

--The following MIB has been developed for use by FDOT. This MIB is based
--on the official NTCIP Data Dictionary MIBs (for Dynamic Message Signs
--and Global object definitions) and their recommended final and draft
--amendments as they are available as of February 28, 2001.
--
-- Standard objects whose functionality was not specified in the FDOT DMS Guide
-- from 1998, or the update from October 2000 / March 2001 were not included
-- unless they were mandatory or complimentary to required functions.

--Author: Joerg 'Nu' Rosenbohm (PB Farradyne Inc)
--Development Date: March 15, 2001
--Version: v01
--Compiled using the NTCIP Exerciser 3.3b

..*****

-- Filename: 1203(DMS).MIB
-- Source: NTCIP 1203:1997
-- NTCIP 1203 - Amendment 1 from March 2001
-- Description: This MIB defines the Dynamic Message Sign Objects
-- MIB Revision History:
-- 01/01/98 Preliminary Release of TS 3.6 DMS-MIB MIB formatted for 80
-- columns and no TABs
-- In 2.4.1.1.1.2.1 >> changed read-write-only to read-only
-- In 2.5.1.1.1.6, 2.5.1.1.1.8, 2.5.1.1.1.9, and 2.5.1.1.1.10 >>
-- changed read write-write to read-write
-- In 2.5.1.1.1.7 >> corrected ACCESS
-- 07/08/98 Added Copyright Notice
-- 03/09/00 Reversed edits for SMIC Compiler
-- Changed filename and copyright years
-- Changed the import of where Display String comes from
-- 08/09/00 Modified header format and wording of copyright and MIB
-- Distribution Notice
-- Changed experimental to nemaExperimental per NTCIP 1101(SMI)
-- Amendment
-- 03/21/01 Deleted several object definitions not applicable to the FDOT
-- requirements.

--
--Copyright 1997-2000 by the American Association of State Highway and
--Transportation Officials (AASHTO), the Institute of Transportation Engineers
--(ITE), and the National Electrical Manufacturers Association (NEMA). All
--intellectual property rights, including, but not limited to, the rights of
--reproduction in whole or in part in any form, translation into other
--languages and display are reserved by the copyright owners under the laws of
--the United States of America, the Universal Copyright Convention, the Berne
--Convention, and the International and Pan American Copyright Conventions.
--Except for the MIB, Do not copy without written permission of either AASHTO,
--ITE, or NEMA.

--
-- Joint NEMA, AASHTO, and ITE
-- NTCIP Management Information Base
-- DISTRIBUTION NOTICE
--

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

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

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

```

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

```

```

DMS-MIB DEFINITIONS ::= BEGIN

```

```

IMPORTS

```

```

    IpAddress, Opaque, Counter, Gauge
        FROM RFC1155-SMI

```

```

-- DisplayString
-- FROM RFC1213-MIB

```

```

OBJECT-TYPE
    FROM RFC-1212

```

```

nemaExperimental
    FROM NEMA_SMI

```

```

OwnerString, devices, DisplayString
    FROM TMIB-II;

```

```

--For the purpose of this section, the following OBJECT IDENTIFIERS are used:
    dms OBJECT IDENTIFIER ::= {devices 3} --Statement 1

```

```

-- Support of Auxiliary objects is not required. Thus, the relevant statement
-- follows in the NTCIP Standard (1203) has been omitted.

```

```

MessageIDCode ::= OCTET STRING (SIZE(5))

```

```

-- 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
-- {
--     msgMemoryType          INTEGER (0..255),
--     messageNumber         INTEGER (0..65535),
--     messageCRC             OCTET STRING (SIZE (2))
-- }

```

```

-- The fields are defined below.

```

```

MessageActivationCode ::= OCTET STRING (SIZE(12))

```

```

-- 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.
-- dmsMsgTimeRemaining 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 dmsMsgRunTimePriority
-- of the currently displayed message, the new message shall be displayed unless
-- errors are detected.
-- messageMemoryType (8 bits) shall indicate the dmsMsgMemoryType of the desired
-- message.
-- messageNumber (16 bits) shall indicate the dmsMsgNumber of the desired
-- message.
-- messageCRC (16 bits) shall indicate the dmsMsgMessageCRC of the desired message.
-- source Address (32 bits) shall indicate the 4-byte IP address of the device
-- which requested the activation.

-- The dmsActivateMsgError object shall be used to indicate the success or failure
-- of activating any message requested by an object of MessageActivationCode SYNTAX.

-- Three special conditions exist with the MessageActivationCode and MessageIDCode
-- structures. The first condition is related to blanking the sign. A blank sign is
-- activated by setting the MsgMemoryType to 'blank' and the MessageNumber to the
-- desired run time priority. Note that these are actual entries into the message
-- table, but there are only 255 blank messages (because there are only 255 priority
-- levels) and therefore the high-order byte of the MessageNumber field shall always
-- be 0x00. Further, to minimize errors in attempting to blank the sign, the CRC
-- for all blank messages shall be 0x0000.

-- The second condition is related to operating the scheduler. The scheduler is
-- activated in a fashion similar to other messages. The dmsMessageMemoryType
-- is set to 'schedule' (value of 6), the MessageNumber is set to '1', and the
-- MessageCRC is set to 0x0000.
-- The schedule has a run-time priority, as defined by dmsRunTimePriority.6.1, that
-- overrides the run-time priority of the called message. Thus, the run-time priority
-- is constant for all scheduled messages and the central system can set this priority
-- by modifying the value of dmsRunTimePriority.6.1 If an invalid message code is
-- received, the sign will continue operations as if the code was not received, after
-- the correct response is transmitted.

-- The third special condition pertains to selecting the currently displayed message.
-- This condition is only valid for the 'MessageIDCode' SYNTAX, not for
-- the 'MessageActivationCode' SYNTAX. Specifying the currentBuffer 5.1 within
-- the MessageMemoryType and MessageNumber fields of the 'MessageIDCode' SYNTAX
-- will be used within the default messages such as
-- dmsShortPowerRecoveryMessage. This allows 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. The
-- messageCRC field of the default messages (such as dmsShortPowerRecoveryMessage)
-- shall be 0x0000, when the messageMemoryType field has a value of 'currentBuffer'.

```

```

dmsSignCfg    OBJECT IDENTIFIER ::= {dms 1}
-- This node is an identifier used to group all objects for DMS sign
-- configurations that are common to all DMS devices.

```

```

dmsSignType   OBJECT-TYPE
SYNTAX  INTEGER{
other (1),
bos (2),
cms (3),
vmsChar (4),
vmsLine (5),
vmsFull (6),
portableOther (129),
portableBOS (130),
portableCMS (131),
portableVMSChar (132),
portableVMSLine (133),
portableVMSFull (134)
}
ACCESS read-only

```

```

STATUS mandatory
DESCRIPTION "Indicates the type of sign."
--The descriptions are:
--other: Device not specified through any other definition, refer to device manual,
--bos: Device is a Blank-Out Sign,
--cms : Device is a Changeable Message Sign,
--vmsChar : Device is a Variable Message Sign with character matrix setup,
--vmsLine : Device is a Variable Message Sign with line matrix setup,
--vmsFull: Device is a Variable Message Sign with full matrix setup.
--Same is true for all portable signs.
::= {dmsSignCfg 2}

-- for LED signs, the value of this object will most likely be (2)
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)}
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Indicates the configuration of the type, numbers and flashing patterns of beacons on a sign."
--The definitions are:
--other: Other types, numbers and patterns of beacons on a sign
--supported by the device
--none: Patterns of beacons not supported by the device
--oneBeacon: 1 beacon flashing
--twoBeaconSyncFlash: 2 beacons, synchronized flashing
--twoBeaconsOppFlash: 2 beacons, opposing flashing
--fourBeaconSyncFlash: 4 beacons, synchronized flashing
--fourBeaconAltRowFlash: 4 beacons, alternate row flashing
--fourBeaconAltColumnFlash: 4 beacons, alternate column flashing
--fourBeaconAltDiagonalFlash: 4 beacons, alternate diagonal flashing
--fourBeaconNoSyncFlash: 4 beacons, no synchronized flashing
--oneBeaconStrobe: 1 beacon, strobe light
--twoBeaconStrobe: two beacons, strobe light
--fourBeaconStrobe: 4 beacons, strobe light
::= {dmsSignCfg 8}

-- this object should be required even though it is part of the
--GUI Appearance Conformance Group. Thus, conformance to this Conformance
--Group cannot be claimed by supporting only this one object.
dmsSignTechnology OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the utilized technology in a bitmap format (Hybrids will have to set the
bits for all technologies that the sign utilizes).
Bit 0- Other,
Bit 1- LED,
Bit 2- Flip Disk,
Bit 3- Fiber Optics,
Bit 4- Shuttered,
Bit 5- Bulb,
Bit 6- Drum
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."
::= {dmsSignCfg 9}

```

```

--All objects under the following node can be used to determine how
--the sign physically looks. However, support of these objects
--is NOT a requirement based on the functional requirements.
--FDOT NEEDS TO DECIDE WHETHER THESE OBJECTS ARE TO BE REQUIRED.
--.
vmsCfg OBJECT IDENTIFIER ::= {dms 2}
-- This subnode is an identifier used to group all objects for support
-- of VMS sign configurations that are common to all VMS devices.

vmsCharacterHeightPixels OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the height of a single character in Pixels. The value zero (0) Indicates a
variable character height."
::= {vmsCfg 1}

vmsCharacterWidthPixels OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the width of a single character in Pixels. The value zero (0) indicates a
variable character width."
::= {vmsCfg 2}
--A full matrix sign is indicated by a height and width of zero (0). A line matrix
-- sign is indicated by a width of zero (0).

vmsSignHeightPixels OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the number of rows of pixels for the entire sign."
::= {vmsCfg 3}
--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
-- for the sign.

vmsSignWidthPixels OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the number of columns of pixels for the entire sign."
::= {vmsCfg 4}
--To determine the number of characters for a character matrix sign, divide the
--vmsSignWidthPixels object value by the vmsCharacterWidthPixels object value.
-- This should result in a whole number, the number of characters per line
-- for the sign.

vmsHorizontalPitch OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the horizontal distance from the center of one pixel to the center of the
neighboring pixel in millimeters."
::= {vmsCfg 5}

vmsVerticalPitch OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the vertical distance from the center of one pixel to the center of the
neighboring pixel in millimeters."
::= {vmsCfg 6}
--If a central figures out that the sign is a character matrix sign, then
-- it should automatically impose some space between the characters.
-- The same is valid for line matrix signs.

-- All objects under the following node are needed to create and edit
-- fonts that are then downloaded into the sign controllers memory.
-- A font is called either automatically by default or manually by

```

```

-- indicating a non-default font in the value of the
-- dmsMessageMultiString object.
fontDefinition OBJECT IDENTIFIER ::= {dms 3}
-- This node is an identifier used to group all objects for DMS
-- font configurations that are common to DMS devices.

-- FDOT requires support of at least 4 fonts, as per Functional Requirement.
numFonts OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the maximum number of fonts that the sign can store."
::= {fontDefinition 1}

fontTable OBJECT-TYPE
SYNTAX SEQUENCE OF FontEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "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 will
yield unpredictable results."
::= {fontDefinition 2}

fontEntry OBJECT-TYPE
SYNTAX FontEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Parameters of the Font Table."
INDEX {fontIndex}
::= {fontTable 1}

FontEntry ::= SEQUENCE {
    fontIndex          INTEGER,
    fontNumber         INTEGER,
    fontName           DisplayString,
    fontHeight         INTEGER,
    fontCharSpacing    INTEGER,
    fontLineSpacing    INTEGER,
    fontVersionID      INTEGER}

fontIndex OBJECT-TYPE
SYNTAX INTEGER (1..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the row number of the entry."
::= {fontEntry 1}

fontNumber OBJECT-TYPE
SYNTAX INTEGER (1..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION "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. A device shall return a GenError if this value is not unique."
::= {fontEntry 2}

fontName OBJECT-TYPE
SYNTAX DisplayString (SIZE (0..64))
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates the name of the font."
::= {fontEntry 3}

fontHeight OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates the height of the font in pixels. Setting this object to zero (0)
invalidates this fontTable row, and also invalidates all corresponding entries into the
characterTable."
::= {fontEntry 4}

```

```

fontCharSpacing OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates the default horizontal spacing (in pixels) between each of the characters
within the font. This object only applies to Full Matrix and Line Matrix VMS. If the font changes
on a line, then the average value of the two fonts shall be used between sequential characters.
Value of spacing is rounded up to nearest whole pixel. Character Matrix VMS shall either set this
object to zero (0), or not support this object."
--DEFVAL {2} -FDOT-specific default of 2 pixel columns between characters.
--This value can be modified via the 'spacing character' tag
--on a per message-basis.
 ::= {fontEntry 5}

fontLineSpacing OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates the default vertical spacing (in pixels) between each of the lines within
the font. This object only applies to Full Matrix. 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 line spacing of each line. Value of spacing is rounded up to nearest whole pixel.
Character Matrix VMS and Line Matrix VMS shall either set this object to zero (0), or not support
this object."
--DEFVAL {2} -FDOT-specific default of 2 pixel rows between lines.
--This value can only be changed via this object and
--is valid for all messages.
 ::= {fontEntry 6}

fontVersionID OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION "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 FontVersionByteStream."
--
-- The following 4 definitions are used to define the above referenced
-- FontVersionByteStream.
--
-- Complete definitions for these referenced objects, including size
-- information, is contained elsewhere in this document.
--
-- CharacterInformation describes the characteristics of a single character and
-- defines the objects and order of the objects within
-- one row of CharacterInfoList.
--
-- CharacterInformation ::= SEQUENCE {
--     characterNumber INTEGER,
--     characterWidth INTEGER,
--     characterBitmap OCTET STRING }
--
-- CharacterInfoList describes the characteristics of each defined character
-- within a given font and orders the information by the
-- characterNumber in an increasing format.
--
-- CharacterInfoList ::= SEQUENCE OF CharacterInformation
--
-- 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.
--
-- FontInformation ::= SEQUENCE {
--     fontNumber INTEGER,
--     fontHeight INTEGER,
--     fontCharSpacing INTEGER,
--     fontLineSpacing INTEGER }
--

```

```

-- FontVersionByteStream defines the order of information used to construct
-- the byte stream which will be used to calculate the CRC. It consists
-- of the main font characteristics followed by n rows of
-- CharacterInfoList. The characterInfoList shall be for the fontNumber
-- indicated within the fontInformation field.
-- CharacterInfoList rows are included in FontVersionByteStream only when the
-- associated characterWidth object value is non-zero.
--
-- FontVersionByteStream ::= SEQUENCE {
--     fontInformation          FontInformation,
--     characterInfoList       CharacterInfoList }
--
::= {fontEntry 7}

maxFontCharacters OBJECT-TYPE
SYNTAX  INTEGER (1..65535)
ACCESS  read-only
STATUS  mandatory
DESCRIPTION "Indicates the maximum number of rows in the character table that can exist for any
given font."
::= {fontDefinition 3}

characterTable OBJECT-TYPE
SYNTAX  SEQUENCE OF CharacterEntry
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION "A table containing the information needed to configure/define each character of a
particular font."
::= {fontDefinition 4}

characterEntry OBJECT-TYPE
SYNTAX  CharacterEntry
ACCESS  not-accessible
STATUS  mandatory
DESCRIPTION "Parameters of the Character Configuration Table."
INDEX  {fontIndex, characterNumber}
::= {characterTable 1}

CharacterEntry ::= SEQUENCE {
    characterNumber          INTEGER,
    characterWidth           INTEGER,
    characterBitmap          OCTET STRING}

characterNumber  OBJECT-TYPE
SYNTAX  INTEGER (1..65535)
ACCESS  read-only
STATUS  mandatory
DESCRIPTION "Indicates the binary value associated with this character of this font. For example,
if the font set followed the ASCII numbering scheme, the character giving the bitmap of 'A' would
be characterNumber 65 (41 hex)."
::= {characterEntry 1}

characterWidth  OBJECT-TYPE
SYNTAX  INTEGER (0..255)
ACCESS  read-write
STATUS  mandatory
DESCRIPTION "Indicates the width of this character in pixels. A width of zero (0) indicates this
row is invalid."
::= {characterEntry 2}

characterBitmap OBJECT-TYPE
SYNTAX  OCTET STRING
ACCESS  read-write
STATUS  mandatory
DESCRIPTION "A bitmap that defines each pixel within a rectangular region as being either ON
(bit=1) or OFF (bit=0). The result of this bitmap is how the character appears on the sign."
::= {characterEntry 3}
-- 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. After the rectangular region is defined, any
-- remaining bits shall be zero (0).

-- All objects under the following node are needed to define default values for
-- the various 'tags' that are used within the MULTI Message String. Their use
--is not necessary (or a requirement), since each tag and tag value
--can be specifically called out. However, the support of these objects
--is useful for shortening the dmsMessageMultiString object.
multiCfg OBJECT IDENTIFIER ::= {dms 4}
-- This subnode is an identifier used to group all objects for support of
-- MULTI language configuration such as all default tag values.

defaultBackgroundColor OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates the color of the background shown on the sign. The allowed values are:
black (0),
red (1),
yellow (2),
green(3),
cyan (4),
blue (5),
magenta (6),
white (7),
orange (8),
amber (9).
Each of the background colors on a sign should map to one of the colors listed. If a color is
requested that is not supported, then a GenError shall be returned."
--DEFVAL {0} -Functional requirements state background to be 'black'
::= {multiCfg 1}

defaultForegroundColor OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates the color of the foreground (the actual text) shown on the sign. The
allowed values are:
black (0),
red (1),
yellow (2),
green(3),
cyan (4),
blue (5),
magenta (6),
white (7),
orange (8),
amber (9).
Each of the colors on a sign should map to one of the colors listed. If a color is requested that
is not supported, then a GenError shall be returned."
--DEFVAL {7} -Functional Requirements indicate foreground to be one color, assumed 'white'
::= {multiCfg 2}

defaultFlashOn OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION "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."
::= { multiCfg 3}

defaultFlashOff OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION "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."
::= { multiCfg 4}

```

```

defaultFont OBJECT-TYPE
SYNTAX INTEGER (1..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates the default font number (fontNumber-object) for a message."
 ::= { multiCfg 5}

defaultJustificationLine OBJECT-TYPE
SYNTAX INTEGER {
    other(1),
    left(2),
    center(3),
    right(4)
    -- full(5) deleted this value since not required/desired according to FunctReqs
}
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates the default line justification for a message."
 ::= {multiCfg 6}

defaultJustificationPage OBJECT-TYPE
SYNTAX INTEGER {
    other(1),
    top(2),
    middle(3),
    bottom(4) }
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates the default page justification for a message."
 ::= {multiCfg 7}

defaultPageOnTime OBJECT-TYPE
SYNTAX INTEGER (1..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION "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."
 ::= {multiCfg 8}

defaultPageOffTime OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION "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."
 ::= {multiCfg 9}

--Kept the following object even though its functionality was not identified
--in the Functional Requirements, because it is mandatory in this group
--and it may be needed in the future.
defaultCharacterSet OBJECT-TYPE
SYNTAX INTEGER {
    other (1),
    eightBit (2)}
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates the default number of bits used to define a single character in a MULTI
string.
other - a character size other than those listed below, refer to the device manual.
eightBit - each characterNumber of a given font is encoded as an 8-bit value."
 ::= {multiCfg 10}
--The intent of this object is to provide a mechanism by which 16-bit character
-- sets (and potentially other character sets ) can be supported in a
-- future version. Currently, this object only provides a standard
-- for 8-bit character encoding.

-- The following node is ONE of TWO mandatory Conformance Groups within
-- the DMS-specific Standard (NTCIP 1203). The objects listed under
-- this node are needed.

```

```

dmsMessage OBJECT IDENTIFIER ::= {dms 5}
-- This node is an identifier used to group all objects for support of
-- DMS Message Table functions that are common to DMS devices.

dmsNumPermanentMsg OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION "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."
::= {dmsMessage 1}

dmsNumChangeableMsg OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the current number of Messages stored in non-volatile, changeable memory.
For CMS and BOS, this number shall be zero (0)."
```

```

::= {dmsMessage 2}

dmsMaxChangeableMsg OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION "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)."
```

```

::= {dmsMessage 3}

dmsFreeChangeableMemory OBJECT-TYPE
SYNTAX INTEGER (0..4294967295)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the number of bytes available within non-volatile, changeable memory. For
CMS and BOS, this number shall be zero (0)."
```

```

::= {dmsMessage 4}

dmsNumVolatileMsg OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the current number of Messages stored in volatile, changeable memory. For
CMS and BOS, this number shall be zero (0)."
```

```

::= {dmsMessage 5}

dmsMaxVolatileMsg OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION "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)."
```

```

::= {dmsMessage 6}

dmsFreeVolatileMemory OBJECT-TYPE
SYNTAX INTEGER (0..4294967295)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the number of bytes available within volatile, changeable memory. For CMS
and BOS, this number shall be zero (0)."
```

```

::= {dmsMessage 7}

dmsMessageTable OBJECT-TYPE
SYNTAX SEQUENCE OF DmsMessageEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "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 has the value of 'valid'."
::= {dmsMessage 8}

dmsMessageEntry OBJECT-TYPE
SYNTAX DmsMessageEntry
```

```
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "Parameters of the Message Table."
INDEX {dmsMessageMemoryType, dmsMessageNumber}
::= {dmsMessageTable 1}
```

```
DmsMessageEntry ::= SEQUENCE {
dmsMessageMemoryType  INTEGER,
dmsMessageNumber      INTEGER,
dmsMessageMultiString OCTET STRING,
dmsMessageOwner       OwnerString,
dmsMessageCRC         INTEGER,
dmsMessageBeacon      INTEGER, --object should not be required for FiberOptic Signs.
dmsMessagePixelFormat INTEGER,
dmsMessageRunTimePriority  INTEGER,
dmsMessageStatus      INTEGER
}
```

```
dmsMessageMemoryType OBJECT-TYPE
```

```
SYNTAX INTEGER {
other (1),
permanent (2),
changeable (3),
volatile (4),
currentBuffer (5),
schedule (6),
blank (7)
}
```

```
ACCESS read-only
```

```
STATUS mandatory
```

```
DESCRIPTION "Indicates the memory-type used to store a message. Also provides access to current message (currentBuffer) and currently scheduled message (schedule)."
```

```
-- 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.
```

```
-- Only one entry in the table can have the value of currentBuffer and the value of
```

```
-- the dmsMessageNumber object must be one (1);
```

```
-- 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 dmsMessageNumber-object-value
```

```
-- for this entry must be 1. This will be the displayed message when
```

```
-- the dmsMessageSourceMode is timebasedScheduler.
```

```
-- 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 (i.e., objects
```

```
-- with SYNTAX of either MessageIDCode or MessageActivationCode) can blank the sign at
```

```
-- a stated run-time priority. The run-time priority of the blank message is equal to
```

```
-- the message number (i.e., 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 dmsMultiString shall be an octet string of length zero (0). The activation
```

```
-- priority shall be determined from the activation priority of
```

```
-- the MessageActivationCode.
```

```
::= {dmsMessageEntry 1}
```

```
dmsMessageNumber OBJECT-TYPE
```

```
SYNTAX INTEGER (1..65535)
```

```
ACCESS read-only
```

```
STATUS mandatory
```

```
DESCRIPTION "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."
```

```
::= {dmsMessageEntry 2}
```

```
dmsMessageMultiString OBJECT-TYPE
```

```
SYNTAX OCTET STRING
```

```
ACCESS read-write
```

```
STATUS mandatory
```

DESCRIPTION "Contains the message written in MULTI-language. When the primary index is 'schedule', 'blank', 'currentBuffer', or 'permanent', this object shall return a genErr 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. The value of the MULTI string is not allowed to have any NULL character."
::= {dmsMessageEntry 3}

dmsMessageOwner OBJECT-TYPE
SYNTAX OwnerString
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates the owner or author of this row."
::= {dmsMessageEntry 4}

dmsMessageCRC OBJECT-TYPE
SYNTAX INTEGER(0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION "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 type or length fields. Note that the calculation shall assume a value of zero (0) for the dmsMessageBeacon object and/or for the dmsMessagePixelService object if they are not supported. For messages of the 'blank' Message Type, the above algorithm shall be ignored and the dmsMessageCRC value shall always be 0. For messages of the 'schedule' Message Type, the CRC value of the currently scheduled message shall always be returned."
::= {dmsMessageEntry 5}

--the following object should NOT be required for signs other than Fiber optic signs.

dmsMessageBeacon OBJECT-TYPE
SYNTAX INTEGER (0..1)
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates if connected beacon(s) are to be activated when the associated message is displayed. Zero (0) = Beacon(s) are Disabled; one (1) = Beacon(s) are Enabled. When the primary index is 'schedule', 'blank', 'currentBuffer', or 'permanent', this object shall return a genErr 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)."
--DEFVAL {0}
::= {dmsMessageEntry 6}

dmsMessagePixelService OBJECT-TYPE
SYNTAX INTEGER (0..1)
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates whether pixel service shall be enabled (1) or disabled (0) while this message is active. When the primary index is 'schedule', 'blank', 'currentBuffer', or 'permanent', this object shall return a genErr 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."
--DEFVAL {0}
::= {dmsMessageEntry 7}

dmsMessageRunTimePriority OBJECT-TYPE
SYNTAX INTEGER (1..255)
ACCESS read-write
STATUS mandatory
DESCRIPTION "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 dmsMessageType is 'schedule,' the value set in this object (i.e., dmsMessageRunTimePriority.6.1) shall override the run-time priority of the scheduled message. When the dmsMessageType is 'currentBuffer' the value returned shall be determined from the dmsMessageRunTimePriority of the message referenced in the dmsMessageTableSource object. . When the dmsMessageType is 'blank', the value returned shall be equal to the dmsMessageNumber of that particular message."
::= { dmsMessageEntry 8}

dmsMessageStatus OBJECT-TYPE
SYNTAX INTEGER {
notUsed (1),

```

modifying (2),
validating (3),
valid (4),
error (5),
modifyReq (6),
validateReq (7),
notUsedReq (8) }
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates the current state of the message. This state-machine allows for defining a
message, validating a message, and freeing message use."
--The enumerated values can be divided into two types, state values and command
-- values. State values can only be read, a GenErr shall occur if a SET is attempted.
-- Command values can be written to, and will cause a state change if accepted, and
-- thus cannot be returned. The states and commands are defined as follows:
--notUsed: This is a state value and indicates that the row does not contain any
-- valid message data. Controller memory may or may not be released to free memory
-- pool in this state. Reading an object from a row when this object is set to
-- notUsed in undetermined, i.e., last contents, or random data may be returned.
-- Setting any object (except this object) for a row that is notUsed shall return
--a GenError. The only valid command in this state is modifyReq.
--modifying: This is a state value and indicates that the row is being modified to
-- define a message. Modifying any objects (except this object) can only be done when
-- the row is in this state, otherwise a GenError shall be returned. The valid
-- commands in this state is validateReq and notUsedReq.
--validating: This is a state value and indicates that the controller is validating
-- all of the message data for the row. When validation is complete, the controller
-- will automatically change the state to either valid (message data is good), or
-- error (some error found within the message data). The only valid command is
--the notUsedReq command, which shall set the state to notUsed or return a GenErr.
--valid: This is a state and indicates the message data is valid and the message can
-- be activated. Activation of a message cannot occur in any other state. The valid
-- commands in this state are notUsedReq and modifyReq.
--error: This is a state and indicates that an error was detected during the
-- validation process. The valid commands in this state are modifyReq and notUsedReq.
--modifyReq: This is a command that indicates the user wishes to modify the row to
-- define a message. A GenError may be returned if the controller is in the notUsed
-- state and there is insufficient memory to define a new message. A successful
-- request will change the state of the row to modifying. An unsuccessful request
-- will leave the row in the same state as it was prior to the command. This
-- command can be issued while in the notUsed, valid and error states.
--validateReq: This is a command that indicates the user wishes to validate the
-- current message data. This command can only be issued while the row is in the
-- modify state, else a GenError shall be returned. A successful request will change
-- the state of the row to validating. An unsuccessful request will leave the row
-- in the same state as it was prior to the command.
--notUsedReq: This is a command that indicates the user wishes to end use of the
-- current message data. This command can be issued while the row is in the modify,
-- validating, valid, and error states. A successful request will change the state of
-- the row to notUsed. An unsuccessful request will leave the row in the same
-- state as it was prior to the command.
::= {dmsMessageEntry 9}

```

```

dmsValidateMessageError OBJECT-TYPE
SYNTAX INTEGER {
other (1),
none (2),
beacons (3),
pixelService (4),
syntaxMULTI (5)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION "This is an error code used to identify why a message was not validated. If multiple
errors occur, only the first value will be indicated. The syntaxMULTI error is further detailed in
the dmsMultiSyntaxError, dmsMultiSyntaxErrorPosition and dmsMultiOtherErrorDescription objects.
The use of values 3 and 4 for this object is strongly discouraged."
::= {dmsMessage 9}
-- Kept the enumeration of 'beacons' because this is a read-only object. However, the
-- sign should never return a value of '3' for this object when the sign is a
-- fiber optic sign.

```

```

signControl OBJECT IDENTIFIER ::= {dms 6}
-- This node is an identifier used to group all objects for support of DMS sign
-- control functions that are common to DMS devices.

dmsControlMode OBJECT-TYPE
SYNTAX INTEGER {
other (1),
local (2),
--external (3), - not needed per requirements
central (4),
centralOverride (5)--,
--simulation (6) - not needed per requirements
}
ACCESS read-write
STATUS mandatory
DESCRIPTION "A value indicating the selected control mode of the sign."
--The available modes are:
--other - Other control mode supported by the device (refer to device manual).
--local - Local control mode.
--external - External control mode.
--central - Central control mode.
--centralOverride - Central station took control over Local control, even though the
-- control switch within the cabinet was set to 'Local'.
--simulation - 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."
::= {signControl 1}

--while optional, this object should be supported
dmsSWReset OBJECT-TYPE
SYNTAX INTEGER (0..1)
ACCESS read-write
STATUS mandatory
DESCRIPTION "A software interface to initiates a controller reset. The execution of the controller
reset shall set this object to the value 0. Value zero (0) = no reset, value one (1) = reset."
::= {signControl 2}

dmsActivateMessage OBJECT-TYPE
SYNTAX MessageActivationCode
ACCESS read-write
STATUS mandatory
DESCRIPTION "A code indicating the message which the sign shall activate. The dmsActivateMsgError
object shall be set appropriately when this object is SET. If a message activation error occurs,
the new message shall not be displayed and a GenErr shall be returned. If a GET is performed on
this object, the sign shall respond with the value for the last message that was successfully
implemented; if this message was activated by a message ID code (such as EndDurationMessage), the
duration will indicate 65535 (infinite), the activate priority will indicate 255, and the source
address will indicate the source address of the sign."
::= {signControl 3}

--while optional, this object should be supported
dmsMessageTimeRemaining OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates the amount of remaining time in minutes that the current message shall be
displayed. 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 management station to extend or shorten the duration of the message. Setting this object to zero
shall result in the immediate display of the dmsEndDurationMessage."
::= {signControl 4}

--while not specifically required in the Functional Requirements, this object must
--be supported
dmsMsgTableSource OBJECT-TYPE
SYNTAX MessageIDCode
ACCESS read-only
STATUS mandatory

```

DESCRIPTION "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 MessageTable. The currently displayed message is stored in the currentBuffer, but the information regarding which message number generated the current message would be lost if not indicated through this object. When the scheduler has control of the sign, the value of this object shall point to the 'schedule' row of the table."

::= {signControl 5}

--while not specifically required, this object must be supported

dmsMsgRequesterID OBJECT-TYPE

SYNTAX IPAddress

ACCESS read-only

STATUS mandatory

DESCRIPTION "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)."

REFERENCE "RFC 1155, May 1990"

::= {signControl 6}

dmsMsgSourceMode OBJECT-TYPE

SYNTAX INTEGER {

other (1),

local (2),

external (3),

otherCom1(4),

otherCom2 (5),

otherCom3 (6),

otherCom4 (7),

central (8),

timebasedScheduler (9),

powerRecovery (10),

reset (11),

commLoss (12),

powerLoss (13),

endDuration (14)}

ACCESS read-only

STATUS mandatory

DESCRIPTION "Indicates the source that initiated the currently displayed message."

::= {signControl 7}

-- Kept all these enumerations not relevant to the installation in, because this is
-- a read-only object. However, the sign should never return the values of '3, 4, 5,
-- 6, 7' for this object because the sign should not support these values
--(based on Functional Reqs.).

--agency-specific conditions (such as exceeded LED Temperature) may also be the cause
-- for a message (BLANK MESSAGE to be specific). If this 'special' condition is
-- valid, the value of this object must be set to one (1 = 'other'), and the actual
-- reason is represented by the value of the 'fdotMsgSourceModeExtension' object.

--while all of the following objects are optional, they should be supported to allow
--activation of different messages based on the various conditions.

dmsShortPowerRecoveryMessage OBJECT-TYPE

SYNTAX MessageIDCode

ACCESS read-write

STATUS mandatory

DESCRIPTION "Indicates the message that is displayed after a short power recovery of the device. The length of time that defines a short power loss is indicated in the dmsShortPowerLossTime-object. "

--DEFVAL {0x07 0x01 0x00 0x00} -suggested value for FDOT (blank the sign)

::= {signControl 8}

dmsLongPowerRecoveryMessage OBJECT-TYPE

SYNTAX MessageIDCode

ACCESS read-write

STATUS mandatory

DESCRIPTION "Indicates the message that is displayed after a power recovery of the device. The length of time that defines a long power loss is indicated in the dmsShortPowerLossTime-object."

--DEFVAL {0x07 0x01 0x00 0x00} - suggested value for FDOT (blank the sign)

::= {signControl 9}

```

dmsShortPowerLossTime OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-write
STATUS mandatory
DESCRIPTION "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."
--DEFVAL {600} - suggested value for FDOT (10 minutes)
::= {signControl 10}

dmsResetMessage OBJECT-TYPE
SYNTAX MessageIDCode
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates the message that is displayed after a Reset (either software or hardware) of
the device. This assumes that the device can differentiate between a reset and a power loss."
--DEFVAL {0x07 0x01 0x00 0x00} - suggested value for FDOT (blank the sign)
::= {signControl 11}

dmsCommunicationsLossMessage OBJECT-TYPE
SYNTAX MessageIDCode
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates the message that is displayed after the loss of communications to the
device. If there is no default message defined after the duration expires, then the sign goes
blank."
--DEFVAL {0x07 0x01 0x00 0x00} - suggested value for FDOT (blank the sign)
::= {signControl 12}

dmsTimeCommLoss OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-write
STATUS mandatory
DESCRIPTION "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),
no communication loss shall occur."
--DEFVAL {600} - suggested value for FDOT (10 minutes)
::= {signControl 13}
--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.

-- this object must be supported for the Fiber optic/Flip disk signs. When the power
-- is out, a message can still be displayed using the flip disks only.
-- Pure LED signs cannot display a message during power outage, thus support of this
-- object for LED only signs is not required.
dmsPowerLossMessage OBJECT-TYPE
SYNTAX MessageIDCode
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates the message that is displayed during the loss of power of the device."
::= {signControl 14}

dmsEndDurationMessage OBJECT-TYPE
SYNTAX MessageIDCode
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates the message that is displayed after the indicated duration for a message has
expired and no other Message had been assigned to replace the previous Message."
::= {signControl 15}

dmsMemoryMgmt OBJECT-TYPE
SYNTAX INTEGER {
other (1),
normal (2),
clearChangeableMessages (3),
clearVolatileMessages (4)
}
ACCESS read-write
STATUS mandatory
DESCRIPTION "Allows the system to manage the device's memory."

```

```

--DEFVAL {2} - assumed
::= {signControl 16}

dmsActivateMsgError OBJECT-TYPE
SYNTAX INTEGER {
other (1),
none (2),
priority (3),
messageStatus (4),
memoryType (5),
messageNumber (6),
messageCRC (7),
syntaxMULTI (8),
localMode (9)}
ACCESS read-only
STATUS mandatory
DESCRIPTION "This is an error code used to identify why a message was not displayed. If multiple
errors occur, only the latest value will be indicated. The syntaxMULTI error is further detailed
in the dmsMultiSyntaxError, dmsMultiSyntaxErrorPosition and dmsMultiOtherErrorDescription objects."
::= {signControl 17}

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)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION "This is an error code used to identify the first detected syntax
error within the MULTI message."
-- other: An error other than one of those listed.
-- none: No error detected.
-- unsupportedTag: The referenced tag is not supported by this device.
-- unsupportedTagValue: The referenced tag value is not supported by this device.
-- textTooBig: Too many characters on a line, too many lines for a page, or font
-- is too large for display.
-- fontNotDefined: The referenced font is not defined in this device.
-- characterNotDefined: The referenced character is not defined in the selected font.
-- fieldDeviceNotExist: The referenced field device does not exist / is not
-- connected to this device.
-- fieldDeviceError: This device is not receiving input from the referenced field
-- device and/or the referenced field device has a fault.
-- flashRegionError: The flashing region selected cannot be flashed by this device.
-- tagConflict: The message cannot be displayed with the combination of tags and/or
-- tag implementation cannot be resolved.
-- tooManyPages: Too many pages of text exists in the message.
::= {signControl 18}

dmsMultiSyntaxErrorPosition OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION "This is the offset from the first character (i.e. first character has offset 0, second
is 1, etc.) of the MULTI message where the SYNTAX error occurred."
::= {signControl 19}

-- while optional, this object should be supported
dmsMultiOtherErrorDescription OBJECT-TYPE
SYNTAX DisplayString (SIZE (0..50))
ACCESS read-write
STATUS mandatory

```

DESCRIPTION "Indicates vendor-specified error message descriptions. Associated errors occurred due to vendor-specific MULTI-tag responses."
::= {signControl 20}

vmsPixelServiceDuration OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-write
STATUS mandatory
DESCRIPTION "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."
::= {signControl 21}

vmsPixelServiceFrequency OBJECT-TYPE
SYNTAX INTEGER (0..1440)
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates the pixel service cycle time (frequency) in minutes. Value of zero indicates continuous pixel service from vmsPixelServiceTime to the epoch of midnight. Value of 1440 indicates one pixel service in a 24-hour period."
--DEFVAL {1440} -FDOT-specific requirement (once every 24 hours)
::= { signControl 22}

vmsPixelServiceTime OBJECT-TYPE
SYNTAX INTEGER (0..1440)
ACCESS read-write
STATUS mandatory
DESCRIPTION "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."
--DEFVAL {0} -assumed value
::= { signControl 23}

illum OBJECT IDENTIFIER ::= {dms 7}
-- This node is an identifier used to group all objects supporting DMS sign
-- illumination functions that are common to DMS devices.

dmsIllumControl OBJECT-TYPE
SYNTAX INTEGER {
other (1),
photocell (2),
timer (3),
manual (4)
}
ACCESS read-write
STATUS mandatory
DESCRIPTION "Indicates the method used to select the Brightness Level. Photocell indicates that the Brightness Level is based on photocell status. Timer indicates the the Brightness Level is set by an internal timer. Manual indicates that the Brightness Level must be changed via the dmsIllumManLevel- object. When switching to manual mode from any other mode, the current brightness level shall automatically be loaded into the dmsIllumManLevel object."
--DEFVAL {2} -automatic control per photocell reading is assumed to be the default
::= {illum 1}

dmsIllumMaxPhotocellLevel OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the maximum value given by the dmsIllumPhotocellLevelStatus-object."
::= {illum 2}

--The following object is used to retrieve the highest reading of all
--3 required photocells.

dmsIllumPhotocellLevelStatus OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION "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

dmsIllumBrightnessValuesError OBJECT-TYPE

```
SYNTAX INTEGER {
  other (1),
  none (2),
  photocellGap (3),
  negativeSlope (4),
  tooManyLevels (5),
  invalidData (6)
}
```

ACCESS read-only

STATUS mandatory

DESCRIPTION "Indicates the error encountered when the brightness table was SET. 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."

::= {illum 8}

-- while optional, this object is needed to report the current light output as
-- required by the Functional Specifications.

dmsIllumLightOutputStatus OBJECT-TYPE

```
SYNTAX INTEGER (0..65535)
```

ACCESS read-only

STATUS mandatory

DESCRIPTION "Indicates the current physical light output value ranging from 0 (darkest) to 65535 (maximum output)."

::= {illum 9}

--Support of a scheduler within the sign controller is not a functional requirement.
--Thus, the Scheduling Action objects are NOT required, and were deleted from this MIB.

dmsStatus OBJECT IDENTIFIER ::= {dms 9}

-- This node is an identifier used to group all objects supporting DMS sign status
-- monitoring functions that are common to DMS devices.

statMultiFieldRows OBJECT-TYPE

```
SYNTAX INTEGER (0..255)
```

ACCESS read-only

STATUS mandatory

DESCRIPTION "Indicates the number of rows in the statMultiFieldTable that are currently being used."

::={dmsStatus 1}

statMultiFieldTable OBJECT-TYPE

```
SYNTAX SEQUENCE OF StatMultiFieldEntry
```

ACCESS not-accessible

STATUS mandatory

DESCRIPTION "A table containing the currently displayed value of a specified Field. The number of rows is given by the value of statMultiFieldRows-object."

::= { dmsStatus 2}

statMultiFieldEntry OBJECT-TYPE

```
SYNTAX StatMultiFieldEntry
```

ACCESS not-accessible

STATUS mandatory

DESCRIPTION "Parameters of the Status Multi Field Table."

INDEX {statMultiFieldIndex}

```

::={statMultiFieldTable 1}

StatMultiFieldEntry ::= SEQUENCE {
statMultiFieldIndex      INTEGER,
statMultiFieldCode       INTEGER,
statMultiCurrentFieldValue OCTET STRING}

statMultiFieldIndex  OBJECT-TYPE
SYNTAX  INTEGER (1..255)
ACCESS  read-only
STATUS  mandatory
DESCRIPTION  "The index number into this table indicating the sequential order of the field within
the MULTI-string."
 ::= {statMultiFieldEntry 1}

statMultiFieldCode  OBJECT-TYPE
SYNTAX  INTEGER (1..255)
ACCESS  read-only
STATUS  mandatory
DESCRIPTION  "Indicates the ID of the value of the statMultiCurrentFieldValue-object.  The field
codes are indicated under the 'Field'-tag in MULTI (see section 3)."
 ::= {statMultiFieldEntry 2}
-- see Section 2 (of this report) to view which 'Field' tags must be
-- supported to fulfill FDOT's requirements.

statMultiCurrentFieldValue  OBJECT-TYPE
SYNTAX  OCTET STRING (SIZE (0..50))
ACCESS  read-only
STATUS  mandatory
DESCRIPTION  "Indicates the currently displayed text of the MULTI-message for the corresponding
Field."
 ::= {statMultiFieldEntry 3}

-- While optional, this object should be supported
watchdogFailureCount  OBJECT-TYPE
SYNTAX  Counter
ACCESS  read-only
STATUS  mandatory
DESCRIPTION  "A counter indicating the number of watchdog failures that have been detected."
 ::= { dmsStatus 5}

-- while optional, this object should be supported.
dmsStatDoorOpen  OBJECT-TYPE
SYNTAX  INTEGER (0..255)
ACCESS  read-only
STATUS  mandatory
DESCRIPTION  "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."
 ::= { dmsStatus 6}

statError  OBJECT IDENTIFIER ::= {dmsStatus 7}
-- This node is an identifier used to group all objects supporting DMS sign message error status
-- functions that are common to DMS devices.

--This object should be returned in the status message indicating whether any of the
--components have an error.  If one is detected, the central system should issue
--a command to the correct set of objects to retrieve further details about the
--error condition.
shortErrorStatus  OBJECT-TYPE
SYNTAX  INTEGER (0..65535)
ACCESS  read-only
STATUS  mandatory
DESCRIPTION  "A bitmap of summary errors where the bits are defined as follows:
Bit 0- other error
Bit 1- communications error
Bit 2- power error
Bit 3- attached device error
Bit 4- lamp error
Bit 5- pixel error

```

```

Bit 6- photocell error
Bit 7- message error
Bit 8- controller error
Bit 9- temperature warning
Bit 10- fan 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. To track a history of transient error conditions
utilize the event logging table located in the Global Objects Definitions (NTCIP 1201).
 ::= {statError 1}

-- The following objects is used to activate pixel tests, and to store the pixel
-- failures in a table, which can be retrieved via polling.
pixelFailureTableNumRows OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION "The number of rows contained in the pixelFailureTable each indicating failed pixels."
 ::= { statError 2}

pixelFailureTable OBJECT-TYPE
SYNTAX SEQUENCE OF PixelFailureEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "A table containing the X and Y location of a failed pixel. The number of rows is
given by the value of pixelFailureTableNumRows -object."
 ::= { statError 3}

pixelFailureEntry OBJECT-TYPE
SYNTAX PixelFailureEntry
ACCESS not-accessible
STATUS mandatory
DESCRIPTION "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              INTEGER,
    pixelFailureXLocation          INTEGER,
    pixelFailureYLocation          INTEGER,
    pixelFailureStatus             INTEGER}

pixelFailureDetectionType OBJECT-TYPE
SYNTAX INTEGER {
other (1),
pixelTest (2),
messageDisplay(3)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the type of test/display that leads to the pixel failure entry. Once a
failed pixel is detected as failed, it is entered in the table as either pixelTest or
messageDisplay. In either table the failed pixel stays in the table until pixelTestActivation is
set to either test or clearTable. Detection type pixelTest and messageDisplay are considered
different methods of testing for failed pixels. The pixelTest method is considered a foreground
processing method of failed pixel detection. During a foreground pixel test, the message on the
display may or may not stay present on the display. The messageDisplay method is considered a
background processing method of failed pixel detection. During a background test, the readability
of the message should not be affected by the test. If background pixel test is supported by the
manufacturer, failed pixels detected during a background pixel test are entered in the
messageDisplay pixel failure type. Failed pixels detected during a foreground pixel test are
entered in the pixelTest pixel failure type."
 ::= {pixelFailureEntry 1}

pixelFailureIndex OBJECT-TYPE
SYNTAX INTEGER (1..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Enumerated listing of row entries. Within each pixelFailureType, entries shall start
with 1 and be sequential."

```

```

 ::= {pixelFailureEntry 2}

pixelFailureXLocation OBJECT-TYPE
SYNTAX INTEGER (1..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION "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."
 ::= {pixelFailureEntry 3}

pixelFailureYLocation OBJECT-TYPE
SYNTAX INTEGER (1..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION "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."
 ::= {pixelFailureEntry 4}

pixelFailureStatus OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the current status of the specified pixel and the operation which made this
determination. This is a bit field with the following format:
Bit 0 0: Stuck Off / 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"
 ::= {pixelFailureEntry 5}

pixelTestActivation OBJECT-TYPE
SYNTAX INTEGER {
other (1),
noTest (2),
test (3),
clearTable (4)
}
ACCESS read-write
STATUS mandatory
DESCRIPTION "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. The pixel failure table will be cleared (both messageDisplay and pixelTest types)
when a pixel test is started (test) or table is cleared (clearTable). Setting the value to test
will start the test, meaning this test will be executed once. Pixel failures identified by setting
this object to test are entered into the pixelTest type of the pixelFailureDetectionType. The sign
controller will automatically set the value of this object back to noTest after completion."
 ::= {statError 4}

-- The following objects are used to activate lamp tests, and to store the lamp
-- failures in the first 2 objects, which can be retrieved via polling.
-- These objects are only required for Fiber optic signs since LED signs
-- don't use lamps.
--The scheduler within the central system must be configured to SET the
--third object (lampTestActivation) to a value of 3 (=Test) to fulfill FDOT's
--requirement of a 'once-per-day' lamp test.
lampFailureStuckOn OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (0..255))
ACCESS read-only
STATUS mandatory
DESCRIPTION "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)."
 ::= { statError 5}

lampFailureStuckOff OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (0..255))
ACCESS read-only
STATUS mandatory
DESCRIPTION "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)."
 ::= { statError 6}

```

```

lampTestActivation OBJECT-TYPE
SYNTAX INTEGER {
other (1),
noTest (2),
test (3)
}
ACCESS read-write
STATUS mandatory
DESCRIPTION "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,
either in the lampFailureStuckOn- or in the lampFailureStuckOff-objects. Setting the value to test
will start the test, meaning this test will be executed once. The sign controller shall
automatically set the value of this object back to noTest after completion."
 ::= {statError 7}

-- The following 2 objects are used to activate fan tests, and to store the fan
-- failures within the first object, which can be retrieved via polling.
--The scheduler within the central system must be configured to SET the
--second object (fanTestActivation) to a value of 3 (=Test) to fulfill FDOT's
--requirement of a 'once-per-day' fan test.
fanFailures OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (0..4))
ACCESS read-only
STATUS mandatory
DESCRIPTION "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."
 ::= {statError 8}

fanTestActivation OBJECT-TYPE
SYNTAX INTEGER {
other (1),
noTest (2),
test (3)
}
ACCESS read-write
STATUS mandatory
DESCRIPTION "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 will start the test, meaning this test will be executed once.
The sign controller will automatically set the value of this object back to noTest after
completion."
 ::= {statError 9}

controllerErrorStatus OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION "A bitmap of controller related errors where the bits are defined as follows:
Bit 0- other controller error
Bit 1- PROM error
Bit 2- program/processor error
Bit 3- RAM 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."
 ::= {statError 10}

statPower OBJECT IDENTIFIER ::= {dmsStatus 8}
-- This node is an identifier used to group all objects supporting DMS sign power
-- status monitoring functions that are common to DMS devices.

--the following object assumes one object to report the sign display voltage for all
--sign display power sources. However, many signs support more than one display
--power source, and the Functional Requirements indicate reporting the
--voltages of every power source including UPS.
--Thus, if a sign supports only one power source, the following object must be used.
--Otherwise, the FDOT-specific 'power voltage' objects indicated below, must be
--used. Support of the following object is then not necessary.

```

```

signVolts OBJECT-TYPE
SYNTAX INTEGER (0..65535)
ACCESS read-only
STATUS mandatory
DESCRIPTION "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 display
voltage."
 ::= {statPower 1}

--The Functional Requirements for Permanent VMS do not require support of a generator.
--Thus, the objects used to monitor fuel levels and engine RPMs have been deleted.

lineVolts OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION "The DMS line voltage measurement in (1.0) volts. The range is 0 volts to 255 volts."
 ::= {statPower 5}

powerSource OBJECT-TYPE
SYNTAX INTEGER {
other (1),
powerShutdown (2),
noSignPower (3),
acLine (4),
generator (5),
solar (6),
battery (7)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the source of power that is currently utilized by the sign."
--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: indicates that the sign and controller are powered by battery with no
--significant charging occurring.
 ::= {statPower 6}

statTemp OBJECT IDENTIFIER ::= {dmsStatus 9}
-- This node is an identifier used to group all objects supporting DMS sign
-- temperature status monitoring functions that are common to DMS devices.

-- The following objects are used to monitor the minimum and maximum temperature
-- readings detected within the sign controller cabinet, the ambient temperature
-- around the sign housing, and the sign housing.
-- FDOT specific objects are indicated in Section 4 of this report, which allow
-- setting a user-defined threshold. This threshold leads to blanking the sign in
-- case the temperature of the LEDs (and therefore the temperature within the sign
-- housing) exceeds a critical threshold.
tempMinCtrlCabinet OBJECT-TYPE
SYNTAX INTEGER (-128..127)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the current temperature, single sensor, or the current minimum temperature
of multiple sensors, within the DMS Control Cabinet in degrees Celsius."
 ::= {statTemp 1}

tempMaxCtrlCabinet OBJECT-TYPE
SYNTAX INTEGER (-128..127)
ACCESS read-only
STATUS mandatory

```

DESCRIPTION "Indicates the current temperature, single sensor, or the current maximum temperature of multiple sensors, within the DMS Control Cabinet in degrees Celsius."
::= {statTemp 2}

tempMinAmbient OBJECT-TYPE
SYNTAX INTEGER (-128..127)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the current outside ambient temperature, single sensor, or the current minimum outside ambient temperature, multiple sensors in degrees Celsius."
::= {statTemp 3}

tempMaxAmbient OBJECT-TYPE
SYNTAX INTEGER (-128..127)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the current outside ambient temperature, single sensor, or the current maximum outside ambient temperature, multiple sensors in degrees Celsius."
::= {statTemp 4}

tempMinSignHousing OBJECT-TYPE
SYNTAX INTEGER (-128..127)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the current temperature, single sensor, or the current minimum temperature, multiple sensors in the sign housing in degrees Celsius."
::= {statTemp 5}

tempMaxSignHousing OBJECT-TYPE
SYNTAX INTEGER (-128..127)
ACCESS read-only
STATUS mandatory
DESCRIPTION "Indicates the current temperature, single sensor, or the current maximum temperature, multiple sensors in the sign housing in degrees Celsius."
::= {statTemp 6}

END