3.1.12	current	read-write
dmsDisplayOffsetLatitude	1.3.6.1.4.1.1206.4.2.3.1.13	current	read-write
dmsDisplayOffsetLongitude	1.3.6.1.4.1.1206.4.2.3.1.14	current	read-write
dmsBeaconCount	1.3.6.1.4.1.1206.4.2.3.1.15.1	current	read-only
dmsBeaconTable	1.3.6.1.4.1.1206.4.2.3.1.15.2	current	not-accessible
dmsBeaconEntry	1.3.6.1.4.1.1206.4.2.3.1.15.2.1	current	not-accessible
dmsBeaconIndex	1.3.6.1.4.1.1206.4.2.3.1.15.2.1.1	current	not-accessible
dmsBeaconXLocation	1.3.6.1.4.1.1206.4.2.3.1.15.2.1.2	current	read-only
dmsBeaconYLocation	1.3.6.1.4.1.1206.4.2.3.1.15.2.1.3	current	read-only
dmsBeaconWidth	1.3.6.1.4.1.1206.4.2.3.1.15.2.1.4	current	read-only
dmsBeaconHeight	1.3.6.1.4.1.1206.4.2.3.1.15.2.1.5	current	read-only
dmsBeaconColorScheme	1.3.6.1.4.1.1206.4.2.3.1.15.2.1.6	current	read-only
dmsBeaconFlashResolution	1.3.6.1.4.1.1206.4.2.3.1.15.2.1.7	current	read-only
dmsBeaconPlanTable	1.3.6.1.4.1.1206.4.2.3.1.15.3	current	not-accessible
dmsBeaconPlanEntry	1.3.6.1.4.1.1206.4.2.3.1.15.	current	not-

	3.1		accessible
dmsBeaconPlanIndex	1.3.6.1.4.1.1206.4.2.3.1.15.3.1.1	current	not-accessible
dmsBeaconPlanDescr	1.3.6.1.4.1.1206.4.2.3.1.15.3.1.2	current	read-create
dmsBeaconPlanCycle	1.3.6.1.4.1.1206.4.2.3.1.15.3.1.3	current	read-create
dmsBeaconPlanStorageType	1.3.6.1.4.1.1206.4.2.3.1.15.3.1.4	current	read-create
dmsBeaconPlanRowStatus	1.3.6.1.4.1.1206.4.2.3.1.15.3.1.5	current	read-create
dmsBeaconPlanDetailTable	1.3.6.1.4.1.1206.4.2.3.1.15.4	current	not-accessible
dmsBeaconPlanDetailEntry	1.3.6.1.4.1.1206.4.2.3.1.15.4.1	current	not-accessible
dmsBeaconPlanDetailCycle	1.3.6.1.4.1.1206.4.2.3.1.15.4.1.1	current	read-create
dmsBeaconPlanDetailOffset	1.3.6.1.4.1.1206.4.2.3.1.15.4.1.2	current	read-create
dmsBeaconPlanDetailFlash	1.3.6.1.4.1.1206.4.2.3.1.15.4.1.3	current	read-create
dmsBeaconPlanDetailColor	1.3.6.1.4.1.1206.4.2.3.1.15.4.1.4	current	read-create
vmsCharacterHeightPixels	1.3.6.1.4.1.1206.4.2.3.2.1	current	read-only
vmsCharacterWidthPixels	1.3.6.1.4.1.1206.4.2.3.2.2	current	read-only
vmsSignHeightPixels	1.3.6.1.4.1.1206.4.2.3.2.3	current	read-only
vmsSignWidthPixels	1.3.6.1.4.1.1206.4.2.3.2.4	current	read-only
vmsHorizontalPitch	1.3.6.1.4.1.1206.4.2.3.2.5	current	read-only
vmsVerticalPitch	1.3.6.1.4.1.1206.4.2.3.2.6	current	read-only
monochromeColor	1.3.6.1.4.1.1206.4.2.3.2.7	current	read-only
numFonts	1.3.6.1.4.1.1206.4.2.3.3.1	current	read-only
fontTable	1.3.6.1.4.1.1206.4.2.3.3.2	current	not-accessible
fontEntry	1.3.6.1.4.1.1206.4.2.3.3.2.1	current	not-accessible
fontIndex	1.3.6.1.4.1.1206.4.2.3.3.2.1.1	current	read-only
fontNumber	1.3.6.1.4.1.1206.4.2.3.3.2.1.2	current	read-write
fontName	1.3.6.1.4.1.1206.4.2.3.3.2.1.3	current	read-write
fontHeight	1.3.6.1.4.1.1206.4.2.3.3.2.1.4	current	read-write
fontCharSpacing	1.3.6.1.4.1.1206.4.2.3.3.2.1.5	current	read-write
fontLineSpacing	1.3.6.1.4.1.1206.4.2.3.3.2.1.6	current	read-write
fontVersionID	1.3.6.1.4.1.1206.4.2.3.3.2.1.7	current	read-only
fontStatus	1.3.6.1.4.1.1206.4.2.3.3.2.1.8	current	read-write

maxFontCharacters	1.3.6.1.4.1.1206.4.2.3.3.3	current	read-only
characterTable	1.3.6.1.4.1.1206.4.2.3.3.4	current	not-accessible
characterEntry	1.3.6.1.4.1.1206.4.2.3.3.4.1	current	not-accessible
characterNumber	1.3.6.1.4.1.1206.4.2.3.3.4.1.1	current	read-only
characterWidth	1.3.6.1.4.1.1206.4.2.3.3.4.1.2	current	read-write
characterBitmap	1.3.6.1.4.1.1206.4.2.3.3.4.1.3	current	read-write
fontMaxCharacterSize	1.3.6.1.4.1.1206.4.2.3.3.5	current	read-only
defaultBackgroundColor	1.3.6.1.4.1.1206.4.2.3.4.1	deprecated	read-write
defaultForegroundColor	1.3.6.1.4.1.1206.4.2.3.4.2	deprecated	read-write
defaultFlashOn	1.3.6.1.4.1.1206.4.2.3.4.3	current	read-write
defaultFlashOnActivate	1.3.6.1.4.1.1206.4.2.3.4.17	current	read-only
defaultFlashOff	1.3.6.1.4.1.1206.4.2.3.4.4	current	read-write
defaultFlashOffActivate	1.3.6.1.4.1.1206.4.2.3.4.18	current	read-only
defaultFont	1.3.6.1.4.1.1206.4.2.3.4.5	current	read-write
defaultFontActivate	1.3.6.1.4.1.1206.4.2.3.4.19	current	read-only
defaultJustificationLine	1.3.6.1.4.1.1206.4.2.3.4.6	current	read-write
defaultJustificationLineActivate	1.3.6.1.4.1.1206.4.2.3.4.20	current	read-only
defaultJustificationPage	1.3.6.1.4.1.1206.4.2.3.4.7	current	read-write
defaultJustificationPageActivate	1.3.6.1.4.1.1206.4.2.3.4.21	current	read-only
defaultPageOnTime	1.3.6.1.4.1.1206.4.2.3.4.8	current	read-write
defaultPageOnTimeActivate	1.3.6.1.4.1.1206.4.2.3.4.22	current	read-only
defaultPageOffTime	1.3.6.1.4.1.1206.4.2.3.4.9	current	read-write
defaultPageOffTimeActivate	1.3.6.1.4.1.1206.4.2.3.4.23	current	read-only
defaultBackgroundRGB	1.3.6.1.4.1.1206.4.2.3.4.12	current	read-write
defaultBackgroundRGBActivate	1.3.6.1.4.1.1206.4.2.3.4.24	current	read-only
defaultForegroundRGB	1.3.6.1.4.1.1206.4.2.3.4.13	current	read-write
defaultForegroundRGBActivate	1.3.6.1.4.1.1206.4.2.3.4.25	current	read-only
defaultCharacterSet	1.3.6.1.4.1.1206.4.2.3.4.10	current	read-write
dmsColorScheme	1.3.6.1.4.1.1206.4.2.3.4.11	current	read-only
dmsSupportedMultiTags	1.3.6.1.4.1.1206.4.2.3.4.14	deprecated	read-only
dmsMaxNumberPages	1.3.6.1.4.1.1206.4.2.3.4.15	current	read-only
dmsMaxMultiStringLength	1.3.6.1.4.1.1206.4.2.3.4.16	current	read-only
dmsSupportedMultiTags	1.3.6.1.4.1.1206.4.2.3.4.17	current	read-only
dmsNumPermanentMsg	1.3.6.1.4.1.1206.4.2.3.5.1	current	read-only
dmsNumChangeableMsg	1.3.6.1.4.1.1206.4.2.3.5.2	current	read-only
dmsMaxChangeableMsg	1.3.6.1.4.1.1206.4.2.3.5.3	current	read-only
dmsFreeChangeableMemory	1.3.6.1.4.1.1206.4.2.3.5.4	current	read-only

dmsNumVolatileMsg	1.3.6.1.4.1.1206.4.2.3.5.5	current	read-only
dmsMaxVolatileMsg	1.3.6.1.4.1.1206.4.2.3.5.6	current	read-only
dmsFreeVolatileMemory	1.3.6.1.4.1.1206.4.2.3.5.7	current	read-only
dmsMessageTable	1.3.6.1.4.1.1206.4.2.3.5.8	current	not-accessible
dmsMessageEntry	1.3.6.1.4.1.1206.4.2.3.5.8.1	current	not-accessible
dmsMessageMemoryType	1.3.6.1.4.1.1206.4.2.3.5.8.1.1	current	read-only
dmsMessageNumber	1.3.6.1.4.1.1206.4.2.3.5.8.1.2	current	read-only
dmsMessageMultiString	1.3.6.1.4.1.1206.4.2.3.5.8.1.3	current	read-write
dmsMessageOwner	1.3.6.1.4.1.1206.4.2.3.5.8.1.4	current	read-write
dmsMessageCRC	1.3.6.1.4.1.1206.4.2.3.5.8.1.5	current	read-only
dmsMessageBeacon	1.3.6.1.4.1.1206.4.2.3.5.8.1.6	current	read-write
dmsMessagePixelService	1.3.6.1.4.1.1206.4.2.3.5.8.1.7	current	read-write
dmsMessageRunTimePriority	1.3.6.1.4.1.1206.4.2.3.5.8.1.8	current	read-write
dmsMessageStatus	1.3.6.1.4.1.1206.4.2.3.5.8.1.9	current	read-write
dmsValidateMessageError	1.3.6.1.4.1.1206.4.2.3.5.9	current	read-only
dmsControlMode	1.3.6.1.4.1.1206.4.2.3.6.1	current	read-write
dmsSWReset	1.3.6.1.4.1.1206.4.2.3.6.2	deprecated	read-write
dmsActivateMessage	1.3.6.1.4.1.1206.4.2.3.6.3	deprecated	read-write
dmsMessageTimeRemaining	1.3.6.1.4.1.1206.4.2.3.6.4	current	read-write
dmsMsgTableSource	1.3.6.1.4.1.1206.4.2.3.6.5	current	read-only
dmsMsgRequesterID	1.3.6.1.4.1.1206.4.2.3.6.6	deprecated	read-only
dmsMsgSourceMode	1.3.6.1.4.1.1206.4.2.3.6.7	current	read-only
dmsShortPowerRecoveryMessage	1.3.6.1.4.1.1206.4.2.3.6.8	current	read-write
dmsLongPowerRecoveryMessage	1.3.6.1.4.1.1206.4.2.3.6.9	current	read-write
dmsShortPowerLossTime	1.3.6.1.4.1.1206.4.2.3.6.10	current	read-write
dmsResetMessage	1.3.6.1.4.1.1206.4.2.3.6.11	current	read-write
dmsCommunicationsLossMessage	1.3.6.1.4.1.1206.4.2.3.6.12	current	read-write
dmsTimeCommLoss	1.3.6.1.4.1.1206.4.2.3.6.13	current	read-write
dmsPowerLossMessage	1.3.6.1.4.1.1206.4.2.3.6.14	current	read-write
dmsEndDurationMessage	1.3.6.1.4.1.1206.4.2.3.6.15	current	read-write
dmsMemoryMgmt	1.3.6.1.4.1.1206.4.2.3.6.16	current	read-write
dmsActivateMsgError	1.3.6.1.4.1.1206.4.2.3.6.17	current	read-only
dmsMultiSyntaxError	1.3.6.1.4.1.1206.4.2.3.6.18	current	read-only
dmsMultiSyntaxErrorPosition	1.3.6.1.4.1.1206.4.2.3.6.19	current	read-only

dmsMultiOtherErrorDescription	1.3.6.1.4.1.1206.4.2.3.6.20	current	read-only
vmsPixelServiceDuration	1.3.6.1.4.1.1206.4.2.3.6.21	deprecated	read-write
vmsPixelServiceFrequency	1.3.6.1.4.1.1206.4.2.3.6.22	current	read-write
vmsPixelServiceTime	1.3.6.1.4.1.1206.4.2.3.6.23	current	read-write
dmsActivateErrorMsgCode	1.3.6.1.4.1.1206.4.2.3.6.24	deprecated	read-only
dmsActivateMessageState	1.3.6.1.4.1.1206.4.2.3.6.25	current	read-only
dmsActivateMessageV4	1.3.6.1.4.1.1206.4.2.3.6.26	current	read-write
dmsActivateErrorMsgCodeV4	1.3.6.1.4.1.1206.4.2.3.6.27	current	read-only
dmsMsgRequesterAddressType	1.3.6.1.4.1.1206.4.2.3.6.28	current	read-only
dmsMsgRequesterAddress	1.3.6.1.4.1.1206.4.2.3.6.29	current	read-only
dmsMsgRequesterSecurityName	1.3.6.1.4.1.1206.4.2.3.6.30	current	read-only
vmsPixelServiceDurationV4	1.3.6.1.4.1.1206.4.2.3.6.31	current	read-write
dmsIllumControl	1.3.6.1.4.1.1206.4.2.3.7.1	current	read-write
dmsIllumMaxPhotocellLevel	1.3.6.1.4.1.1206.4.2.3.7.2	deprecated	read-only
dmsIllumPhotocellLevelStatus	1.3.6.1.4.1.1206.4.2.3.7.3	deprecated	read-only
dmsIllumNumBrightLevels	1.3.6.1.4.1.1206.4.2.3.7.4	current	read-only
dmsIllumBrightLevelStatus	1.3.6.1.4.1.1206.4.2.3.7.5	current	read-only
dmsIllumManLevel	1.3.6.1.4.1.1206.4.2.3.7.6	current	read-write
dmsIllumBrightnessValues	1.3.6.1.4.1.1206.4.2.3.7.7	current	read-write
dmsIllumBrightnessValuesError	1.3.6.1.4.1.1206.4.2.3.7.8	current	read-only
dmsIllumLightOutputStatus	1.3.6.1.4.1.1206.4.2.3.7.9	current	read-only
numActionTableEntries	1.3.6.1.4.1.1206.4.2.3.8.1	current	read-only
dmsActionTable	1.3.6.1.4.1.1206.4.2.3.8.2	current	not-accessible
dmsActionEntry	1.3.6.1.4.1.1206.4.2.3.8.2.1	current	not-accessible
dmsActionIndex	1.3.6.1.4.1.1206.4.2.3.8.2.1.1	current	read-only
dmsActionMsgCode	1.3.6.1.4.1.1206.4.2.3.8.2.1.2	current	read-write
statMultiFieldRows	1.3.6.1.4.1.1206.4.2.3.9.1	current	read-only
statMultiFieldTable	1.3.6.1.4.1.1206.4.2.3.9.2	current	not-accessible
statMultiFieldEntry	1.3.6.1.4.1.1206.4.2.3.9.2.1	current	not-accessible
statMultiFieldIndex	1.3.6.1.4.1.1206.4.2.3.9.2.1.1	current	read-only
statMultiFieldCode	1.3.6.1.4.1.1206.4.2.3.9.2.1.2	current	read-only
statMultiCurrentFieldValue	1.3.6.1.4.1.1206.4.2.3.9.2.1.3	current	read-only
dmsCurrentSpeed	1.3.6.1.4.1.1206.4.2.3.9.3	deprecated	read-only
dmsCurrentSpeedLimit	1.3.6.1.4.1.1206.4.2.3.9.4	current	read-write
watchdogFailureCount	1.3.6.1.4.1.1206.4.2.3.9.5	deprecated	read-only

dmsStatDoorOpen	1.3.6.1.4.1.1206.4.2.3.9.6	deprecated	read-only
shortErrorStatus	1.3.6.1.4.1.1206.4.2.3.9.7.1	current	read-only
controllerErrorStatus	1.3.6.1.4.1.1206.4.2.3.9.7.10	deprecated	read-only
dmsOtherShortErrorDescr	1.3.6.1.4.1.1206.4.2.3.9.7.1	current	read-only
dmsPowerFailureStatusMap	1.3.6.1.4.1.1206.4.2.3.9.7.11	deprecated	read-only
dmsPowerNumRows	1.3.6.1.4.1.1206.4.2.3.9.7.12	deprecated	read-only
dmsPowerStatusTable	1.3.6.1.4.1.1206.4.2.3.9.7.13	deprecated	not-accessible
dmsPowerStatusEntry	1.3.6.1.4.1.1206.4.2.3.9.7.13.1	deprecated	not-accessible
dmsPowerIndex	1.3.6.1.4.1.1206.4.2.3.9.7.13.1.1	deprecated	read-only
dmsPowerDescription	1.3.6.1.4.1.1206.4.2.3.9.7.13.1.2	deprecated	read-only
dmsPowerMfrStatus	1.3.6.1.4.1.1206.4.2.3.9.7.13.1.3	deprecated	read-only
dmsPowerStatus	1.3.6.1.4.1.1206.4.2.3.9.7.13.1.4	deprecated	read-only
dmsPowerVoltage	1.3.6.1.4.1.1206.4.2.3.9.7.13.1.5	deprecated	read-only
dmsPowerType	1.3.6.1.4.1.1206.4.2.3.9.7.13.1.6	deprecated	read-only
fanFailures	1.3.6.1.4.1.1206.4.2.3.9.7.8	deprecated	read-only
fanTestActivation	1.3.6.1.4.1.1206.4.2.3.9.7.9	deprecated	read-write
dmsClimateCtrlStatusMap	1.3.6.1.4.1.1206.4.2.3.9.7.14	deprecated	read-only
dmsClimateCtrlNumRows	1.3.6.1.4.1.1206.4.2.3.9.7.16	deprecated	read-only
dmsClimateCtrlStatusTable	1.3.6.1.4.1.1206.4.2.3.9.7.17	deprecated	not-accessible
dmsClimateCtrlStatusEntry	1.3.6.1.4.1.1206.4.2.3.9.7.17.1	deprecated	not-accessible
dmsClimateCtrlIndex	1.3.6.1.4.1.1206.4.2.3.9.7.17.1.1	deprecated	read-only
dmsClimateCtrlDescription	1.3.6.1.4.1.1206.4.2.3.9.7.17.1.2	deprecated	read-only
dmsClimateCtrlMfrStatus	1.3.6.1.4.1.1206.4.2.3.9.7.17.1.3	deprecated	read-only
dmsClimateCtrlErrorStatus	1.3.6.1.4.1.1206.4.2.3.9.7.17.1.4	deprecated	read-only
dmsClimateCtrlOnStatus	1.3.6.1.4.1.1206.4.2.3.9.7.17.1.5	deprecated	read-only
dmsClimateCtrlTestActivation	1.3.6.1.4.1.1206.4.2.3.9.7.17.1.6	deprecated	read-write
dmsClimateCtrlAbortReason	1.3.6.1.4.1.1206.4.2.3.9.7.17.1.7	deprecated	read-only
dmsClimateCtrlType	1.3.6.1.4.1.1206.4.2.3.9.7.17.1.8	deprecated	read-only
pixelFailureTableNumRows	1.3.6.1.4.1.1206.4.2.3.9.7.2	current	read-only

pixelFailureTable	1.3.6.1.4.1.1206.4.2.3.9.7.3	current	not-accessible
pixelFailureEntry	1.3.6.1.4.1.1206.4.2.3.9.7.3.1	current	not-accessible
pixelFailureDetectionType	1.3.6.1.4.1.1206.4.2.3.9.7.3.1.1	current	read-only
pixelFailureIndex	1.3.6.1.4.1.1206.4.2.3.9.7.3.1.2	current	read-only
pixelFailureXLocation	1.3.6.1.4.1.1206.4.2.3.9.7.3.1.3	current	read-only
pixelFailureYLocation	1.3.6.1.4.1.1206.4.2.3.9.7.3.1.4	current	read-only
pixelFailureStatus	1.3.6.1.4.1.1206.4.2.3.9.7.3.1.5	current	read-only
pixelTestActivation	1.3.6.1.4.1.1206.4.2.3.9.7.4	current	read-write
pixelStatusTable	1.3.6.1.4.1.1206.4.2.3.9.7.18	current	not-accessible
pixelStatusEntry	1.3.6.1.4.1.1206.4.2.3.9.7.18.1	current	not-accessible
dmsPixelStatusIndex	1.3.6.1.4.1.1206.4.2.3.9.7.18.1.1	current	read-only
dmsPixelStatus	1.3.6.1.4.1.1206.4.2.3.9.7.18.1.2	deprecated	read-only
dmsPixelStatusV4	1.3.6.1.4.1.1206.4.2.3.9.7.18.1.3	current	read-only
dmsPixelFailureTestRows	1.3.6.1.4.1.1206.4.2.3.9.7.19	current	read-only
dmsPixelFailureMessageRows	1.3.6.1.4.1.1206.4.2.3.9.7.20	current	read-only
lampFailureStuckOn	1.3.6.1.4.1.1206.4.2.3.9.7.5	deprecated	read-only
lampFailureStuckOff	1.3.6.1.4.1.1206.4.2.3.9.7.6	deprecated	read-only
lampTestActivation	1.3.6.1.4.1.1206.4.2.3.9.7.7	deprecated	read-write
dmsLampNumRows	1.3.6.1.4.1.1206.4.2.3.9.7.23	deprecated	read-only
dmsLampStatusTable	1.3.6.1.4.1.1206.4.2.3.9.7.24	deprecated	not-accessible
dmsLampStatusEntry	1.3.6.1.4.1.1206.4.2.3.9.7.24.1	deprecated	not-accessible
dmsLampIndex	1.3.6.1.4.1.1206.4.2.3.9.7.24.1.1	deprecated	read-only
dmsLampDescription	1.3.6.1.4.1.1206.4.2.3.9.7.24.1.2	deprecated	read-only
dmsLampMfrStatus	1.3.6.1.4.1.1206.4.2.3.9.7.24.1.3	deprecated	read-only
dmsLampStatus	1.3.6.1.4.1.1206.4.2.3.9.7.24.1.4	deprecated	read-only
dmsLampPixelTop	1.3.6.1.4.1.1206.4.2.3.9.7.24.1.5	deprecated	read-only
dmsLampPixelLeft	1.3.6.1.4.1.1206.4.2.3.9.7.24.1.6	deprecated	read-only
dmsLampPixelBottom	1.3.6.1.4.1.1206.4.2.3.9.7.24.1.7	deprecated	read-only
dmsLampPixelRight	1.3.6.1.4.1.1206.4.2.3.9.7.2	deprecated	read-only

	4.1.8		
dmsDrumStatusMap	1.3.6.1.4.1.1206.4.2.3.9.7.25	deprecated	read-only
dmsDrumNumRows	1.3.6.1.4.1.1206.4.2.3.9.7.26	deprecated	read-only
dmsDrumStatusTable	1.3.6.1.4.1.1206.4.2.3.9.7.27	deprecated	not-accessible
dmsDrumStatusEntry	1.3.6.1.4.1.1206.4.2.3.9.7.27.1	deprecated	not-accessible
dmsDrumIndex	1.3.6.1.4.1.1206.4.2.3.9.7.27.1.1	deprecated	read-only
dmsDrumDescription	1.3.6.1.4.1.1206.4.2.3.9.7.27.1.2	deprecated	read-only
dmsDrumMfrStatus	1.3.6.1.4.1.1206.4.2.3.9.7.27.1.3	deprecated	read-only
dmsDrumStatus	1.3.6.1.4.1.1206.4.2.3.9.7.27.1.4	deprecated	read-only
dmsLightSensorStatusMap	1.3.6.1.4.1.1206.4.2.3.9.7.28	deprecated	read-only
dmsLightSensorNumRows	1.3.6.1.4.1.1206.4.2.3.9.7.29	deprecated	read-only
dmsLightSensorStatusTable	1.3.6.1.4.1.1206.4.2.3.9.7.30	deprecated	not-accessible
dmsLightSensorStatusEntry	1.3.6.1.4.1.1206.4.2.3.9.7.30.1	deprecated	not-accessible
dmsLightSensorIndex	1.3.6.1.4.1.1206.4.2.3.9.7.30.1.1	deprecated	read-only
dmsLightSensorDescription	1.3.6.1.4.1.1206.4.2.3.9.7.30.1.2	deprecated	read-only
dmsLightSensorCurrentReading	1.3.6.1.4.1.1206.4.2.3.9.7.30.1.3	deprecated	read-only
dmsLightSensorStatus	1.3.6.1.4.1.1206.4.2.3.9.7.30.1.4	deprecated	read-only
dmsHumiditySensorStatusMap	1.3.6.1.4.1.1206.4.2.3.9.7.31	deprecated	read-only
dmsHumiditySensorNumRows	1.3.6.1.4.1.1206.4.2.3.9.7.32	deprecated	read-only
dmsHumiditySensorStatusTable	1.3.6.1.4.1.1206.4.2.3.9.7.33	deprecated	not-accessible
dmsHumiditySensorStatusEntry	1.3.6.1.4.1.1206.4.2.3.9.7.33.1	deprecated	not-accessible
dmsHumiditySensorIndex	1.3.6.1.4.1.1206.4.2.3.9.7.33.1.1	deprecated	read-only
dmsHumiditySensorDescription	1.3.6.1.4.1.1206.4.2.3.9.7.33.1.2	deprecated	read-only
dmsHumiditySensorCurrentReading	1.3.6.1.4.1.1206.4.2.3.9.7.33.1.3	deprecated	read-only
dmsHumiditySensorStatus	1.3.6.1.4.1.1206.4.2.3.9.7.33.1.4	deprecated	read-only
dmsTempSensorStatusMap	1.3.6.1.4.1.1206.4.2.3.9.7.34	deprecated	read-only
dmsTempSensorNumRows	1.3.6.1.4.1.1206.4.2.3.9.7.35	deprecated	read-only
dmsTempSensorStatusTable	1.3.6.1.4.1.1206.4.2.3.9.7.3	deprecated	not-

	6		accessible
dmsTempSensorStatusEntry	1.3.6.1.4.1.1206.4.2.3.9.7.36.1	deprecated	not-accessible
dmsTempSensorIndex	1.3.6.1.4.1.1206.4.2.3.9.7.36.1.1	deprecated	read-only
dmsTempSensorDescription	1.3.6.1.4.1.1206.4.2.3.9.7.36.1.2	deprecated	read-only
dmsTempSensorCurrentReading	1.3.6.1.4.1.1206.4.2.3.9.7.36.1.3	deprecated	read-only
dmsTempSensorHighWarningTemperature	1.3.6.1.4.1.1206.4.2.3.9.7.36.1.4	deprecated	read-only
dmsTempSensorLowWarningTemperature	1.3.6.1.4.1.1206.4.2.3.9.7.36.1.5	deprecated	read-only
dmsTempSensorHighCriticalTemperature	1.3.6.1.4.1.1206.4.2.3.9.7.36.1.6	deprecated	read-only
dmsTempSensorLowCriticalTemperature	1.3.6.1.4.1.1206.4.2.3.9.7.36.1.7	deprecated	read-only
dmsTempSensorStatus	1.3.6.1.4.1.1206.4.2.3.9.7.36.1.8	deprecated	read-only
dmsTempSensorHighestCriticalTempThreshold	1.3.6.1.4.1.1206.4.2.3.9.7.37	deprecated	read-only
dmsTempSensorLowestCriticalTempThreshold	1.3.6.1.4.1.1206.4.2.3.9.7.38	deprecated	read-only
signVolts	1.3.6.1.4.1.1206.4.2.3.9.8.1	deprecated	read-only
lowFuelThreshold	1.3.6.1.4.1.1206.4.2.3.9.8.2	deprecated	read-write
fuelLevel	1.3.6.1.4.1.1206.4.2.3.9.8.3	deprecated	read-only
engineRPM	1.3.6.1.4.1.1206.4.2.3.9.8.4	deprecated	read-only
lineVolts	1.3.6.1.4.1.1206.4.2.3.9.8.5	deprecated	read-only
powerSource	1.3.6.1.4.1.1206.4.2.3.9.8.6	deprecated	read-only
tempMinCtrlCabinet	1.3.6.1.4.1.1206.4.2.3.9.9.1	deprecated	read-only
tempMaxCtrlCabinet	1.3.6.1.4.1.1206.4.2.3.9.9.2	deprecated	read-only
tempMinAmbient	1.3.6.1.4.1.1206.4.2.3.9.9.3	deprecated	read-only
tempMaxAmbient	1.3.6.1.4.1.1206.4.2.3.9.9.4	deprecated	read-only
tempMinSignHousing	1.3.6.1.4.1.1206.4.2.3.9.9.5	deprecated	read-only
tempMaxSignHousing	1.3.6.1.4.1.1206.4.2.3.9.9.6	deprecated	read-only
tempSensorWarningMap	1.3.6.1.4.1.1206.4.2.3.9.9.7	deprecated	read-only
tempSensorCriticalTempMap	1.3.6.1.4.1.1206.4.2.3.9.9.8	deprecated	read-only
dmsSrsaPixelsTable	1.3.6.1.4.1.1206.4.2.3.9.10.1	current	not-accessible
dmsSrsaPixelsEntry	1.3.6.1.4.1.1206.4.2.3.9.10.1.1	current	not-accessible
dmsSrsaPixelsTop	1.3.6.1.4.1.1206.4.2.3.9.10.1.1.1	current	read-only
dmsSrsaPixelsLeft	1.3.6.1.4.1.1206.4.2.3.9.10.1.1.2	current	read-only
dmsSrsaPixelsBottom	1.3.6.1.4.1.1206.4.2.3.9.10.1.1.3	current	read-only
dmsSrsaPixelsRight	1.3.6.1.4.1.1206.4.2.3.9.10.1.1.4	current	read-only

dmsGraphicMaxEntries	1.3.6.1.4.1.1206.4.2.3.10.1	current	read-only
dmsGraphicNumEntries	1.3.6.1.4.1.1206.4.2.3.10.2	current	read-only
dmsGraphicMaxSize	1.3.6.1.4.1.1206.4.2.3.10.3	current	read-only
availableGraphicMemory	1.3.6.1.4.1.1206.4.2.3.10.4	deprecated	read-only
dmsGraphicBlockSize	1.3.6.1.4.1.1206.4.2.3.10.5	current	read-only
dmsGraphicTable	1.3.6.1.4.1.1206.4.2.3.10.6	current	not-accessible
dmsGraphicEntry	1.3.6.1.4.1.1206.4.2.3.10.6.1	current	not-accessible
dmsGraphicIndex	1.3.6.1.4.1.1206.4.2.3.10.6.1.1	current	read-only
dmsGraphicNumber	1.3.6.1.4.1.1206.4.2.3.10.6.1.2	current	read-write
dmsGraphicName	1.3.6.1.4.1.1206.4.2.3.10.6.1.3	current	read-write
dmsGraphicHeight	1.3.6.1.4.1.1206.4.2.3.10.6.1.4	current	read-write
dmsGraphicWidth	1.3.6.1.4.1.1206.4.2.3.10.6.1.5	current	read-write
dmsGraphicType	1.3.6.1.4.1.1206.4.2.3.10.6.1.6	current	read-write
dmsGraphicID	1.3.6.1.4.1.1206.4.2.3.10.6.1.7	current	read-only
dmsGraphicTransparentEnabled	1.3.6.1.4.1.1206.4.2.3.10.6.1.8	current	read-write
dmsGraphicTransparentColor	1.3.6.1.4.1.1206.4.2.3.10.6.1.9	current	read-write
dmsGraphicStatus	1.3.6.1.4.1.1206.4.2.3.10.6.1.10	current	read-write
dmsGraphicBitmapTable	1.3.6.1.4.1.1206.4.2.3.10.7	current	not-accessible
dmsGraphicBitmapEntry	1.3.6.1.4.1.1206.4.2.3.10.7.1.1	current	not-accessible
dmsGraphicBitmapIndex	1.3.6.1.4.1.1206.4.2.3.10.7.1.1	current	read-only
dmsGraphicBlockNumber	1.3.6.1.4.1.1206.4.2.3.10.7.1.2	current	read-only
dmsGraphicBlockBitmap	1.3.6.1.4.1.1206.4.2.3.10.7.1.3	current	read-write
dmsAvailableGraphicMemoryV4	1.3.6.1.4.1.1206.4.2.3.10.8	current	read-only
dmsNumCallableMessages	1.3.6.1.4.1.1206.4.2.3.11.1	current	read-only
dmsCallableMessageTable	1.3.6.1.4.1.1206.4.2.3.11.2	current	not-accessible
dmsCallableMessageEntry	1.3.6.1.4.1.1206.4.2.3.11.2.1	current	not-accessible
dmsCallableMessageIndex	1.3.6.1.4.1.1206.4.2.3.11.2.1.1	current	not-accessible
dmsCallableMessageCode	1.3.6.1.4.1.1206.4.2.3.11.2.1.2	current	read-create
dmsCallableMessageStorageType	1.3.6.1.4.1.1206.4.2.3.11.2.1.3	current	read-create

dmsCallableMessageRowStatus	1.3.6.1.4.1.1206.4.2.3.11.2.1.4	current	read-create

Annex C
Test Procedures [Informative]

The current test procedures for this document can be found online at https://github.com/ite-org/NTCIP1203.

Annex D
Documentation of Revisions [Informative]

Annex D identifies the changes that have been made to NTCIP 1203 v03 to develop NTCIP 1203 v04.

The primary purpose of NTCIP 1203 v04 is to improve the security of the interface, which includes migration to SNMPv3 from SNMPv1 and to ISO 26048-1 from NTCIP 1201.. Several other changes and corrections were also made. Annex D identifies the changes between NTCIP 1203 v03 and NTCIP 1203 v04, and why each of these changes have been made.

D.1 Security enhancements

D.1.1 Support for ISO 15784-2 and NTCIP 2301 v03

Previous versions of this document were designed to use NTCIP 2301 v02, which was based in SNMPv1 and similar protocols with no significant security with an expectation that it woud be deployed over UDP/IP. This version is updated to use NTCIP 2301 v03, which references ISO 15784-2 and is based on SNMPv3 over TLS and uses the TLS provided security certificate as teh authentication mechanism for teh user to have access control to SNMP data.

D.1.2 Update design to use ISO 26048-1

As a part of the security update, some of the data structures needed to be revised to allow for proper access control; this was particularly true for the global objects defined in NTCIP 1201. For example, the logs and notifications defined in NTCIP 1201 were designed such that any user with access to the log or notifiction feature would have access to the any data within the device, including access to other rows in the event log and notification tables, even if they were created by someone else. Other problems existed with the NTCIP 1201 data as well, such as the use of unsigned 4-octet integers and a representation of time that is nearing a rollover and does not support millseconds.

In addition, this major version change provided an opportunity to make other revisions to better align NTCIP standards with more useful IETF standards. For example, the NTCIP module table is replaced with the system object resource objects (sysOR), which form an integral part of SNMPv3, and entity objects (RFC 6933).

The end result is that virtually all objects defined in NTCIP 1201 and NTCIP 1103 needed to be updated. Rather than redefining all of this data, it was deemed to be more efficient to shift the reference to ISO 26048-1, which mimics the original NTCIP designs but addresses all of the issues related to SNMPv3.

D.1.3 More secure and robust requester identifier

Prior versions of this document defined dmsMsgRequesterID, which provided the IPv4 address as provided within the Activation code (or the loopback address, if internally activated). This mechanism allowed an IPv4 address to be reported, even if the request was made NTCIP 2201 (i.e., a nearly empty transport profile that did not indicate an IP address). The new design populates the requester information based on the address contained within the network layer of the incoming request and stores the information in a pair of objects, which allows the address to be IPv4 or IPv6 (or any of a number of other potential protocols).

D.1.4 Definition of conformance and consistency

Version 04 introduces the distinction between conformant devices, which do not support any protocol that is not secure, and consistent devices, which support protocols that are not secure as long as they can be disabled.

It is envisioned that for some transition period, agencies will need to be able to procure consistent devices (i.e., they will need to be integrated with a management station that does not support SNMPv3 yet). However, as soon as all such management stations are replaced, the devices should be upgraded to be conformant.

D.2 Changes due to migration to SNMPv3

D.2.1 Update the MIB to use SMIv2

SNMPv3 MIBs use the SMIv2 format. This format is similar to what was used with SNMPv1, with the following notable differences:

a) The IMPORTS statement was revised to reference updated RFCs (e.g., RFC 2578)
b) ACCESS is changed to MAX-ACCESS. This reflects the fact that in some situations instances of an object-type might not support the full access. For example, a permanent row within a table can restrict write access to an object instance that would otherwise be writable.
c) STATUS is changed to reflect whether the object is considered appropriate for current design or not rather than coupling this information for conformance information. The conformance information is handled separately in SNMPv3 through conformance groups and compliance statements.
d) The allowed SYNTAX values have been revised to be more precise and to strictly enforce SNMPv1 guidance that was sometime overlooked by NTCIP. For example, INTEGER has been replaced with Integer32 (or a textual convention derived from this syntax), when allowed. However, when the prior version of this document defined an INTEGER with a range that included the top bit of a 4-octet integer (e.g., INTEGER (0.. 4294967295)), the syntax was replaced with Unsigned32 (or a textual convention derived from this syntax).
e) Units information, previously defined within the DESCRIPTION field has moved to the SMIv2 UNITS field

D.2.2 MIB distribution permissions

With SMIv2, implementations should indicate their capabilities by defining a separate AGENT-CAPABILITIES statement rather than creating a modified MIB. The MIB distribution permissions jhave been revised to accommodate this changed approach.

D.2.3 Updated error codes

Previous versions of this document refered to specific SNMPv1 error codes. Most of these codes are deprecated in SNMPv3 and the references have been updated to refer to teh more meaningful SNMPv3 error codes.

D.3 Revisions to use more generic designs

D.3.1 Revised diagnostics to use supplemental roadside sensors and acturators

Version 04 revised the design of several diagnostic capabilities to use the supplemental roadside sensors and actuators feature. The diagnostic features include:

- door status,
- temperature readings,
- humidity readings,
- light sensor status,
- speed detector data,
- power information (e.g., line volts, power supplies, battery charge, fuel level),
- lamp status, and
- drum status.

D.3.2 Object group definitions

Object groups have been defined at the end of the MIB in Section 5. These allow a device to be accompanied with an agent capabilities statement that identifies which objects groups are supported. A management station can then query the field device to determine its sysORID and identify its capability statement and then determine the full capabilities of the device in a manner that is much more detailed than previously provided by the NTCIP 1201 module table.

D.3.3 Support for dmsMessagePixelService and dmsMessageBeacon

Rather than allowing implementations to optionally support dmsMessagePixelService and dmsMessageBeacon, support for these objects is now required, although their supported ranges can be constrained to zero (i.e., indicating that the referenced feature is not supported). This allows a more streamlined standardized dialog while still allowing optional support for the underlying feature.

D.3.4 Replaced Annex G and H with references to ISO 26048-1

In v03, Annex G was used to define generic dialogs and Annex H was used to define user needs, rquirements, and dialogs that would eventually be migrated to NTCIP 2301. This version replaces this information with references directly to ISO 26048-1.

D.3.5 Octet encoding rules

The reference to octet encoding rules (OER) was changed from NTCIP 1102 to ITU-T X.696. The latter is entirelly consistent with the former (and was derived from it) but is recognized as an international standard and is maintained with ASN.1 as it is updated. It is threfore deemed a more appropriate reference and lowers maintenance effort for the NTCIP standards.

D.4 New capabilities

D.4.1 Display messages based on conditional triggers

The trigger logic defined in ISO 26048-1 allows for calling actions, such as displaying messages. As a result, condition-based message display, which had been requested in previous versions of NTCIP 1203 but never designed, has been added to v04.

D.4.2 Added support for Unicode for text-based objects

In v03, text-based objects were defined using DisplayString, which limits the character set to NVT-ASCII. v04 has updated the syntax of these objects to SnmpAdminString, which allows for any UTF-8 character. This revision was made to better support the international community.

Both DisplayString and SnmpAdminString resolve to OCTET STRING (SIZE (0..255)) and all NVT-ASCII characters are encoded the same way as ASCII and UTF-8. As a result, the change to SnmpAdminString was made without deprecating the previously defined object. However, support for characters beyond NVT-ASCII are considered optional and should be required within the PICS when needed.

D.4.3 New utf8Style encoding of MultiString

The MULTI string object was extended to support a new utf8Style encoding format to support better support for messages containing non-ASCII characters.

D.4.4 Display location objects

Version 04 adds support for describing the display location using latitude, longitude, and elevation, as requested by the international community.

D.4.5 Enhanced beacon configuration and control

Version 04 adds enhanced support for describing, configuring, and controlling beacons that are attached to the sign display. Prior versions assumed that beacons had a fixed configuration with a single operational capability (e.g., alternating flash) while the new design allows the flashing sequence to be controlled (e.g., alternating flash for one message and simultaneous flash for another).

D.5 Resolving ambiguities

D.5.1 Pixel status

Prior versions of this document defined dmsPixelStatus as a bit string encoded within an OCTET STRING; however, the mapping of bits within the bitstring was ambiguous. The object was deprecated and replaced with a new object that has an unambigous ordering of bits.

D.5.2 Added version conformance table and object refinement table

Prior versions of the standard indicated that all objects could be subranged, which if taken to its literal extream would imply that a device could be conformant even if no object values could be changed. This version of the document adopts the SNMPv3 approach to requiring implementations to support the full standardized range of an object, unless an object refinement explicitly allows implementations to define a subrange. Even in that case, the standard can define a minimum subrange to be supported. This is achieved through the new object refinement table in Annex A.

In addition, Annex A defines a version conformance table, which sets a baseline of which object groups are required to be supported in each version of the standard, starting with this version.

D.6 Other changes

D.6.1 Pixel service duration

Pixel service object was originally designed for flip-disk type signs with an assumption that service would require at least a second to perform. However, the concept is now used for LED signs, but the service period is typically measured in milliseconds. The previous vmsPixelServiceDuration object was replaced with a new version to accommodate this more precise timing.

D.6.2 Changes to requirement and dialog numbering

As many of the previous user needs, requirements dialogs, and objects no longer apply, many of the clauses from the previous version of this document were no longer needed. This document

a) Retains the heading numbers for user needs, marking those user needs that are no longer needed as "reserved" and providing an explanation as to where they are addressed;
b) Deletes dialogs that are no longer referenced and renumbers all subsequent dialogs;
c) Deletes requirements that are no longer referenced and renumbers all subsequent requirements;
d) Retains the heading numbers for objects and including a "superseded by" or "supplanted by" subclause, where appropriate to guide the reader to the replacement object.

D.6.3 Updated reference to National ITS Arcitecture

This document traces its functionality to the national ITS architecture (2.8), which is now known as the Architecture Reference for Cooperative and Intelligent Transport (ARC-IT). The references to ARC-IT were updated to the most recent version.

D.6.4 Updated DMS reference architecture to use approach consistent with ARC-IT

This document describes the environment for which it is intended with its own reference architecture (2.3). The presentation of this architecture was updated to use the physical view conventions, as defined by ARC-IT.

D.6.5 Revised boilerplate text

The migration to SNMPv3 is affecting the entire NTCIP 12xx series. As a result, the documents are being updated to use more consistent wording throughout their introductory and tutorial sections to improve consistency among standards.

Annex E
Frequently Asked Questions [Informative]

Annex E addresses questions that readers of this document might ask. The intent is to clarify issues in this document that are not easily understood and to point out features that are intentionally not covered by this document.

E.1 Does NTCIP 1203 include a feature to automatically blank a sign (or take other action) in the event that the sign becomes illegible due to pixel errors?

The idea here is to have the sign monitor the number of pixel errors and, through some non-standardized, non-defined, vendor-specific algorithm, determine whether the sign is considered "legible." If illegible, the sign could automatically blank itself or take another action. This is conceptually a good idea, but practically very difficult to effectively implement and therefore *not recommended* by the NTCIP DMS Working Group (WG). The difficulty comes in determining what constitutes illegibility. The WG determined that purely basing illegibility on a percentage or number of failed pixels is not sufficient. For example, consider the following cases:

a) A hundred (100) failed pixels in an unused portion of the sign face and the maximum allowable number of pixels is defined as ninety (90) pixels. The message would still be legible, but the number of failed pixels would have been exceeded. However, another message might not be legible depending on line and/or page justification, characters used, etc.
b) Consider an "8" character that appears as a "3" by failing only four pixels in the left column. A more complicated algorithm may be possible, but the computational requirements would likely exceed typical sign controllers.

From a larger perspective, if the displayed message is important, it is not desirable to arbitrarily blank the sign, when the intent of the message may still be discernable even with a large number of failed pixels.

However, there is a solution to this potential user need, which may not be executed via the interface between the central computer and the sign controller (the content of this document), but instead be determined via the central computer software. Such a feature could be implemented in the central (with no interface to the sign) by examining pixel error information that is already provided by the data defined in this document.

Alternatively, the legibility feature could be implemented in a sign without requiring any interface between central and sign. However, this implementation would necessitate the need for additional (vendor-specific) objects so that the operator at the central computer is alerted, when a sign controller takes a non-operator-recommended action based on the results of a sign controller-internal legibility algorithm.

E.2 Does this document include a feature to automatically dim an LED sign at a defined high temperature in an attempt to reduce internal heat?

The NTCIP DMS WG does not believe that implementing this feature necessarily requires any interaction between the central and the sign; therefore, no additional MIB objects are defined.

Since dimming a sign will reduce the sign's legibility by some degree, there is a concern that arbitrarily doing so—with no guarantee that the sign's temperature will NOT continue to increase to the point of a sign display shutdown due to exceeding the critical temperature—could be counterproductive.

E.3 Does this document include a feature to control multiple physical signs from a single controller?

Each SNMPv3 agent can support multiple contexts, which allow the same controller to support multiple instances of the same MIB. If the controller only supports a single sign display, it would support a single context with a empty name (i.e., ""). If the sign controller supported multiple displays, it can be designed to support multiple contexts, each with an informative name (e.g., "westbound" and "eastbound").

E.4 Does this document include a testing/training mode whereby a central can operate signs without any messages appearing on the face of the sign?

The NTCIP DMS Working Group does not recommend such a mode, which is the reason why the 'simulation' mode as previously defined within the dmsControlMode object was deprecated. The main concern is that with such a mode, the operator could easily forget that a sign was in the testing/training mode and falsely believe s/he was activating a real message on the face of the sign. It was also decided that there was insufficient demand for such a mode to warrant the time required to carefully defining such a mode and how it would impact all other operations and data in the sign.

E.5 Does this document include a feature to control external devices such as HOV lane gates?

This functionality could be implemented through the SRSA object definitions defined in ISO 26048-1. However, great care should be taken in the design of any system that entails safety issues and the SRSA table was not initially envisioned for this type of use. It is left to the agency to ensure appropriate safeguards are applied.

E.6 Wouldn't it be useful to have an object to report back the version of NTCIP 1203 that is implemented in the device (e.g. DMS)?

The NTCIP DMS WG decided to use the approach defined by the IETF for identifying the capabilities of the device through the definition of object groups. A device's AGENT-CAPABILITIES statement will identify which objet groups are supported, which allow a management station to obtain a very detailed list of what the device supports, including which version numbers of the standard are supported.

E.7 Does this document support the control of Lane Use Signals.

This document considers each Lane Use Signal Head as a separate DMS. Each possible indication (Red X, Green ↓, and potentially other indications) would be defined as a message in the dmsMessageTable.

While the NTCIP DMS WG understands that this is not the most efficient implementation and that coordinated activation of subsequent signal heads over a particular lane cannot be managed via this method, the decision was made not to further complicate this document by attempting to address this advanced function. Additionally, the coordinated activation could potentially be addressed by a Central System managing the LUS.

E.8 Why is the range of the "brightness output" in the dmsIllumBrightnessValues table 0..65535 instead of 0..dmsIllumNumBrightLevels?

This question points out one confusing aspect of the illumination objects, and their usage. That is the relationship between 'Brightness Levels', 'Light Output' and the brightness table defined by 'dmsIllumBrightnessValues'. Brightness level refers to the values used by the objects dmsIllumNumBrightLevels, dmsIllumBrightLevelStatus and dmsIllumManLevel. There is no direct correlation between the Brightness Level and the amount of light actually used to illuminate the sign (except for level 0, which indicates no illumination, and that numerically higher Brightness levels do not have a Light Output lower than the previous level). The actual amount of light, or Light Output, used for a given Brightness Level is defined using the dmsIllumBrightnessValues (in the fields labeled lightOutput x) and is reported to the user in the dmsIllumLightOutputStatus object. In addition to defining the light output for a given lightOutput n, the dmsIllumBrightnessValues object also defines the photocell readings used to move up or down to other levels when dmsIllumControl is set to photocell (2). When defining the light output for the different lightOutput values, it is important to note that the range for these values is always 0 (off) to 65536 (Full brightness), and cannot be sub ranged, thus providing for interoperability and interchangeability. The brightness control system in the DMS can map these values to whatever the hardware requires to generate the selected illumination levels. The table that used to exist within the MIB follows.

```
--  0                  1                  2                  3
--  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
-- +-+-+-+-+-+-+-+-+
-- |NumEntries = n |
-- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
-- |      lightOutput 1            |  Photocell-Level-Down point 1   |
-- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
-- |   Photocell-Level-Up point 1  |     lightOutput 2               |
-- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
-- |   Photocell-Level-Down point 2 |   Photocell-Level-Up point 2   |
-- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
--
-- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
-- |   Photocell-Level-Down point n |   Photocell-Level-Up point n   |
-- +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

E.9 What is the correct way to interpolate a brightness table, and why would you do it?

A system may be designed to interpolate a brightness table when the DMS supports more Brightness Levels than NTCIP 1203 supports, or when the dmsIllumBrightnessValues object specifies fewer Brightness Levels than the DMS supports. This may be done for several reasons, such as making changes to the illumination less visible to the viewer. If a DMS supports fewer levels than what the management station attempts to set, interpolation should not occur and dmsBrightnessValuesError should be set to tooManyLevels (5). The actual method to interpolate the data is manufacturer specific. However, if a system is designed to interpolate the Brightness Table defined by the dmsIllumBrightnessValues object, this interpolation should not be discernable by inspecting the DMS Illumination objects. That is to say that the DMS should 'un-interpolate' data before setting values to the status objects, should not modify the value of dmsIllumBrightnessValues, and should select the correct 'interpolated' level when a level is selected manually. The only exception to this is the dmsIllumLightOutputStatus object, which should report the actual Light Output mapped from the hardware to the 0 to 65535 range.

E.10 Does this document address traps?

This document allows support for the notification feature (i.e., sometimes referred to as "traps"), which enables the device to notify one or more management stations of an unusual event per its configuration.

Uncolntrolled, generic traps are not to be used as they can result in flooding the communications network.

E.11 How does this document address inverted fonts?

The intended and correct way of creating an inverted font is to use the color rectangle MULTI tag in conjunction with the color foreground MULTI tag.

Another method is to create another font with zero character spacing and zero line spacing; but this is not the preferred method.

§