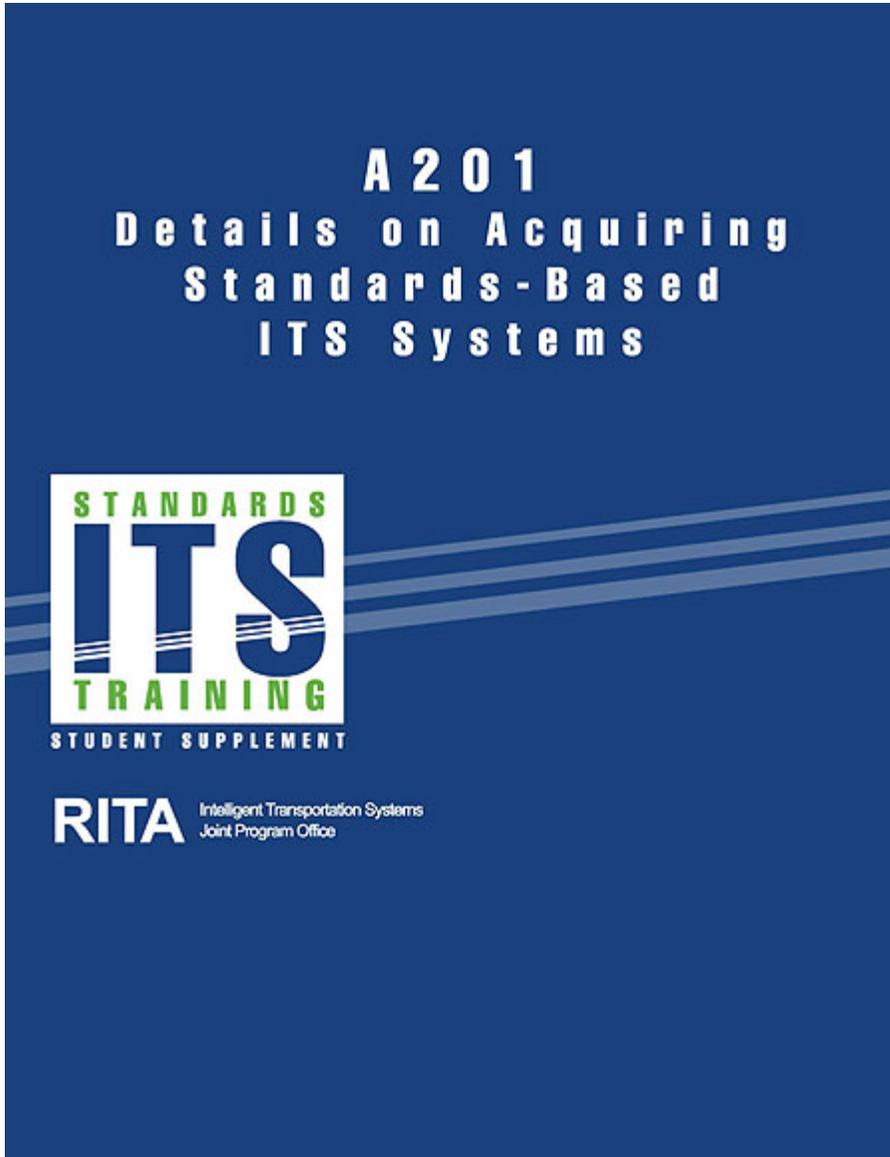


Module 6 - A201

Details on Acquiring Standards - Based ITS Systems

Student Supplement



(Extended Text Description: Large graphic cover page with dark blue background with the title in white letters “A201: Details on Acquiring Standards-Based ITS Systems” At the middle left is the “Standards ITS Training” logo with a white box and letters in blue and green. The words “Student Supplement” and “RITA Intelligent Transportation Systems Joint Program Office” in white lettering are directly underneath the logo. Three light blue lines move diagonally across the middle of the blue background.)

A201: Details on Acquiring Standards-based ITS Systems Table of Contents

Background - 2

Systems Engineering V Diagram - 3

TMDD Volume I Excerpts (Traffic Management Data Dictionary and

TMDD Volume II Excerpts Message Sets for External TMC Communications) - 5

Excerpts from NTCIP 1203 - Objects for DMS - 16

Excerpts from NTCIP 1201 - Global Objects - 26

Excerpts from NTCIP 1202 - Objects for Actuated Signal Controllers - 37

References - 56

Background

The enclosed supplemental material is intended to provide samples of what is contained in the ITS standards, more specifically the National Transportation Communications for ITS Protocol (NTCIP) and Traffic Management Data Dictionary (TMDD) standards. The excerpts were simply pulled from the pages of the standard and should not be used for construction.

The excerpts enclosed include two that have been through the Systems Engineering Process (SEP): the TMDD standard and the 1203 standard for DMS and two that have not: NTCIP 1201 for Global Objects and NTCIP 1202 for Actuated Signal Controllers.

- The excerpt from TMDD standard Volume 1 shows the development of the user needs (Section 2, pages 21-22) and the resulting development of the requirements (Section 3, pages 94-95); it then presents a sample of the *Needs to Requirements Traceability Matrix* (NRTM) Pages 220-221. These are representative of the discussion contained within the course material.
- The excerpt from TMDD Volume 2 shows samples of the dialog definitions, (pages 5758) and the message definitions (page 157). These are not discussed and are shown for reference only; most agencies will not need to work with this detail, unless they are extending the standard with custom changes.
- The excerpt from NTCIP 1203 standard for DMS shows the development of the user needs (Section 2, pages 25-26) and the Protocol Requirements List (PRL) table (Section 3 pages 39-40) and the detailed requirements (pages 81-82). Section 4, page 110 is representative of the detail for a specific dialog for downloading a message for display on the sign. Section 5 pages 140-141 are a sample of the object details.

For the Non SEP standards these materials include excerpts from NTCIP 1201 for global objects and NTCIP 1202 objects for actuated signal controllers based on the functionality and terminology described in NEMA TS2.

- The NTCIP 1201 excerpt shows the table of contents, which jumps straight into the object definitions. Samples of the object definitions (pages 7, 10-14) are shown and provide the only description of the functionality encompassed in the standard. There is a discussion of specific transactions in Annex A (enclosed pages 45-47).
- The NTCIP 1202 excerpt includes portions of the table of contents that show the general content and includes an excerpt from the object definitions (pages 8-11), block object definitions (pages 112-114), and the concept of conformance groups (similar to the PRL) that deal with groups of features/functions (pages 133-136); Annex B describes some of the consistency checks (page 156) to be performed when data is downloaded using the database feature; and Annex C (pages 160-162) shows dialogs for selected functions.

analyzed, and transformed into verifiable requirements that define what the system will do but not how the system will do it. Working closely with stakeholders, the requirements are elicited, analyzed, validated, documented, and version controlled.

The system design is created based on the system requirements including a high-level design that defines the overall framework for the system. Subsystems of the system are identified and decomposed further into components. Requirements are allocated to the system components and interfaces are specified in detail. Detailed specifications are created for the hardware and software components to be developed and final product selections are made for off-the-shelf components.

Hardware and software solutions are created for the components identified in the system design. Part of the solution may require custom hardware and/or software development and part may be implemented with off-the-shelf items modified as needed to meet the design specifications. The components are tested and delivered ready for integration and installation.

Once the software and hardware components have been developed, the "V" changes direction to an upward slope. The various components are individually verified and then integrated to produce higher-level assemblies or subsystems. These assemblies are also individually verified before being integrated with others to produce yet larger assemblies, until the complete system has been integrated and verified. Note that one portion of this testing should be conformance to the standards as documented in the procurement specification. While the system may work as an isolated system, if the future is the interconnection of systems, and geographical expansion with additional ITS devices, conformance to the standards is critical so that future procurements and connections with external systems can proceed as intended.

Eventually, the system is installed in the operational environment and transferred from the project development team to the organization that will own and operate it. The transfer also includes support equipment, documentation, operator training, and other enabling products that support ongoing system operation and maintenance. Acceptance tests are conducted to confirm that the system performs as intended in the operational environment. A transition period and warranty ease the transition to full system operation.

After the ITS system has passed system verification and is installed in the operational environment, the system owner/operator, whether the state DOT, a regional agency, or another entity, runs its own set of tests to make sure (i.e., validate) that the deployed system meets the original needs identified in the Concept of Operations.

Once the customer has approved the ITS system, the system operates in its typical steady state. System maintenance is routinely performed and performance measures are monitored. As issues, suggested improvements, and technology upgrades are identified, they are documented, considered for addition to the system baseline, and incorporated as funds become available. An abbreviated version of the systems engineering process is used to evaluate and implement each change. This occurs for each change or upgrade until the ITS system reaches the end of its operational life.

Finally, operation of the ITS system is periodically assessed to determine its efficiency. If the cost to operate and maintain the system exceeds the cost to develop a new ITS system, the existing system becomes a candidate for replacement. A system retirement plan will be generated to retire the existing system gracefully.

To find more information on the systems engineering process, see the links provided in the References section of this supplement.

TMDD Standard for Traffic Management Center-to-Center Communications
Volume I: Concept of Operations and Requirements

Balloted Standard
November 12, 2008

--	--	--

Joint ITE/AASHTO TMDD STANDARD FOR TRAFFIC MANAGEMENT CENTER-TO-CENTER COMMUNICATIONS	Standard Number:	Rev. 3.0 Balloted Standard
		Published: November 12, 2008
TMDD STANDARD FOR TRAFFIC MANAGEMENT CENTER-TO-CENTER COMMUNICATIONS <i>Volume I:</i> Concept of Operations and Requirements		

2.3.6.3.4 Need to Control a Remote Video Switch

Centers need to be able to request changes to the inputs and outputs of a video switch operated by another center.

When a control request is received, the center that controls the video switch needs to make a determination if the control request will be implemented, queued, or rejected. Then, the center that controls the video switch needs to send a response to the center that originated the request describing the status (action taken) on the control request.

2.3.6.3.5 Need to Verify Video Switch Control Status

The center that sends a request to control a video switch operated by another center needs to verify the status of such a control request. The status may be that the control was implemented, was queued, or was rejected.

2.3.6.3.6 Need to Cancel Video Switch Control Requests

Centers need to be able to "cancel" a video switch control request so the owner center knows that the control request is no longer required.

2.3.6.4 Need to Share DMS Status and Control

Dynamic message signs (DMS) are used by centers to help manage the surface transportation system. They can be used to:

- Provide travelers information that help the travelers select routes;
- Inform travelers about traffic congestion;
- Inform travelers about travel times;
- Inform travelers about roadway or traffic conditions;
- Inform travelers about planned activities that may affect traffic conditions;
- Provide information about transportation alternatives; and
- Provide other public service announcements.

2.3.6.4.1 Need to Share DMS Inventory

Centers need to exchange DMS inventory information so that DMSs operated by a center can become known to other centers. Centers need to exchange DMS device attributes so that the capabilities of the DMS devices operated by the owner center can become known to external centers.

Inventory information includes static DMS device attributes such as:

- *Location (including direction of traffic the DMS is facing);*
- *Size (physical dimensions, characters per line, number of lines); and*

- *Type (technology, permanent versus portable).*

2.3.6.4.2 Need to Share Updated DMS Inventory

Centers need to exchange updated inventory information as DMSs are added, removed, or changed. As centers add, remove, or change DMSs, the inventory should be provided to other centers, without requiring operator intervention. Changes may include the DMS location or technology.

2.3.6.4.3 Need to Share DMS Status

Centers need to exchange status information for each DMS. Status information includes:

- Communications status (connected, disconnected, failed);
- Operational status (available, not-available); and
- Current operational state information (contents of the display on the sign, etc.).

2.3.6.4.4 Need to Display a Message on a Remote DMS

Centers need to request that a specific message be displayed on a DMS controlled by another center. Messages may be either freeform text messages, in MULTI-string format, or from a library associated with the DMS.

When a control request is received the center that controls the DMS needs to make a determination if the message will be implemented, queued, or rejected. Then, the center that controls the DMS needs to send a response to the center that originated the request describing the status (action taken) on the control request.

2.3.6.4.5 Need to Verify DMS Control Status

The center that sends a request to display a specific message on a DMS operated by another center needs to confirm if the message was displayed. Possible statuses include that the message request was implemented, was queued, or was rejected.

2.3.6.4.6 Need to View DMS Message Queue

The center that sends a request to display a specific message on a DMS operated by another center needs to view the message queue for that DMS device if the request has been queued.

This control model assumes that there may be competing priorities for the use of a DMS and that a center may implement a queue or priority list of the control requests received, whether from the owner center itself or external centers, and their priority. External centers thus need to be able to read this queue to determine where their specific request to display a message currently "sits."

2.3.6.4.7 Need to Cancel DMS Message Requests

Centers need to be able to "cancel" a request to display a message on a DMS operated by another center. Once a message displayed on another center's DMS is no longer needed, a center should be able to cancel the message.

2.3.6.4.8 Need to Share DMS Message Appearance

Centers need to exchange information on how a message actually appears on the active face of a DMS operated by another center. Centers should properly confirm how a message will look on a DMS controlled by another center. How a message appears on a DMS may vary based on the physical attributes of the DMS (color, number of lines, number of characters per line, physical size, etc.) and its capabilities (fonts supported, MULTI-tags support, default values, etc.). Centers should send either freeform text messages, in MULTI-string format, or from a library associated with the DMS.

2.3.6.4.9 Need to Share DMS Message Inventory

Centers need to exchange the contents of a message library from a DMS operated by another center. This message library may be at the DMS itself or at the center. The contents of the message library are needed so that a center may know what messages are available for the DMS.

3.3.6.5.1 Share DMS Inventory Information

The requirements for sharing DMS inventory information with other authorized centers are as follows:

3.3.6.5.1.1 Send DMS Inventory Information Upon Request

An owner center shall respond to an authorized external center requesting DMS inventory with a message containing the owner center's DMS inventory information.

3.3.6.5.1.2 Publish DMS Inventory Information

An owner center shall publish a message containing its DMS inventory information to all authorized, subscribing external centers.

3.3.6.5.1.3 Subscribe to DMS Inventory Information

An external center shall send a subscription message to an owner center requesting its DMS inventory information.

3.3.6.5.1.4 Contents of the DMS Inventory Request

The requirements for DMS inventory requests from an external center to an owner center are found in Section 3.3.6.1.1.1, "Contents of Device Information Request", with the device type set to "dynamic message sign" and device information type set to "device inventory."

3.3.6.5.1.5 Contents of the DMS Inventory Information

An owner center shall send DMS inventory information to external centers.

3.3.6.5.1.5.1 Required DMS Inventory Content

The DMS inventory information sent from an owner center to an external center shall include:

- a. Generic device inventory header information (See Section 3.3.6.1.2.1); and
- b. Sign type (BOS, CMS, vmsChar, vmsLine, vmsFull, portable BOS, portable CMS, portable vmsChar, portable vmsLine, portable vmsFull, other, portable other, unknown) of each device.

3.3.6.5.1.5.2 Optional DMS Inventory Content

The following are optional requirements that an owner center may include in the DMS inventory information sent to an external center.

3.3.6.5.1.5.2.1 Sign Technology

The owner center shall provide the sign technology of the DMS as part of the DMS inventory information for each DMS. Supported values shall include LED, flip-disk, fiber optic, shuttered, bulb, drum, other and unknown.

3.3.6.5.1.5.2.2 Sign Height

The owner center shall provide the height of the DMS face, in millimeters, as part of the DMS inventory information for each DMS.

3.3.6.5.1.5.2.3 Sign Width

The owner center shall provide the width of the DMS face, in millimeters, as part of the DMS inventory information for each DMS.

3.3.6.5.1.5.2.4 Horizontal Border

The owner center shall provide the horizontal border around the DMS face, in millimeters, as part of the DMS inventory information for each DMS.

3.3.6.5.1.5.2.5 Vertical Border

The owner center shall provide the vertical border around the DMS face, in millimeters, as part of the DMS inventory information for each DMS.

3.3.6.5.1.5.2.6 Character Pixel Height

The owner center shall provide the character pixel height of the DMS as part of the DMS inventory information for each DMS. This is the pixel height of a single character. A value of 0 indicates a full matrix sign.

3.3.6.5.1.5.2.7 Character Pixel Width

The owner center shall provide the character pixel width of the DMS as part of the DMS inventory information for each DMS. This is the pixel width of a single character. A value of 0 indicates a line matrix or full matrix sign.

3.3.6.5.1.5.2.8 Sign Pixel Height

The owner center shall provide the pixel height of the DMS face as part of the DMS inventory information for each DMS.

3.3.6.5.1.5.2.9 Sign Pixel Width

The owner center shall provide the pixel width of the DMS face as part of the DMS inventory information for each DMS.

3.3.6.5.1.5.2.10 Sign Horizontal Pixel Pitch

The owner center shall provide horizontal distance between the center of one pixel to the center of the neighboring pixel in millimeters (pixel spacing) as part of the DMS inventory information for each DMS.

3.3.6.5.1.5.2.11 Sign Vertical Pixel Pitch

The owner center shall provide vertical distance between the center of one pixel to the center of the neighboring pixel in millimeters (pixel spacing) as part of the DMS inventory information for each DMS.

3.3.6.5.1.5.2.12 DMS Beacon Type

The owner center shall provide the beacon type that the DMS supports as part of the DMS inventory information for each DMS. Supported values shall include none, one beacon, two beacon sync flash, two beacon opposed flash, four beacon sync flash, four beacon alternate row flash, four beacon alternate column flash, four beacon alternate diagonal flash, four beacon no sync, one beacon strobe, two beacon strobe, four beacon strobe and other.

3.3.6.5.1.5.2.13 Maximum Number of Pages

The owner center shall provide the maximum number of pages that can be supported for a single message as part of the DMS inventory information for each DMS.

3.3.6.5.1.5.2.14 Maximum Message Length

The owner center shall provide the maximum length that can be supported for a single message as part of the DMS inventory information for each DMS.

UN ID	User Need	UN Selected	Requirement ID	Requirement	Conformance	Support	Other Requirements
			3.3.6.1.5.2	Contents of the Device Control Status Request	M	Yes	
				Contents of Device			

			3.3.6.1.5.3	Control Status Response	M	Yes	
			3.3.6.4.4	Request Video Switch Control Status	M	Yes	
2.3.6.3.6	Need to Cancel Video Switch Control Requests	Yes / No	3.3.6.1.6.1	Send Cancel Control Response Upon Request	M	Yes	
			3.3.6.1.6.2	Contents of Cancel Device Control Request	M	Yes	
			3.3.6.1.6.3	Contents of Cancel Device Control Request Response	M	Yes	
			3.3.6.4.5	Cancel Control Requests for Remote Video Switches	M	Yes	
			3.3.6.1.1.1	Contents of Device Information Request	M	Yes	
			3.3.6.1.1.1.1	Required Device Information Request Content	M	Yes	
			3.3.6.1.1.1.2.1	Username of the Requesting Operator	O	Yes / No	
			3.3.6.1.1.1.2.2	Password of the Requesting Operator	O	Yes / No	
			3.3.6.1.1.1.2.3	Owner Organization	O	Yes / No	
			3.3.6.1.1.1.2.4	External Center Organization	O	Yes / No	
			3.3.6.1.1.1.3	Content of Device Information Request	O	Yes / No	

2.3.6.4.1	Need to Share DMS Inventory	Yes / No		Filter			
			3.3.6.1.1.1.3.1	Device Identifier Filter	O	Yes / No	
			3.3.6.1.1.1.3.3	Roadway Network Identifier Filter	O	Yes / No	
			3.3.6.1.1.1.3.4	Link Identifier Filter	O	Yes / No	
			3.3.6.1.1.1.3.5	Route Designator Filter	O	Yes / No	
			3.3.6.1.1.1.3.6	Linear Reference Filter	O	Yes / No	
			3.3.6.1.2.1	Contents of the Device Inventory Header	M	Yes	
			3.3.6.1.2.1.1	Required Device Inventory Content	M	Yes	
			3.3.6.1.2.1.2.1	Device Description	O	Yes / No	
			3.3.6.1.2.1.2.2	Device Control Type	O	Yes / No	
			3.3.6.1.2.1.2.3	Controller Description	O	Yes / No	
			3.3.6.1.2.1.2.4	Uniform Resource Locator (URL)	O	Yes / No	

UN ID	User Need	UN Selected	Requirement ID	Requirement	Conformance	Support	Other Requirements
				Roadway Network		Yes /	

3.3.6.1.2.1.2.5	Identifier	O	No	
3.3.6.1.2.1.2.6	Node Identifier	O	Yes / No	
3.3.6.1.2.1.2.7	Node Name	O	Yes / No	
3.3.6.1.2.1.2.8	Link Identifier	O	Yes / No	
3.3.6.1.2.1.2.9	Link Name	O	Yes / No	
3.3.6.1.2.1.2.10	Link Direction	O	Yes / No	
3.3.6.1.2.1.2.11	Route Designator	O	Yes / No	
3.3.6.1.2.1.2.12	Linear Reference	O	Yes / No	
3.3.6.1.2.1.2.13	Linear Reference Version	O	Yes / No	
3.3.6.1.2.1.2.14	Owner Organization	O	Yes / No	
3.3.6.1.2.1.2.15	Inventory Date and Time Change Information	O	Yes / No	
3.3.6.5.1.1	Send DMS Inventory Information Upon Request	M	Yes	
3.3.6.5.1.2	Publish DMS Inventory Information	Subscription:O	Yes / No /	The owner center shall begin sending the updated response message within _ms after the information

				NA	is updated in the owner center.
3.3.6.5.1.3	Subscribe to DMS Inventory Information	Subscription:O		Yes/No / NA	
3.3.6.5.1.4	Contents of the DMS Inventory Request	M		Yes	
3.3.6.5.1.5	Contents of the DMS Inventory Information	M		Yes	
3.3.6.5.1.5.1	Required DMS Inventory Content	M		Yes	
3.3.6.5.1.5.2.1	Sign Technology	O		Yes / No	
3.3.6.5.1.5.2.2	Sign Height	O		Yes / No	
3.3.6.5.1.5.2.3	Sign Width	O		Yes / No	
3.3.6.5.1.5.2.4	Horizontal Border	O		Yes / No	
3.3.6.5.1.5.2.5	Vertical Border	O		Yes / No	
3.3.6.5.1.5.2.6	Character Pixel Height	O; WYSIWYG for Matrix Signs:M		Yes / No	

(Additional Notes from the Author: This is a table with columns labeled US ID for user need ID with a reference to the section of the standards which described the user need in detail; User Need – which is a short description of the user need; UN Selected is a column to allow the user to specify the user needs required to their procurement – they circle a yes/no options; requirement ID which references the section of the standard where the requirement is described; Conformance which indicates whether this requirement is mandatory or optional for the specific user need selected; Support allows the user or vendor to indicate whether this requirement is satisfied by their device/system or whether this requirement is required; and other requirements where qualifications are allowed. This is taken directly from the November 12, 2008 balloted version of TMDD Volume 1.)

TMDD Standard for Traffic Management Center-to-Center Communications
Volume II: Design Content

Balloted Standard
November 12, 2008

<i>Joint ITE/AASHTO</i> TMDD STANDARD FOR TRAFFIC MANAGEMENT CENTER-TO-CENTER COMMUNICATIONS	Standard Number:	Rev. 3.0 Balloted Standard
	Issued: November 12, 2008	
TMDD STANDARD FOR TRAFFIC MANAGEMENT CENTER-TO-CENTER COMMUNICATIONS <i>Volume II:</i> <i>Design Content</i>		

3.1.5.3.3 ASN.1 REPRESENTATION

```
dlDeviceInformationSubscription ITS-INTERFACE-DIALOGUE ::= {
  DESCRIPTIVE-NAME "ExternalCenter<-DIDeviceInformationSubscription->OwnerCenter" ASN-
  NAME "DIDeviceInformationSubscription"
  ASN-OBJECT-IDENTIFIER { tmddDialogs 15 }
  URL "Sub.gif"
  DEFINITION "A request-response dialog that allows an external center to request an owner center set up
  a subscription for updates on an owner center's device inventory, status and control schedule."
  DESCRIPTIVE-NAME-CONTEXT {"Manage Traffic"}
  ARCHITECTURE-REFERENCE { "traffic information coordination"
}
  ARCHITECTURE-NAME {"U.S. National ITS Architecture"}
  ARCHITECTURE-VERSION {"6.0"}
  DATA-CONCEPT-TYPE interface-dialogue
  STANDARD "TMDD"
  REFERENCED-MESSAGES {
  { tmddMessages 20 }, -- Input
  { c2cMessages c2cMessageReceipt(1) }, -- Output
  { tmddMessages 10 } -- Fault
}
  REFERENCED-OBJECT-CLASSES {
  { tmddObjectClasses ownerCenter(18) }, { tmddObjectClasses externalCenter(9) }
}
}
```

3.1.5.3.4 XML REPRESENTATION

```
<operation xmlns="http://schemas.xmlsoap.org/wsdl/"
name="DIDeviceInformationSubscription">
```

```

<input message="tns:MSG_DeviceInformationSubscription"/>
<output message="tns:MSG_ConfirmationReceipt"/>
<fault name="errorReport" message="tns:MSG_ErrorReport"/>
</operation>

```

3.1.6 DMS Class Dialogs

3.1.6.1 DIDMSControlRequest

3.1.6.1.1 DEFINITION

A request-response dialog that allows an external center to request an owner center to perform a control action on an owner center's dynamic message sign.

3.1.6.1.2 DIALOG REFERENCE

See Clause 2.4.1 Generic Request-Response Dialog

3.1.6.1.3 ASN.1 REPRESENTATION

```

dIDMSControlRequest ITS-INTERFACE-DIALOGUE ::= {
  DESCRIPTIVE-NAME "ExternalCenter<-DIDMSControlRequest->OwnerCenter"
  ASN-NAME "DIDMSControlRequest"
  ASN-OBJECT-IDENTIFIER { tmddDialogs 22 }
  URL "R-R.gif"
  DEFINITION "A request-response dialog that allows an external center to request an owner center to
  perform a control action on an owner center's dynamic message sign."
  DESCRIPTIVE-NAME-CONTEXT {"Manage Traffic"}
  ARCHITECTURE-REFERENCE { "traffic control coordination"
  }
  ARCHITECTURE-NAME {"U.S. National ITS Architecture"}
  ARCHITECTURE-VERSION {"6.0"}
  DATA-CONCEPT-TYPE interface-dialogue
  STANDARD "TMDD"
  REFERENCED-MESSAGES {
  { tmddMessages 22 }, -- Input
  { tmddMessages 18 }, -- Output
  { tmddMessages 10 ) -- Fault
  }
  REFERENCED-OBJECT-CLASSES {
  { tmddObjectClasses ownerCenter(18) },
  { tmddObjectClasses externalCenter(9) }
  }
}

```

3.1.6.1.4 XML REPRESENTATION

```

<operation xmlns="http://schemas.xmlsoap.org/wsdl/" name="DIDMSControlRequest">
<input message="tns:MSG_DMSSControlRequest"/>
<output message="tns:MSG_DeviceControlResponse"/>
<fault name="errorReport" message="tns:MSG_ErrorReport"/>
</operation>

```

3.1.6.2 DIDMSFontTableRequest

3.1.6.2.1 DEFINITION

A request-response dialog that allows an external center to request an owner center provide the font tables

for the owner center's dynamic message signs.

3.1.6.2.2 DIALOG REFERENCE

See Clause 2.4.1 Generic Request-Response Dialog

3.1.6.2.3 ASN.1 REPRESENTATION

```
dIDMSFontTableRequest ITS-INTERFACE-DIALOGUE ::= {
  DESCRIPTIVE-NAME "ExternalCenter<-DIDMSFontTableRequest->OwnerCenter"
  ASN-NAME "DIDMSFontTableRequest"
  ASN-OBJECT-IDENTIFIER { tmddDialogs 21 }
  URL "R-R.gif"
  DEFINITION "A request-response dialog that allows an external center to request an owner center
  provide the font tables for the owner center's dynamic message signs."
  DESCRIPTIVE-NAME-CONTEXT {"Manage Traffic"}
  ARCHITECTURE-REFERENCE { "traffic information coordination"
  }
  ARCHITECTURE-NAME {"U.S. National ITS Architecture"}
  ARCHITECTURE-VERSION {"6.0"}
  DATA-CONCEPT-TYPE interface-dialogue
  STANDARD "TMDD"
  REFERENCED-MESSAGES {
    { tmddMessages 24 }, -- Input
    { tmddMessages 2 3 }, -- Output
    { tmddMessages 10} -- Fault
  }
  REFERENCED-OBJECT-CLASSES {
    { tmddObjectClasses ownerCenter(18) },
    { tmddObjectClasses externalCenter(9)}
  }
  DESCRIPTIVE-NAME "dMSControlRequestMsg:message"
  ASN-NAME "dMSControlRequestMsg"
  ASN-OBJECT-IDENTIFIER { tmddMessages 22 }
  DEFINITION "The information content necessary to request a control action of an owner center's
  dynamic message sign."
  DESCRIPTIVE-NAME-CONTEXT {"Manage Traffic"}
  ARCHITECTURE-REFERENCE {
    "traffic control coordination"
  }
  ARCHITECTURE-NAME {"U.S. National ITS Architecture"}
  ARCHITECTURE-VERSION {"6.0"}
  DATA-CONCEPT-TYPE message
  STANDARD "TMDD"
  META-DATA-SOURCE direct
  PRIORITY "routine"
  FREQUENCY-OR-MESSAGE-MODE "on demand"
  REFERENCED-DATA-FRAMES {
    { tmddDataFrames 3 8 }
  }
  DATA-TYPE "
  DMSControlRequestMsg ::= DMSControlRequest
  "
}
```

3.2.6.1.3 XML REPRESENTATION

```
<xs:element name="dMSControlRequestMsg" type="DMSControlRequest"/>
```

3.2.6.2 DMSFontTableMsg

3.2.6.2.1 DEFINITION

The information content describing an owner center's dynamic message sign font tables.

3.2.6.2.2 ASN.1 REPRESENTATION

```
dMSFontTableMsg ITS-MESSAGE ::= {
  DESCRIPTIVE-NAME "dMSFontTableMsg:message"
  ASN-NAME "dMSFontTableMsg"
  ASN-OBJECT-IDENTIFIER { tmddMessages 23 }
  DEFINITION "The information content describing an owner center's dynamic message sign font tables."
  DESCRIPTIVE-NAME-CONTEXT {"Manage Traffic"}
  ARCHITECTURE-REFERENCE {
    "traffic information coordination"
  }
  ARCHITECTURE-NAME {"U.S. National ITS Architecture"}
  ARCHITECTURE-VERSION {"6.0"}
  DATA-CONCEPT-TYPE message
  STANDARD "TMDD"
  META-DATA-SOURCE direct
  PRIORITY "routine"
  FREQUENCY-OR-MESSAGE-MODE "on demand"
  REFERENCED-DATA-FRAMES {
    { tmddDataFrames 3 9 }
  }
  DATA-TYPE "
DMSFontTableMsg ::= DMSFontTable
"
}
```

A Recommended Standard of the Joint Committee on the NTCIP

NTCIP 1203 version v02.39

National Transportation Communications for ITS Protocol

Object Definitions for Dynamic Message Signs (DMS)

v02.39b November 2010

This is a draft pre-standard document, which is distributed for review and ballot 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 ballot to AASHTO, ITE, or NEMA. Please ensure that all copies include this notice. This pre-standard contains recommended information that is subject to approval.

Published by

American Association of State Highway and Transportation Officials (AASHTO)

444 North Capitol Street, N.W., Suite 249 Washington, D.C. 20001

Institute of Transportation Engineers (ITE)

1627 I ("Eye") Street, NW, Suite 600 Washington, D.C. 20006

2.4.2 Operational Environment

NTCIP 1203 v02 addresses the interface between a DMS and a management station (e.g., a central computer). To enable communications between these components, the transportation system manager needs to establish a communication system that links the DMS with a management station. For some systems, communications resource requirements may be minimal, and the system may be designed for constant polling; other systems may encounter significant resource requirements for communicating with the DMS, and the system may be designed to minimize data exchanges.

When deploying a DMS, the system designers consider which of the following operational environments need to be supported.

2.4.2.1 Live Data Exchange

The typical operational environment allows the management system to monitor and control the DMS by issuing requests (e.g., requests to access information, alter information, or control the sign). In this environment, the DMS responds to requests from the management station immediately (e.g., through the provision of live data, success/failure notice of information alteration, or success/failure of the command).

2.4.2.2 Logged Data Exchange

Some operational environments do not have always-on connections (e.g., dial-up links). In such environments, a transportation system operator may wish to define conditions under which data is placed into a log, which can then be uploaded at a later time. For example, the operator may wish to maintain a log of all messages posted on the sign, regardless of which management station or algorithm posted the message.

2.4.2.3 Exceptional Condition Reporting

In some operational environments, it may be desirable to have the DMS automatically transmit data to the management station when certain conditions occur. Under this scenario, the transportation system operator can define under what conditions the operator wishes to be notified, and the device automatically notifies the management station when the condition occurs. An example may be the transportation system manager who wants to know when the cabinet door is open.

2.5 FEATURES

Section 2.5 identifies and describes the various standardized user needs addressed by and 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.

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 the DMS Identity
- b. Determine Sign Display Capabilities
- c. Manage Fonts
- d. Manage Graphics
- e. Manage Brightness
- f. Address Multi-Version Interoperability (MVI, formerly known as backward compatibility)

Sub-feature details follow.

© 2010 AASHTO / ITE / NEMA
NTCIP 1203 v02.39 Page 26

2.5.1.1 Determine the DMS Identity

This feature allows the operator to determine basic information about the DMS, such as the type, technology, manufacturer, model, and version number of the DMS. It includes the ability to access information about both hardware and software elements of the DMS.

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 font height, width, and color. 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 Configure Low Fuel Threshold

This feature allows the operator to configure the threshold at which the fuel in a generator is to be considered low. This feature is associated with DMS using generators, which are typically portable signs.

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. Remotely Reset the Sign Controller

- c. Control the Sign Face
- d. Control External Devices
- e. Control the Brightness Outputs
- f. Perform Preventative Maintenance

Sub-feature details follow.

© 2010 AASHTO / ITE / NEMA
 NTCIP 1203 v02.39 Page 39

UN Section Number	User Need (UN)	FR Section Number	Functional Requirement (FR)	Conformance	Support / Project Requirement	Additional Project Requirements
		3.4.1.1	Retrieve Data	M	Yes	
		3.4.1.2	Deliver Data	M	Yes	
		3.4.1.3	Explore Data	M	Yes	
		3.4.4.1	Determine Current Access Settings	M	Yes	
		3.4.4.2	Configure Access	M	Yes	The DMS shall support at least access levels in addition to the administrator.
	Logged Data Exchange			O	Yes / No	
		H.2.2.1	Set Time	M	Yes	
		H.2.2.2	Set Time Zone	H.2.2.1:O	Yes / No	NOTE—New objects were added in NTCIP 1201 v02 GO and 1201 v03 GO to address functionality issues. Pay close attention to the implementation and interoperability of these objects.
		H.2.2.3	Set Daylight Savings Mode	H.2.2.1:O	Yes / No	
		H.2.2.4	Verify Current Time	M	Yes	Place a checkmark beside the GO major version(s) that the DMS is NOT required to support: NTCIP 1201:1996 (v01) NTCIP 1201:2005 (v02) NTCIP 1201 (v03)
2.4.2.2			Supplemental			

	H.2.6 t	Requirements for Event Monitoring	M	Yes	
	3.4.2.1	Determine Current Configuration of Logging Service	M	Yes	
	3.4.2.2	Configure Logging Service	M	Yes	
	3.4.2.3	Retrieve Logged Data	M	Yes	
	3.4.2.4	Clear Log	M	Yes	

© 2010 AASHTO / ITE / NEMA
Copy Per PRL Distribution Notice
NTCIP 1203 v02.39 Page 40

UN Section Number	User Need (UN)	FR Section Number	Functional Requirement (FR)	Conformance	Support / Project Requirement	Additional Project Requirements
		3.4.2.5	Determine Capabilities of Event Logging Service	M	Yes	
		3.4.2.6	Determine Total Number of Events	M	Yes	
2.4.2.3	Exceptional Condition Reporting			X	No	Exception Reporting is not yet supported by NTCIP.
2.5	Features			M	Yes	
2.5.1	Manage the DMS Configuration			M	Yes	
	Determine the DMS Identity			M	Yes	

2.5.1.1		3.5.1.1.1	Determine Sign Type and Technology	M	Yes	
		H.2.1	Determine Device Component Information	M	Yes	
		H.2.4	Determine Supported Standards	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	
		3.5.1.2.1.3	Determine Beacon Type	M	Yes	
		3.5.1.2.1.4	Determine Sign Access and Legend	M	Yes	
		3.5.1.2.2.1	Determine Sign Face Size in Pixels	Matrix:M	Yes / NA	
		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	See PRL 3.6.6.2.1.	

(Additional Notes from the Author: This is a PRL table taken directly from the NTCIP 1203 version 2.29 standard; it is the PRL example and is a table with the following headings: US Section Number, User Need description, FR Section Number for functional requirement – identifying the section which describes the requirement; Functional requirement which is a brief text description of the requirement; conformance which indicates whether the requirement is mandatory or optional for the user need; Support / Project Requirement which indicates if the project requires this feature/function; Additional Project Requirements which allow the agency to specify ranges or specific values.)

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.

3.5.2.3.5.1.4 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.

3.5.2.3.5.1.5 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.

3.5.2.3.5.1.6 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.

3.5.2.3.6 Activate a Message with Status

The DMS shall adhere to requirement 3.5.2.3.1 "Activate a Message". The DMS shall provide status of any message activation for slow activating message signs such as drum signs.

3.5.2.4 Control External Devices

The following requirements apply to a DMS supporting control and monitoring of any connected external devices (e.g. gates). Requirements pertaining to usage of any data received by any external devices are outside of the scope of NTCIP 1203 v02. The received data might be used by the DMS or be passed through to the management station.

NOTE—Generally, the following requirements are applicable to DMS conforming to either Version 1 or Version 2 of NTCIP 1203. However, one object was added in Version 2 and the object identifiers were modified. Any requirements not applicable to Version 1 have been pointed out below, in the PRL, as well as other sections.

3.5.2.4.1 Determine Configuration of External Device Ports

The following requirements allow a management station to determine the configuration characteristics of any pre-configured ports controlling external devices supported by the DMS.

3.5.2.4.1.1 Determine Base Configuration of External Device Ports

The DMS shall allow a management station to determine the basic configuration characteristics of any pre-configured ports controlling external devices supported by the DMS. These configuration characteristics shall be:

- a. type of port - digital or analog
- b. port number
- c. resolution of the port - the number of bits used for a port controlling an external device
- d. direction of the port - input, output, or bi-directional

3.5.2.4.1.2 Further Define Ports

The DMS shall allow a management station to set the port description of the external device configuration parameter to further define and/or describe the purpose of the supported ports.

3.5.2.4.1.3 Number of External Devices Supported

The DMS shall support the number of external devices specified in the agency specification. Note: this functional requirement is not applicable to version 1.

3.5.2.4.2 Monitoring of External Devices

The following requirements allow a management station to monitor pre-defined external devices via any configured port, if the port is configured for input in the DMS being monitored.

3.5.2.4.2.1 Retrieving Data from External Devices

The DMS shall allow a management station to retrieve data from an external device via any configured port via the 'as input'-configured ports to support monitoring of the external device.

3.5.2.4.3 Controlling of External Devices

The following requirements allow a management station to control pre-defined external devices via any configured port, if the port is configured for output in the DMS being controlled.

3.5.2.4.3.1 Passing Data to External Devices

The DMS shall allow a management station to send data to an external device via any configured port via the 'as output'-configured ports to support control of the external device.

3.5.2.4.3.2 Determine Status of External Devices

The DMS shall allow a management station to determine the last commanded state sent to the external device via the 'as output'-configured ports.

3.5.2.4.4 Controlling of Bi-directionally Connected External Devices

The following requirements allow a management station to monitor and control pre-defined external devices via any configured port, if the port is configured for bi-directional data exchange in the DMS being monitored.

3.5.2.4.4.1 Retrieving Data from External Devices

The DMS shall allow a management station to retrieve data from an external device via any configured port via the bi-directionally configured ports to support monitoring of the external device.

3.5.2.4.4.2 Passing Data to External Devices

The DMS shall allow a management station to send data to an external device via any configured port via the bi-directionally configured ports to support control of the external device.

3.5.2.4.4.3 Determine Status of External Devices

The DMS shall allow a management station to determine the last commanded state sent to the external device via the bi-directionally configured ports.

Note: this functional requirement is not applicable to version 1.

3.5.2.5 Control Sign Brightness

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

3.5.2.5.1 Determine Number of Brightness Levels

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

3.5.2.5.2 Determine Current Photocell Readings

The DMS shall allow a management station to determine the current photocell readings.

3.5.2.5.3 Manually Direct-Control Brightness (Version 2)

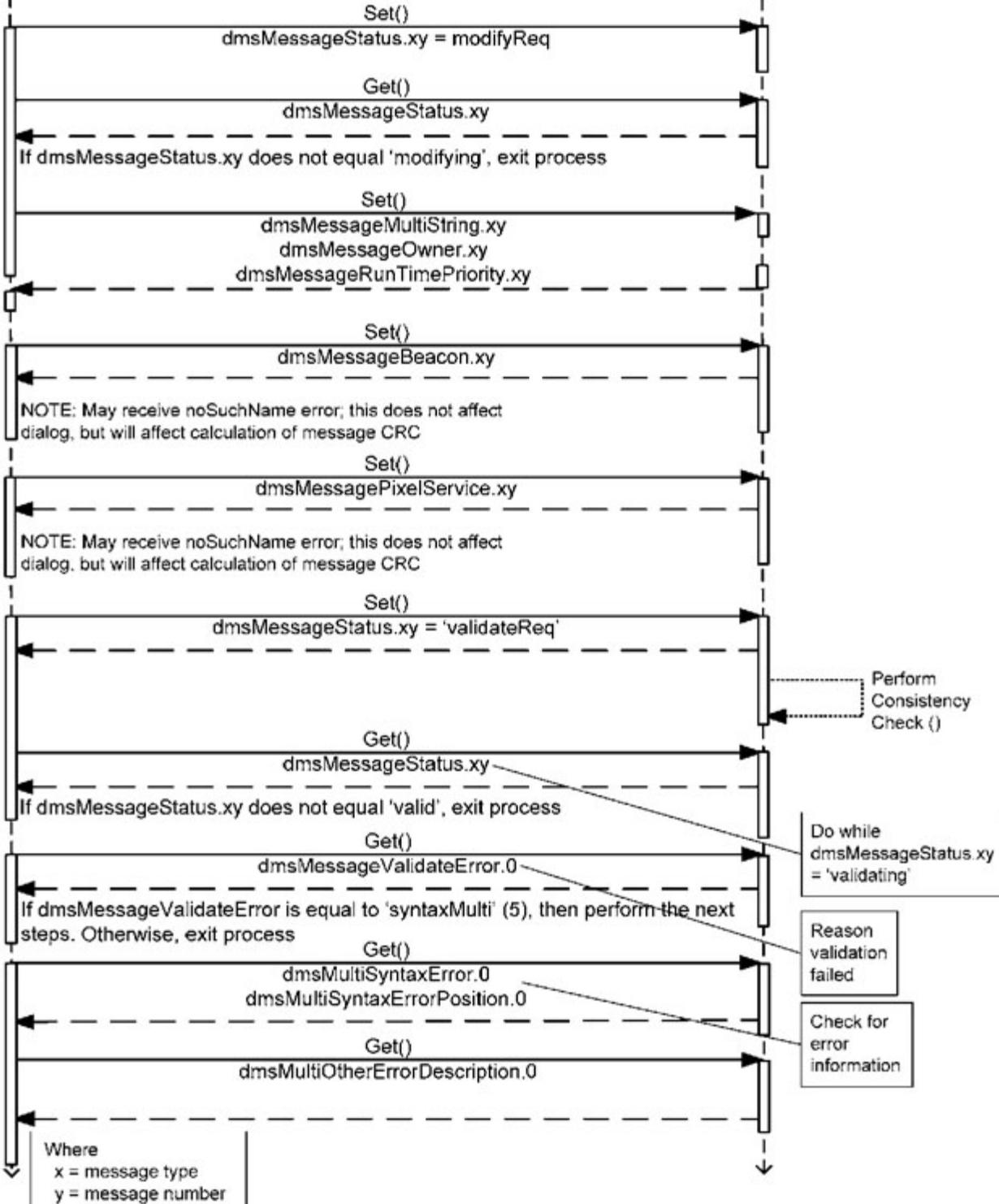
The DMS shall allow a management station to manually control the light output of the display by selecting any of the brightness levels supported by the DMS.

Preconditions:
 The management station shall ensure that the DMS supports the desired volatile or changeable message number and the tags within the messages. The management station should not attempt this procedure for any other message type.

Preconditions:
 The management station shall ensure that there is sufficient storage space remaining for the message to be downloaded.

: Management Station

: DMS



(Extended Text Description: Relevant descriptive information provided by author for this figure: Figure 6 is a complex UML sequence diagram which shows the dialog between the management station (on the left) and the device (on the right). It is a sequence of messages between these two devices for defining a message. This is taken directly from the standard NTCIP 1203 V2.29 page 110. For each message there is a horizontal line indicating the message action (SET/GET) and the object included in the message.)

Figure 6 Defining a Message

© 2010 AASHTO / ITE / NEMA
NTCIP 1203 v02.39 Page 140

5.3.7 Monochrome Color Parameter

monochromeColor OBJECT-TYPE
SYNTAX OCTET STRING (SIZE (6))
ACCESS read-only
STATUS mandatory
DESCRIPTION

"<Definition> Indicates the color supported by a monochrome sign. If the 'monochrome1Bit' or 'monochrome8Bit' scheme is used, then this object contains 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). <Object Identifier> 1.3.6.1.4.1.1206.4.2.3.2.7" { vmsCfg 7 }

5.4 FONT DEFINITION OBJECTS

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.

5.4.1 Number of Fonts Parameter

numFonts OBJECT-TYPE
SYNTAX INTEGER (0..255)
ACCESS read-only
STATUS mandatory
DESCRIPTION

"<Definition> Indicates the maximum number of fonts that the sign can store. See the Agency specification in association with the supplemental requirements for fonts to determine the number of fonts that the DMS must support.
<Unit>font
<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
ACCESS not-accessible
STATUS mandatory
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.

--NOTE: 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

ACCESS not-accessible

STATUS mandatory

DESCRIPTION "<Definition>Parameters of the Font Table.

"

INDEX { fontIndex }

::= { fontTable 1 }

Copy Per MIB Distribution Notice © 2010 AASHTO / ITE / NEMA

NTCIP 1203 v02.39 Page 141

FontEntry ::= SEQUENCE{

fontIndex INTEGER,

fontNumber INTEGER,

fontName DisplayString,

fontHeight INTEGER,

fontCharSpacing INTEGER,

fontLineSpacing INTEGER,

fontVersionID INTEGER,

fontStatus INTEGER

}

5.4.2.1 Font Index Parameter

fontIndex OBJECT-TYPE

SYNTAX INTEGER (1..255)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"<Definition> Indicates the row number of the entry.

<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 INTEGER (1..255)

ACCESS read-write

STATUS mandatory

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. A device shall return a badValue error if this value is not unique. <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 DisplayString (SIZE (0..64))
ACCESS read-write
STATUS mandatory
DESCRIPTION
"<Definition> Indicates the name of the font.
<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 INTEGER (0..255)
ACCESS read-write
STATUS mandatory
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.
<Unit>pixel
<Object Identifier> 1.3.6.1.4.1.1206.4.2.3.3.2.1.4"
::= { fontEntry 4 }

© 2010 AASHTO / ITE / NEMA Copy Per MIB Distribution Notice

*A DRAFT Proposed Recommended Standard to the Joint Committee on the
NTCIP*

NTCIP 1201 version v03.13

National Transportation Communications for ITS Protocol

Global Object (GO) Definitions

v03.13 May 2010

This is a proposed recommended standard, 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 AASHTO, ITE, and NEMA. Please ensure that all copies reproduce or distributed contain this notice. This document contains preliminary information that is subject to change.

Published by

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

Institute of Transportation Engineers (ITE)
1627 Eye Street, N.W., Suite 600 Washington, D.C. 20006

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

CONTENTS

Pages

Section 1 GENERAL - 1

- 1.1** Scope - 1
- 1.2** References - 1
 - 1.2.1** Normative References - 1
 - 1.2.2** Other References -1
 - 1.2.3** Contact Information - 2
- 1.3** Terms, Definitions, and Acronyms - 2
- 1.4** Object Tree - 3

Section 2 OBJECT DEFINITIONS [NORMATIVE]- 5

- 2.1** NTCIP Objects - 5
- 2.2** Global Configuration Node - 7
 - 2.2.1** Global Set ID Parameter - 7
 - 2.2.2** Maximum Modules Parameter - 8
 - 2.2.3** Module Table - 8
 - 2.2.4** Base Standards Parameter - 10
- 2.3** Global Database Management Node - 10
 - 2.3.1** Database Creation Transaction - 11
 - 2.3.2** Database Error Type Parameter - 14
 - 2.3.3** Database Error ID Parameter -14
 - 2.3.4** Database Transaction ID Parameter -14
 - 2.3.5** Database Make ID Parameter - 15
 - 2.3.6** Database Verify Status Parameter - 15
 - 2.3.7** Database Verify Error Parameter - 15

- 2.4 Global Time Management Node - 16**
 - 2.4.1 Global Time Parameter - 16**
 - 2.4.2 Global Daylight Saving Parameter - 16**
 - 2.4.3 TimeBase Event Scheduler Node - 17**
 - 2.4.4 Day Plan Parameters - 20**
 - 2.4.5 Global Local Time Differential Parameter - 24**
 - 2.4.6 Standard Time Zone Parameter - 24**
 - 2.4.7 Local Time Parameter - 24**
 - 2.4.8 Daylight Saving Time (DST) Node - 25**
- 2.5 Report Parameter Node - 31**
- 2.6 STMP Object Node - 31**
- 2.7 PMPP Object Node - 31**
 - 2.7.1 Maximum HDLC Group Address Parameter - 31**
 - 2.7.2 HDLC Group Address Table - 31**
- 2.8 Security Node - 33**

© 2010 AASHTO / ITE / NEMA
 NTCIP 1201v03.13 Page 7

--warranties, so the above limitations may not apply to you.
 --Use of these materials does not constitute an endorsement or affiliation by
 --or between AASHTO, ITE, or NEMA and you, your company, or your products and \\
 --services.

--NTCIP is a trademark of AASHTO/ITE/NEMA.

_____*****

—NTCIP OBJECTS

NTCIP1201-v03 DEFINITIONS ::= BEGIN

— NTCIP 8004 v02 Header

--For the purpose of this section, the following OBJECT IDENTIFIERS are used: IMPORTS

OBJECT-TYPE
 FROM RFC-1212

DisplayString
 FROM RFC1213-MIB

devices, Protocols, profiles, global
FROM NTCIP8004v02
Opaque, Counter, Gauge, null
FROM RFC1155-SMI;

— global OBJECT IDENTIFIER { devices 6 } 1.3.6.1.4.1.1206.4.2.6

2.2 GLOBAL CONFIGURATION NODE

globalConfiguration OBJECT IDENTIFIER ::= { global 1 }

-- This node is an identifier used to group all objects for support of
-- configuration functions that are common to most device types.
-- <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.1

2.2.1 Global Set ID Parameter

globalSetIDParameter OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS mandatory

DESCRIPTION

"<Definition> Specifies a relatively unique ID (e.g., this could be a counter, a check-sum, etc.) for all user-changeable parameters of the particular device-type currently implemented in the device. Often this ID is calculated using a CRC algorithm.

This value shall be calculated when a change of any static database object has occurred. The value reported by this object shall not change unless there has been a change in the static data since the last request. If the actual objects, which are to be included to create this object value, are not defined in the actual device-level standard such as 1202 or 1203, then the general guidance is to include all configuration objects that are stored in a type of memory that survives power outages.

A management station can use this object to detect any change in the static database objects by monitoring this value after it has established a baseline.

<Object Identifier> 1.3.6.1.4.1.1206.4.2.6.1.1"

::= { globalConfiguration 1 }

© 2010 AASHTO / ITE / NEMA Copy per MIB Distribution Notice
NTCIP 1201 v03.13 Page 10

2.2.3.6 Module Type Parameter

moduleType OBJECT-TYPE

SYNTAX INTEGER {

other (1),

hardware (2),

software (3) }

ACCESS read-only

STATUS mandatory

DESCRIPTION

"<Definition>This object specifies whether the associated module is a hardware or software module.

<Object Identifier> 1.3.6.1.4.1.1206.4.2.6.1.3.1.6"

::= { moduleTableEntry 6 }

2.2.4 Base Standards Parameter

controllerBaseStandards OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (0..256))

ACCESS read-only
STATUS mandatory
DESCRIPTION

"<Definition>For use in this object, an ASCII string that shall identify all of the standard document numbers that define or reference MIBs upon which the device is based. Where applicable, profiles shall be referenced rather than the base standards. The version string shall be constructed as follows: The acronym of the standards development organization (or other body) that developed and approved the standard; a space; the standards document number; a colon; and the documents version number as designated by the standards development organization (or other body). Separate entries in the list of standards shall be separated by a carriage return (0x0d) and line feed (0x0a). In the case of NTCIP documents prior to formal approval, the version number shall be the version number in the form of lower case 'v' followed by the major version followed by a period followed by the minor revision. In the case of approved NTCIP standards, the publication year shall precede the version number. In the case of amended NTCIP standards, the version number shall be replaced by the four digit year of publication of the published standard followed by the upper case letter 'A', followed by the amendment number.

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

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

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

2.3 GLOBAL DATABASE MANAGEMENT NODE

```
globalDBManagement OBJECT IDENTIFIER ::= { global 2 }
-- <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.2
```

Copy per MIB Distribution Notice © 2010 AASHTO / ITE / NEMA
NTCIP 1201v03.13 Page 11

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

2.3.1 Database Creation Transaction

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

DESCRIPTION

"<Definition>This object provides transaction control for device configuration. The transaction mode changes the behavior of the agent to force buffering of database objects until all related database objects have been modified. In the normal mode, SET operations to any database object shall either be stored in a device's database immediately with no regard to whether other changes will be made or be rejected (as defined in the device-specific Information Profile). In the transaction mode, SET operations to any database object shall be buffered until a verify state performs a consistency check. When the consistency check completes, the device automatically transitions to the done state where a normal or transaction command may be issued. A database object is a user-provided piece of setup information (or it may be defined in an information profile) that is necessary for the proper operation of a device. It is static in nature in that the agent would never change it without direction from the management station. For example, a parameter that defines a default mode of operation would be a database object. A parameter that indicates the current state of the device would not be a database object.

The states and commands are defined as:

NORMAL: SET operations behave as normal SETs and shall have an immediate effect on the value of any database objects used by the device if none of the objects contained in the operation require the use of the transaction mode (as defined in the device-specific Information Profile). A SET operation containing any database object that requires the use of transaction mode shall result in a genErr. This is the default state of this object.

The only command that may be written to dbCreateTransaction while in this state is TRANSACTION. Any other values written to this object in this state shall result in an error response of 'badValue'.

TRANSACTION: A SET operation of one or more database objects that use the same community name as used in the request for the TRANSACTION state are buffered by the agent device for later consistency checks and a normal response is returned. A SET operation of one or more database objects using different community names shall result in a genErr with the index set to

© 2010 AASHTO / ITE / NEMA Copy per MIB Distribution Notice
NTCIP 1201 v03.13 Page 12

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

A SET operation containing both database and non-database objects shall be processed in full according to these two rules. Thus, if it contains the same community name as used in the request for the TRANSACTION state, the non-database objects shall be stored immediately while the database objects shall be buffered. If it uses a different community name, the entire request will be rejected and a genErr with an index of zero shall be returned. GET operations on any object shall return the values of the data stored in the controller and shall ignore any values contained in the buffer.

Any valid community name may read this (dbCreateTransaction) object when in this state, but only the community name used to command the object to the transaction mode and the administrator community name can set this object. A set from any other community name shall result in a genErr with an index of zero. The only commands that can be written to dbCreateTransaction while in this state are VERIFY and NORMAL. A VERIFY command will change the state to VERIFY. If a NORMAL command is received, all buffered data is discarded and the state is returned to NORMAL. Any other values written to this object when in this state shall result in an error response of 'badValue'.

VERIFY: Specific database objects are checked for consistency. When consistency checks are complete the device will automatically advance to the DONE state.

The state of dbCreateTransaction cannot be changed when in the VERIFY state. Any values written to this object in this state shall result in an error response of 'badValue'. The consistency check analyzes certain critical objects 'in context' and treats them as an interrelated whole rather than separate non-related data items. The consistency check rules are not defined in NTCIP 1201 v03, since these are device and implementation specific. Where applicable, the consistency check rules are defined in application specific object definition standards. A specific implementation may add additional checks beyond those defined in NTCIP standards.

A SET operation containing any database objects while in the VERIFY state shall result in a genErr with the index set to zero.

DONE: This state is entered automatically once consistency checks have completed in the VERIFY mode. The value of dbVerifyStatus and dbVerifyError indicate whether the consistency check found any errors.

A SET operation containing any database objects while in the DONE state shall result in a genErr with the index set to zero. Any valid community name may read this (dbCreateTransaction) object when in this state, but only the community name used to command the object to the transaction mode and the administrator community name can set this object. A set from any other community name shall result in a genErr with an index of zero. The only commands that can be written to dbCreateTransaction while in this

Copy per MIB Distribution Notice 31 © 2010 AASHTO / ITE / NEMA
 NTCIP 1201v03.13 Page 13

state are NORMAL and TRANSACTION. Any other values written to this object in this state shall result in an error response of 'badValue'.

If a NORMAL command is issued and dbVerifyStatus indicates doneWithNoError, the buffered data is transferred to the device memory and the state is returned to NORMAL. If a NORMAL command is issued and dbVerifyStatus indicates something other than doneWithNoError then the buffered data is discarded and the state is returned to NORMAL.

If a TRANSACTION command is issued, regardless of dbVerifyStatus, no action takes place (the buffered data is not changed) and the TRANSACTION state is re-entered.

		COMMANDED STATE (9)			
		<i>transaction</i>	<i>verify</i>	<i>normal</i>	<i>done</i>
CURRENT STATE	normal	transaction (1)	normal (2)	normal (2)	normal (2)
	transaction	transaction (2)	verify (3)	normal (4)	transaction (2)
	verify (7)	verify (2)	verify(2)	verify (2)	verify (2)
	done (8)	transaction (5)	done(2)	normal (6)	done (2)

(Additional Notes from the Author: contains a table showing the various states for the database management message exchanges using NTCIP. The columns are labeled transaction, verify, normal, and done. The rows are labeled normal, transaction, verify, and done. The rows indicate the current state, and the columns indicate the commanded state. This table shows that the general sequence for this operation is to first start a transaction (1), then continue in the transaction state downloading data elements until they are all completed after which the commanded state is verify. The management station must monitor the response by checking the dbVerifyStatus awaiting the completion of the transaction; if the transaction is completed without errors, then the new data was accepted; if the transaction was completed with errors, the management station must read the dbVerifyError to determine the type of error experienced.)

Operational procedures and error responses:

1. Once a copy of all database objects is placed in a buffer, the state is changed to transaction and error response indicates noError. If the operation fails, the state remains the same and error response indicates genErr.
2. No action takes place, the state remains the same, but response indicates badValue.
3. The state is changed to verify, a consistency check is started, and response indicates noError. Once the consistency check is completed, the state automatically changes to done.
4. The buffered copy of all database objects is discarded, the state is changed to normal, and response indicates noError.
5. The buffered copy of all database objects is not changed or reloaded, the state is changed to transaction, and response indicates noError.
6. If dbVerifyStatus indicates doneWithNoError, then the copy of all database objects is transferred to memory, the state is changed to normal and response indicates noError. If dbVerifyStatus indicates doneWithError then the buffered data is discarded, the state is changed to NORMAL, and response indicates noError.
7. The state automatically changes to done when the consistency check completes.
8. dbVerifyStatus and dbVerifyError are only valid in this state.
9. All SET operations on this (dbCreateTransaction) parameter shall be made using a protocol that uses a community name, or equivalent field (e.g., SNMP).

<Object Identifier> 1.3.6.1.4.1.1206.4.2.6.2.1"
 DEFVAL {normal}

© 2010 AASHTO / ITE / NEMA Copy per MIB Distribution Notice
 NTCIP 1201 v03.13 Page 14

::= { globalDBManagement 1 }

2.3.2 Database Error Type Parameter

-- This object has been deprecated. See Annex D.1.8 for more information.

dbErrorType OBJECT-TYPE
 SYNTAX INTEGER { tooBig (1),
 noSuchName (2),
 badValue (3),
 readOnly (4),
 genError (5),
 updateError (6),
 noError (7)}
 ACCESS read-only
 STATUS deprecated
 DESCRIPTION

"This object returns the current error status of the transaction. The value of this object is only valid when the dbCreateTransaction object is in the Done or Error state.

<Object Identifier> 1.3.6.1.4.1.1206.4.2.6.2.2"
 ::= { globalDBManagement 2 }

2.3.3 Database Error ID Parameter

-- This object has been deprecated. See Annex D.1.8 for more information.

dbErrorID OBJECT-TYPE
 SYNTAX OBJECT IDENTIFIER
 ACCESS read-only
 STATUS deprecated
 DESCRIPTION

"This object contains the object identifier of the first object in the transaction buffer that caused an error while dbCreateTransaction object was in the Verifying or Updating state. The value of this object is only valid when the dbCreateTransaction object is in the Error state. It is undefined when the dbCreateTransaction object is in other states.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.6.2.3"
 ::= { globalDBManagement 3 }

2.3.4 Database Transaction ID Parameter

-- This object has been deprecated. See Annex D.1.8 for more information.

dbTransactionID OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS deprecated

DESCRIPTION

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

<Object Identifier> 1.3.6.1.4.1.1206.4.2.6.2.4"

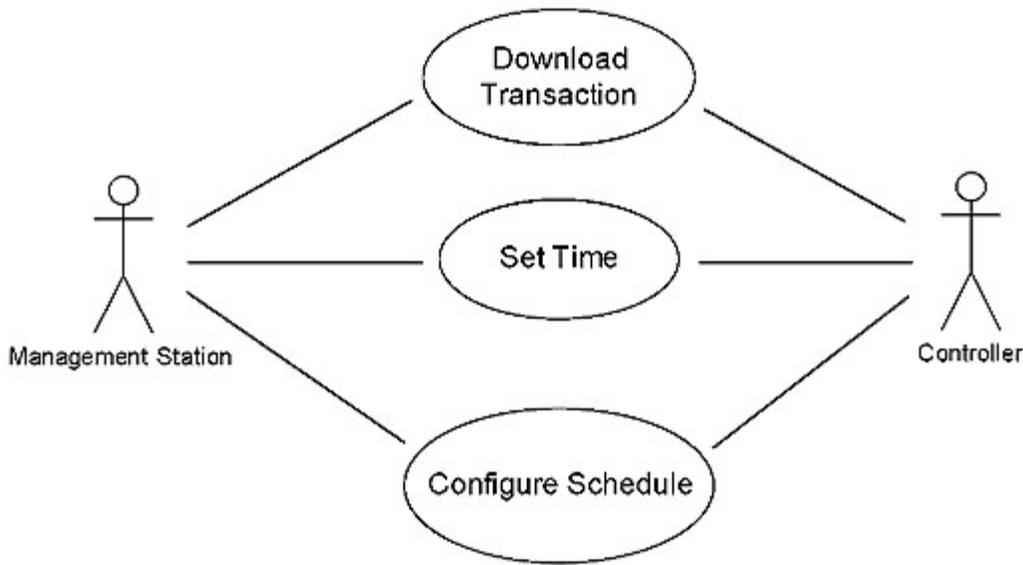
::= { globalDBManagement 4 }

Copy per MIB Distribution Notice © 2010 AASHTO / ITE / NEMA
NTCIP 1201v03.13 Page 45

Annex A CONCEPT OF OPERATIONS [NORMATIVE]

Annex A provides examples of how a management station may interface with a device complying with NTCIP '1201 v03. Any device claiming conformance to the NTCIP 1201 v03 features depicted (download transaction, set time, and configure schedule) shall support the exchanges as shown. However, the flexible design of the NTCIP protocols allows a large number of other possibilities and these figures do not limit any other requirements of these standards. These diagrams promote a common understanding of how systems may be designed to increase the likelihood of interchangeability in deployed systems.

Three use cases are shown in Figure 2.



(*Extended Text Description:* Figure 2: In this diagram, there are two entities represented by stick figures. The entity on the left is labeled "Management Station". The entity on the right is labeled "Controller". In between the two entities are three ovals, arranged one on top of the other in a single column. The top oval is labeled "Download Transaction". Going down, the middle oval is labeled "Set Time". The bottom oval is labeled "Configure Schedule". Each stick person has a line connecting it to each of the three center ovals.)

Figure 2 Global Use Cases

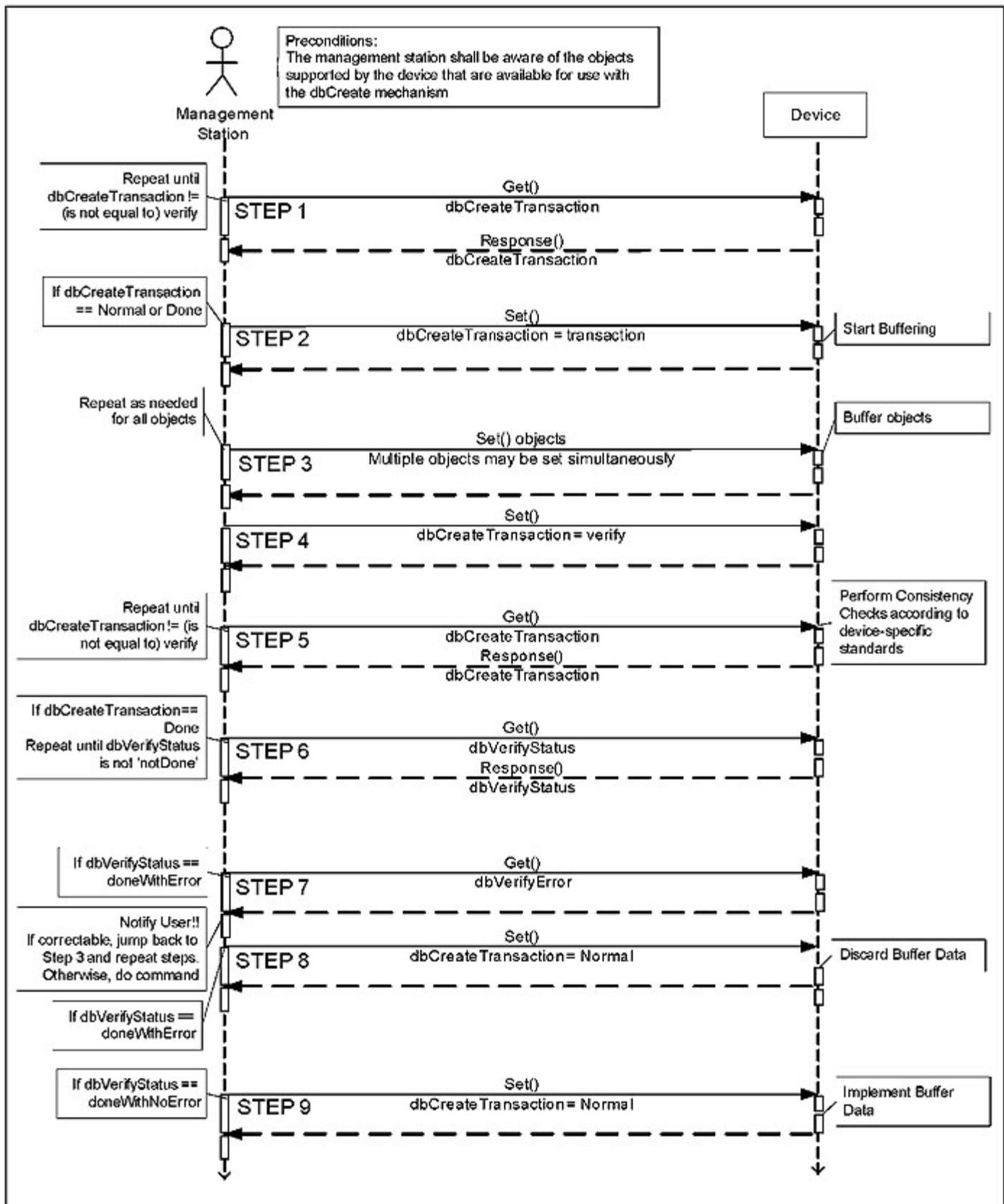
A.1 DOWNLOAD TRANSACTION USE CASE

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

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

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

The normal, fault-free process is in Figure 3.



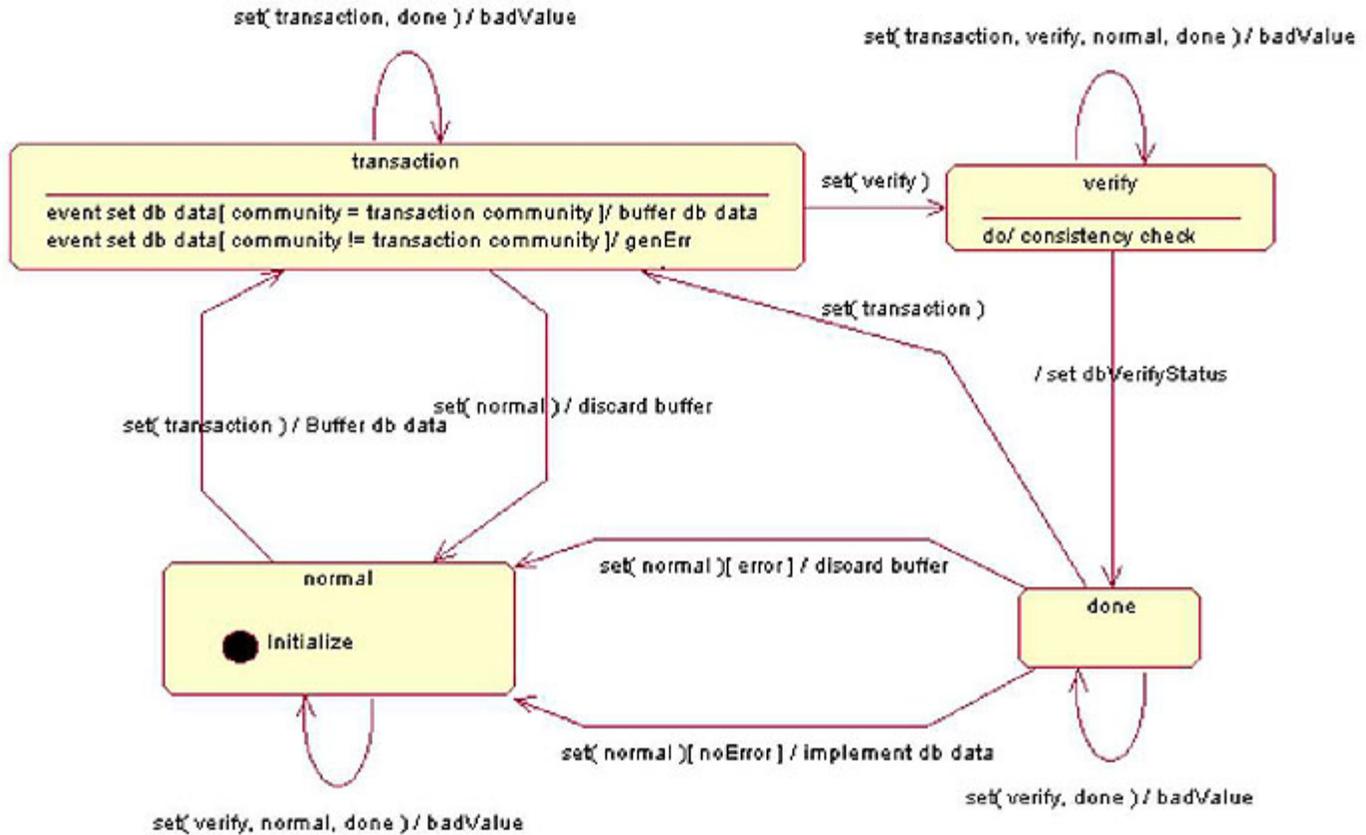
(Extended Text Description: Relevant descriptive information for this figure: Figure 3 is a complex sequence diagram which shows the dialog between the management station (on the left) and the device (on the right). It is a sequence of messages between these two devices for defining a Download Transaction Process Sequence. This is taken directly from the standard NTCIP 1201 v03.13 Page 46. For each message there is a horizontal line indicating the message action (SET/GET) and the object included in the message. The figure is broken into a series of 9 steps, indicating the sequence of SET/GET message actions.)

Figure 3 Download Transaction Process Sequence Diagram

Within this mode, the controller operates as a state machine as described in the definition of dbCreateTransaction. Figure 4 supplements this definition and is a state diagram that provides a formal UML representation of the state machine.

© 2010 AASHTO / ITE / NEMA

NTCIP 1201v03.13 Page 47



(Extended Text Description: Figure 4: This is a state diagram that provides a formal UML representation of the state machine. In this diagram, there are four rectangular boxes arranged in a rectangular formation. Going clockwise, on the upper left is the "Transaction" box, on the upper right is the "Verify" box on the lower right is the "Done" box, and on the lower left is the "Normal" box. There are arrows indicating the process flow of information. Beginning from the "Normal" box, there are two possibilities for process flow: "set(verify, normal, done)/badValue" results in a return loop back to "Normal", or "set(transaction)/Buffer db data) that results in a direct arrow to the "Transaction" box. From the "Transaction" box, there are three possible direction of process flow: "Set(transaction,done)/bad)/badValue results in a return loop arrow back to the "Transaction" box; "Set(verify) results in an arrow to the "Verify" box; "Set(normal)/discard buffer" results in a direct arrow back to the "Normal Box". If the information is sent to the "Verify" box, there are two possible flow paths: "Set(transaction, verify, normal, done)/badValue" results in a return loop arrow back to the "Verify" box; "/set dbVerifyStatus" results in a direct arrow to the done box. If the information is sent to the "Done" box, there are four options for process flow: "Set(verify,done)/badValue" results in a return loop back to "Done", "set(transaction)" results in a direct arrow back to the "Transaction" box, "set(normal)[error]/discard buffer" results in a direct arrow back to "Normal", and "set(normal)[noError]/implement db data" which also results in a direct arrow back to "Normal".)

Figure 4 Controller State Diagram

A.2 SET TIME

The second use case is to set the time in a controller. Four key parameters affect the local time stored in the controller:

- a. globalTime (which is time in UTC)
- b. globalDaylightSaving
- c. begin and end DST objects
- d. controllerStandardTimeZone (which is the offset between local Standard Time and UTC)

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

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

A.2.1 Examples—Operation of the Daylight Saving Time (DST) Mechanism

Figures 5 through 7 and Table 1 further document how DST is to be implemented.

NOTE—The examples shown are unusual programming examples to demonstrate the flexibility of the DST functionality.

© 2010 AASHTO / ITE / NEMA

A Joint Standard of AASHTO, ITE, and NEMA

NTCIP 1202:2005 V02.19

National Transportation Communications for ITS Protocol

Object Definitions for Actuated Traffic Signal Controller (ASC) Units - version 02

November 2005

A major revision of NTCIP 1202:1997 v01.07, and includes the draft 1202 Amendment 1 v06

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

Published by

American Association of State Highway and Transportation Officials (AASHTO)

444 North Capitol Street, N.W., Suite 249 Washington, D.C. 20001

Institute of Transportation Engineers (ITE)

1627 Eye Street, N.W., Suite 600 Washington, D.C. 20006

National Electrical Manufacturers Association (NEMA)

1300 North 17th Street, Suite 1752 Rosslyn, Virginia 22209-3806

© 2005 AASHTO / ITE / NEMA. All rights reserved.

NTCIP 1202:2005 v02.19 Page vi

CONTENTS

Page

Acknowledgements - i

Foreword - ii

Introduction - v

v02.10 Revisions - xiii

v02.11 Revisions -xiv

v02.12 Revisions - xiv

v02.13 Revisions - xv

v02.14 Revisions - xvi

v02.15 Revisions - xvii

v02.16 Revisions - xviii

v02.17 Revisions - xviii

v02.18 Revisions - xix

v02.19 Revisions - xix

Section 1 GENERAL - 1

1.1 Scope - 1

1.2 References -1

1.2.1 Normative References - 1

1.2.2 Other References - 2

1.3 Actuated Controller Unit Terms - 3

1.4 Abbreviations and Acronyms - 7

Section 2 OBJECT DEFINITIONS - 8

2.1 MIB Header - 8

2.2 Phase Parameters - 8

2.2.1 Maximum Phases - 9

2.2.2 Phase Table - 9

2.2.2.1 Phase Number - 10

- 2.2.2.2 Phase Walk Parameter - 10**
- 2.2.2.3 Phase Pedestrian Clear Parameter - 11**
- 2.2.2.4 Phase Minimum Green Parameter - 11**
- 2.2.2.5 Phase Passage Parameter - 11**
- 2.2.2.6 Phase Maximum Green 1 Parameter - 12**
- 2.2.2.7 Phase Maximum Green 2 Parameter - 12**
- 2.2.2.8 Phase Yellow Change Parameter - 13**
- 2.2.2.9 Phase Red Clear Parameter - 13**
- 2.2.2.10 Phase Red Revert - 13**
- 2.2.2.11 Phase Added Initial Parameter - 14**

© 2005 AASHTO / ITE / NEMA
NTCIP 1202:2005 v02.19 Page xi

- 3.3.1 Vehicle Detector Block Example - 114**
- 3.4 Pedestrian Detector Block Data - 115**
 - 3.4.1 Pedestrian Detector Block Example - 115**
- 3.5 Pattern Block Data - 116**
 - 3.5.1 Pattern Block Example - 116**
- 3.6 Split Block Data - 117**
 - 3.6.1 Split Block Example - 117**
- 3.7 Time Base Block Data - 118**
 - 3.7.1 Time Base Block Example - 118**
- 3.8 Preempt Block Data - 119**
 - 3.8.1 Preempt Block Example - 119**
- 3.9 Sequence Block Data - 120**
 - 3.9.1 Sequence Block Example - 120**
- 3.10 Channel Block Data - 121**
 - 3.10.1 Channel Block Example - 121**
- 3.11 Overlap Block Data - 122**
 - 3.11.1 Overlap Block Example - 122**

3.12 Port 1 Block Data - 123

3.12.1 Port 1 Block Example - 123

3.13 Schedule Block Data - 123

3.13.1 sCHEDULE Block Example - 124

3.14 Day Plan Block Data - 124

3.14.1 Day Plan Block Example - 125

3.15 Event Log Config Block Data - 125

3.15.1 Event Log Config Block Example - 126

3.16 Event Class Block Data - 127

3.16.1 Event Class Block Example - 128

3.17 Dynamic Object Config Block Data - 128

3.17.1 Dynamic Object Config Block Example - 129

3.18 Dynamic Object Owner Block Data - 129

3.18.1 Dynamic Object Owner Block Example - 130

3.19 Dynamic Object Status Block Data - 130

3.19.1 Dynamic Object Status Block Example - 131

3.20 Miscellaneous ASC Block Data - 131

3.20.1 Miscellaneous ASC Block Example - 131

Annex A Information Profile (Normative) - 133

A.1 Notation - 134

A.1.1 Type Symbols - 134

A.1.2 Status Symbols - 134

A.1.3 Conditional Status Notation - 134

A.1.4 Support Column - 135

A.2 ASC Requirements - 135

A.3 Phase Conformance Group - 136

A.4 Detector Conformance Group - 137

A.5 Volume Occupancy Report Conformance Group - 138

- A.6 Unit Conformance Group - 138**
- A.7 Special Function Conformance Group - 139**
- A.8 Coordination Conformance Group - 139**
- A.9 Time Base Conformance Group - 140**
- A.10 Preempt Conformance Group - 141**
- A.11 Ring Conformance Group - 142**
- A.12 Channel Conformance Group - 143**
- A.13 Overlap Conformance Group - 143**
- A.14 TS 2 Port 1 Conformance Group - 144**
- A.15 Block Object Conformance Group - 144**

© 2005 AASHTO / ITE / NEMA
NTCIP 1202:2005 v02.19 Page xii

- A.16 Configuration Conformance Group - 144**
- A.17 Database Management Conformance Group - 145**
- A.18 Report Conformance Group - 145**
- A.19 AuxIO Group - 146**
- A.20 PMPP Group - 146**
- A.21 SNMP Group - 146**
- A.22 System Group - 147**
- A.23 SFMP Group - 148**
- A.24 STMP Group - 148**
- A.25 Logical Name Group - 149**
- A.26 Trap Management Group - 149**
- A.27 Security Group - 150**
- A.28 RS232 Group - 150**
- A.29 HDLC Group - 151**
- A.30 Interfaces Group - 152**
- A.31 IP Group - 152**
- A.32 ICMP Group - 154**

A.33 TCP Group - 154

A.34 UDP Group - 155

A.35 Ethernet Group - 155

Annex B CONSISTENCY CHECKS (Normative) - 156

B.1 Consistency Check Rules - 156

B.2 Manufacturer Specific Consistency Checks - 159

Annex C CONCEPT OF OPERATIONS (Informative) - 160

C.1 Get Type 'C' - 'P' - 'S' Objects - 160

C.2 Get Block Data - 160

C.3 Set Type 'C' or 'P' Objects - 161

C.4 Set Type 'P2' Objects - 161

C.5 Set Block Data - 162

C.6 Overlap Supplemental - 163

Annex D DEPRECATED OBJECTS (Normative) - 164

D.1 Special Function Output State - 164

© 2005 AASHTO / ITE / NEMA
NTCIP 1202:2005 V02.19

Page 8

Section 2 OBJECT DEFINITIONS

This section defines those objects which are specifically used by actuated traffic signal controllers. The objects are defined using the OBJECT-TYPE macro specified in RFC 1212. The text provided from Clause 2.1 through the end of the section (except the clause headings) constitutes the NTCIP Standard ASC MIB.

The clauses below present the objects in lexicographical order of their OBJECT IDENTIFIERS which correspond to their physical location within the global naming tree. All of the objects defined in this document reside under the "asc" node of the global naming tree. To aid in object management, the "asc" node has been subdivided into logical categories, each defined by a node under the "asc" node. The individual objects are then located under the appropriate node.

Nodes should not be confused with Conformance Groups, which are defined in Annex A. A Conformance Group is a logical grouping of objects which is used for conformance statements. While Conformance Groups will frequently correspond to the nodal structure, a Conformance Group may contain objects which are not lexicographically ordered. For example, a Schedule Conformance Group contains both "global" and "asc" specific objects.

An object Status of Optional should not be confused with a conformance status of optional or mandatory as defined in Annex A. The object Status of Optional in the MIB means that the object and object definition is current. The status of

optional or mandatory in Annex A dictates whether the object is required or not.

Text preceded by a double hyphen in the MIB definitions represent normative text for this standard.

All management applications shall reference the specific device MIB as provided by the device manufacturer for support and constraints (sub-ranges).

2.1 MIB HEADER

NTCIP1202-200x DEFINITIONS ::= BEGIN

— the following OBJECT IDENTIFIERS are used in the ASC MIB:

IMPORTS

Counter

FROM RFC1155-SMI

OBJECT-TYPE

FROM RFC-1212

OwnerString, devices

FROM TMIB-II;

asc OBJECT IDENTIFIER ::= { devices 1 }

2.2 PHASE PARAMETERS'

phase OBJECT IDENTIFIER

::= { asc 1 }

Copy Per MIB Distribution Notice © 2005 AASHTO / ITE / NEMA

NTCIP 1202:2005 v02.19

Page 9

-- This node shall contain objects that configure, monitor or
-- control phase functions for this device.

2.2.1 Maximum Phases

maxPhases OBJECT-TYPE

SYNTAX INTEGER (2..255)

ACCESS read-only

STATUS optional

DESCRIPTION

"<Definition> The Maximum Number of Phases this Actuated Controller Unit supports. This object indicates the maximum rows which shall appear in the phaseTable object.

<DescriptiveName> NTCIP-1202::ASC.maxPhases

<DataConceptType> Data Element

<Unit> phase"

::= { phase 1 }

2.2.2 Phase Table

phaseTable OBJECT-TYPE

SYNTAX SEQUENCE OF PhaseEntry

ACCESS not-accessible

STATUS optional

DESCRIPTION"

<Definition> A table containing Actuated Controller Unit phase parameters. The number of rows in this table is equal to the maxPhases object.

<TableType> static

<DescriptiveName> NTCIP-1202::ASC.phaseTable

<DataConceptType> Entity Type"

::= { phase 2 }

phaseEntry OBJECT-TYPE

SYNTAX PhaseEntry

ACCESS not-accessible

STATUS optional

DESCRIPTION

"<Definition> Parameters for a specific Actuated Controller Unit phase.

<DescriptiveName> NTCIP-1202::ASC.phaseEntry

<DataConceptType> Entity Type

<Unit> "

INDEX { phaseNumber }

::= { phaseTable 1 }

© 2005 AASHTO / ITE / NEMA

Copy Per MIB Distribution Notice

NTCIP 1202:2005 V02.19 Page 10

```
PhaseEntry ::= SEQUENCE {
    phaseNumber INTEGER,
    phaseWalk INTEGER,
    phasePedestrianClear INTEGER,
    phaseMinimumGreen INTEGER,
    phasePassage INTEGER,
    phaseMaximum1 INTEGER,
    phaseMaximum2 INTEGER,
    phaseYellowChange INTEGER,
    phaseRedClear INTEGER,
    phaseRedRevert INTEGER,
    phaseAddedInitial INTEGER,
    phaseMaximumInitial INTEGER,
    phaseTimeBeforeReduction INTEGER,
    phaseCarsBeforeReduction INTEGER,
    phaseTimeToReduce INTEGER,
    phaseReduceBy INTEGER,
    phaseMinimumGap INTEGER,
    phaseDynamicMaxLimit INTEGER,
    phaseDynamicMaxStep INTEGER,
    phaseStartup INTEGER,
    phaseOptions INTEGER,
    phaseRing INTEGER,
    phaseConcurrency OCTET STRING }
```

2.2.2.1 Phase Number

phaseNumber OBJECT-TYPE

SYNTAX INTEGER (1..255)

ACCESS read-only

STATUS optional

DESCRIPTION

"<Definition> The phase number for objects in this row. This value shall not exceed the maxPhases object value.

<DescriptiveName> NTCIP-1202::ASC.phaseNumber

<DataConceptType> Data Element

<Unit> phase"

::= { phaseEntry 1 }

2.2.2.2 Phase Walk Parameter

phaseWalk OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS optional

DESCRIPTION

"<Definition> Phase Walk Parameter in seconds. This shall control the amount of time the Walk indication shall be displayed.

<DescriptiveName> NTCIP-1202::ASC.phaseWalk

<DataConceptType> Data Element

<Unit> second"

REFERENCE

"NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.2.a"

::= { phaseEntry 2 }

Copy Per MIB Distribution Notice

© 2005 AASHTO / ITE / NEMA

NTCIP 1202:2005 v02.19

Page 11

2.2.2.3 Phase Pedestrian Clear Parameter

phasePedestrianClear OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS optional

DESCRIPTION

"<Definition> Phase Pedestrian Clear Parameter in seconds. This shall control the duration of the Pedestrian Clearance output (if present) and the flashing period of the Don't Walk output.

<DescriptiveName> NTCIP-1202::ASC.phasePedestrianClear

<DataConceptType> Data Element

<Unit> second"

REFERENCE

"NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.2.b"

::= { phaseEntry 3 }

2.2.2.4 Phase Minimum Green Parameter

phaseMinimumGreen OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-write

STATUS optional

DESCRIPTION

"<Definition> Phase Minimum Green Parameter in seconds (NEMA TS 2 range: 1-255 sec). The first timed portion of the Green interval which may be set in consideration of the storage of vehicles between the zone of detection for the approach vehicle detector(s) and the stop line.

<DescriptiveName> NTCIP-1202::ASC.phaseMinimumGreen
 <DataConceptType> Data Element
 <Unit> second"
 REFERENCE
 "NEMA TS 2 Clause 3.5.3.1 and 3.5.3.2.1.a.(1)"
 ::= { phaseEntry 4 }

2.2.2.5 Phase Passage Parameter

phasePassage OBJECT-TYPE
 SYNTAX INTEGER (0..255)
 ACCESS read-write
 STATUS optional
 DESCRIPTION

"<Definition> Phase Passage Parameter in tenth seconds (0-25.5 sec). Passage Time, Vehicle Interval, Preset Gap, Vehicle Extension: the extensible portion of the Green shall be a function of vehicle actuations that occur during the Green interval. The phase shall remain in the extensible portion of the Green interval as long as the passage timer is not timed out. The timing of this portion of the green interval shall be reset with each subsequent vehicle actuation and shall not commence to time again until the vehicle actuation is removed.

© 2005 AASHTO / ITE / NEMA
 Copy Per MIB Distribution Notice
 NTCIP 1202:2005 V02.19
 Page 112

Section 3 BLOCK OBJECT DEFINITIONS

3.1 BLOCK DATA TYPE & ID

All ASC Block Objects shall begin with two octets that define the Data Type and Data ID.

The Data Type octet (ascBlockDataType) provides for the definition of both NTCIP Standard and Device Proprietary data blocks. NTCIP Standard Data Blocks shall utilize an 'ascBlockDataType' of zero. Device Proprietary Data Blocks shall utilize an 'ascBlockDataType' equal to the Private Node Number (PNN) as assigned by NEMA (1.3.6.1.4.1.1206.3.PNN).

dataType	Description
0x00	Standard Data Block
0XPNN	Device Proprietary Data Block

The Data ID octet (ascBlockDataID) provides for definition of included data parameters. NCTIP Standard Data Blocks shall include an 'ascBlockDataID' as listed below:

ascBlockData-dataID Definitions		

dataID	Name	Description
0x00	AscPhaseBlock	Phase Data (see 3.2)
0x01	AscVehDetectorBlock	Vehicle Detector Data (see 3.3)
0x02	AscPedDetectorBlock	Pedestrian Detector Data (see 3.4)
0x03	AscPatternBlock	Pattern Data (see 3.5)
0x04	AscSplitBlock	Split Data (see 3.6)
0x05	AscTimebaseBlock	Time Base Data (see 3.7)
0x06	AscPreemptBlock	Preempt Data (see 3.8)
0x07	AscSequenceBlock	Sequence Data (see 3.9)
0x08	AscChannelBlock	Channel Data (see 3.10)
0x09	AscOverlapBlock	Overlap Data (see 3.11)
0x0A	AscPort1Block	Port 1 Data (see 3.12)
0x0B	AscScheduleBlock	Schedule Data (see 3.13)
0x0C	AscDayPlanBlock	Day Plan Data (see 3.14)
0x0D	AscEventConfigBlock	Event Config Data (see 3.15)
0x0E	AscEventClassBlock	Event Class Data (see 3.16)
0x0F	AscDynObjConfigBlock (*)	Dynamic Obj Config Data (see 3.17)
0x10	AscDynObjOwnerBlock (*)	Dynamic Obj Owner Data (see 3.18)

0x11	AscDynObjStatusBlock (*)	Dynamic Obj Status Data (see 3.19)
0x12	AscMiscBlock	Miscellaneous ASC Data (see 3.20)
0x13-0xFF		Reserved For NTCIP ASC Usage

(*) Any attempt to GET or SET this data via STMP shall result in a genError

New versions of this Standard shall NOT change the structure (content or definition) for any dataID block. New dataID blocks may be added for ascBlockData for expansion to cover other parameters. When a dataID block needs to be revised, the standard writers shall deprecate ascBlockData and establish a new OID (i.e., ascBlockData1) for all the current dataID blocks.

Proprietary Device Blocks shall include an 'ascBlockDataID' as defined in their separate documentation

Copy Per MIB Distribution Notice
 © 2005 AASHTO / ITE / NEMA
 NTCIP 1202:2005 v02.19
 Page 113

3.2 PHASE BLOCK DATA

-- ascBlockData values for standard Block
 -- Phase Data shall be as follows:

```
AscPhaseBlock ::= SEQUENCE
{
  ascBlockDataType INTEGER (0..255), -- 0x00 standard block
  ascBlockDataID INTEGER (0..255), -- 0x00 phase data
  ascBlockIndex1 INTEGER (0..255), -- phaseNumber
  ascBlockQuantity1 INTEGER (0..255), — ## of phases
```

```
-- for {
-- x = ascBlockIndex1;
-- x < (ascBlockIndex1 + ascBlockQuantity1);
-- x++)
```

```
data SEQUENCE OF AscPhaseBlockData
}
```

```
AscPhaseBlockData ::= SEQUENCE
{
  phaseWalk.x INTEGER (0..255),
  phasePedestrianClear.x INTEGER (0..255),
  phaseMinimumGreen.x INTEGER (0..255),
  phasePassage.x INTEGER (0..255),
  phaseMaximum1.x INTEGER (0..255),
  phaseMaximum2.x INTEGER (0..255),
  phaseYellowChange.x INTEGER (0..255),
  phaseRedClear.x INTEGER (0..255),
```

```

phaseRedRevert.x INTEGER (0..255),
phaseAddedInitial.x INTEGER (0..255),
phaseMaximumInitial.x INTEGER (0..255),
phaseTimeBeforeReduction.x INTEGER (0..255),
phaseCarsBeforeReduction.x INTEGER (0..255),
phaseTimeToReduce.x INTEGER (0..255),
phaseReduceBy.x INTEGER (0..255),
phaseMinimumGap.x INTEGER (0..255),
phaseDynamicMaxLimit.x INTEGER (0..255),
phaseDynamicMaxStep.x INTEGER (0..255),
phaseStartup.x INTEGER (1..6),
phaseOptions.x INTEGER (0..65535),
phaseRing.x INTEGER (0..255),
phaseConcurrency.x OCTET STRING
}

```

3.2.1 Phase Block Example

```

-- The following provides an example octet string value for -- a set or get of a phase block.
-- SEQUENCE
-- 00 ascBlockDataType (standard block)
-- 00 ascBlockDataID (phase data)
-- 02 ascBlockIndex1 (start with phaseNumber=2)
-- 02 ascBlockQuantity1 (## of phases=2)

```

© 2005 AASHTO / ITE / NEMA Copy Per MIB Distribution Notice
NTCIP 1202:2005 V02.19 Page 114

```

-- SEQUENCE OF
-- 01 02 quantity of items (ascBlockQuantity1)
-- SEQUENCE # 1 (phaseNumber=2)
-- 06 phaseWalk.2 (6 sec)
-- 0C phasePedestrianClear.2 (12 sec)
-- | etc, etc, to:
-- 01 phaseRing.2 (ring 1)
-- 02 05 06 phaseConcurrency.2 (ph 5 & 6)
-- SEQUENCE # 2 (phaseNumber=3)
-- 00 phaseWalk.3 (0 sec)
-- 00 phasePedestrianClear.3 (0 sec)
-- | etc, etc, to:
-- 01 phaseRing.3 (ring 1)
-- 02 07 08 phaseConcurrency.3 (ph 7 & 8)

```

3.3 VEHICLE DETECTOR BLOCK DATA

```

-- ascBlockData values for standard Block
-- Vehicle Detector Data shall be as follows:

```

```

AscVehDetectorBlock ::= SEQUENCE
{
ascBlockDataType INTEGER (0..255) -- 0x00 standard block
ascBlockDataID INTEGER (0..255), -- 0x01 veh detector data
ascBlockIndex1 INTEGER (0..255), -- vehicleDetectorNumber
ascBlockQuantity1 INTEGER (0..255), -- ## of veh detectors

-- for (

```

```

-- x = ascBlockIndex1;
-- x < (ascBlockIndex1 + ascBlockQuantity1);
-- x++)

data SEQUENCE OF AscVehDetectorBlockData
}

AscVehDetectorBlockData ::= SEQUENCE
{
vehicleDetectorOptions.x INTEGER (0..255),
vehicleDetectorCallPhase.x INTEGER (0..255),
vehicleDetectorSwitchPhase.x INTEGER (0..255),
vehicleDetectorDelay.x INTEGER (0..65535),
vehicleDetectorExtend.x INTEGER (0..255),
vehicleDetectorQueueLimit.x INTEGER (0..255),
vehicleDetectorNoActivity.x INTEGER (0..255),
vehicleDetectorMaxPresence.x INTEGER (0..255),
vehicleDetectorErraticCounts.x INTEGER (0..255),
vehicleDetectorFailTime.x INTEGER (0..255)
}

```

3.3.1 Vehicle Detector Block Example

```

-- The following provides an example octet string value for
-- a set or get of a vehicle detector block.
-- SEQUENCE
-- 00 ascBlockDataType (standard block)
-- 01 ascBlockDataID (veh detector data)
-- 02 ascBlockIndex1 (start with vehicleDetectorNumber=2)
-- 02 ascBlockQuantity1 (## of veh det=2)

```

Copy Per MIB Distribution Notice © 2005 AASHTO / ITE / NEMA
NTCIP 1202:2005 v02.19
Page 133

Annex A INFORMATION PROFILE (Normative)

A conformance group is a basic unit of conformance and is used to specify a collection of related managed objects. The conformance group designation applied to a set of objects provides a systematic means for determining which objects are required to support a function. If a device has multiple functions, a Conformance Group will be defined for each function. Conformance group definitions will be found in the NTCIP Object Definition Standard documents. The Object Definition Standard may define a Conformance Group with objects that are not in lexicographic order and only apply to devices of that type.

The related managed objects of a conformance group may include mandatory and/or optional objects. Mandatory objects within a conformance group shall be implemented. Optional objects shall be implemented only if a defined function of the device requires that particular object.

For example, assume a device implements an asynchronous RS-232 interface. It must implement all the mandatory objects in the asynchronous conformance group of the RS-232 MIB. It would not have to implement the Synchronous Conformance Group of objects unless it also provided a synchronous interface.

Assume also that the Asynchronous Conformance Group has a *CRC error counter* object that is optional. The *CRC error counter* object would not have to be implemented unless the device used CRC checking on the asynchronous interface.

Conformance groups are defined as either mandatory or optional. If a conformance group is mandatory, all of the objects with STATUS "mandatory" that are part of the Conformance Group shall be present for a device to claim conformance to the Conformance Group. If a Conformance Group is optional, all of the objects that are part of the Conformance Group with the STATUS "mandatory" shall be present if the device supports the Conformance Group. Objects with the STATUS "optional" may be supported.

When a table is included in a conformance group, all objects contained in the table are included by reference. This is because a table is defined as a SEQUENCE OF {SEQUENCE}. Thus, all objects listed in the sequence are defined as an integral part of the table. Tables are defined as either mandatory or optional. If a table is mandatory, all of the objects with STATUS "mandatory" shall be present. If a table is optional, all of the objects with the STATUS "mandatory" shall be present if the device supports the table. Objects in the table with the STATUS "optional" may be supported.

© 2005 AASHTO / ITE / NEMA
 Copy Per PRL Reproduction Notice
 NTCIP 1202:2005 V02.19
 Page 134

A.1 NOTATION

The following notations and symbols are used to indicate status and conditional status within this standard.

A.1.1 Type Symbols

The following symbols are used to indicate type:

Symbol	Type
C	Control Object - use of 'dbCreateTransaction' in NTCIP 1201 Clause 2.3.1 shall NOT delay a SET to this object.
P	Parameter Object - use of 'dbCreateTransaction' in NTCIP 1201 Clause 2.3.1 to SET this object is optional. NOTE—The device must support both the normal SNMP SET and a SET via dbCreateTransaction.
P2	Parameter Object - use of 'dbCreateTransaction' in NTCIP 1201 Clause 2.3.1 to SET this object is mandatory. NOTE—The device must NOT allow a normal SNMP SET.
S	Status / Information Object - this object is read only therefore a SET is not permitted.

A.1.2 Status Symbols

The following symbols are used to indicate status:

Symbol	Status
--------	--------

M	Mandatory
M.<n>	Support of every item of the group labeled by the same numeral <n> required, but only one is active at time.
O	Optional
O.<n>	Optional, but support of at least one of the group of options labeled by the same numeral <n> is required
C	Conditional
N/A	Non-applicable (i.e., logically impossible in the scope of the profile)
X	Excluded or prohibited

A.1.3 Conditional Status Notation

The following predicate notations is used:

Notation	Status
"<predicate>: M	Item is conditional on the <predicate>.

The <predicate>: notation means that the Status following it applies only when the feature or features identified by the predicate are supported. In the simplest case, <predicate> is the identifying tag of a single item.

Copy Per PRL Reproduction Notice

© 2005 AASHTO / ITE / NEMA

NTCIP 1202:2005 v02.19

Page 135

A.1.4 Support Column

This section is in the form of a PICS and, therefore, includes a support column. An implementer claims support of an item by circling the appropriate answer (Yes or No) in the support column:

A.2 ASC REQUIREMENTS

The Conformance Group definitions for Actuated Signal Controllers are defined in this clause. An Actuated Signal Controller has multiple functions; thus, Conformance Groups are defined for each function.

The following table lists functional requirements for an Actuated Signal Controller, and asks if the listed features have been implemented.

Ref	Areas	Clause of Profile	Status	Support

A.3	Phase Conformance Group	NTCIP 1202 - 2.2	M	Yes
A.4	Detector Conformance Group	NTCIP 1202 - 2.3	M	Yes
A.5	Volume Occupancy Report Conformance Group	NTCIP 1202 - 2.3	O	Yes / No
-----	Unit Conformance Group	NTCIP 1202-2.4	O	Yes / No
A.7	Special Function Conformance Group	NTCIP 1202 - 2.4	O	Yes / No
A.8	Coordination Conformance Group	NTCIP 1202 - 2.5	O	Yes / No
A.9	Time Base Conformance Group	NTCIP 1202-2.6	O	Yes / No
A.10	Preempt Conformance Group	NTCIP 1202 - 2.7	O	Yes / No
A.11	Ring Conformance Group	NTCIP 1202 - 2.8	O	Yes / No
A.12	Channel Conformance Group	NTCIP 1202-2.9	O	Yes / No
A.13	Overlap Conformance Group	NTCIP 1202 - 2.10	O	Yes / No
A.14	TS 2 Port 1 Conformance Group	NTCIP 1202 - 2.11	O	Yes / No
A.15	Block Object Conformance Group	NTCIP 1202-2.12	O	Yes / No
	Configuration Conformance Group	NTCIP 1201 -2.2	M	Yes
A.17	Database Management Conformance Group	NTCIP 1201 - 2.3	M	Yes
A.18	Report Conformance Group	NTCIP 1201 - 2.5	O	Yes / No
A.19	Auxiliary I/O Group	NTCIP 1201 -2.8	N/A	No
A.20	PMPP Group	NTCIP1201 - 2.6	O	Yes / No
A.21	SNMP Group	rfc1213	M	Yes

A.22	System Group	rfc1213	M	Yes
A.23	SFMP Group	NTCIP 1103 - A.4	NA	No
	STMP Group	NTCIP 1103-A.5	O	Yes / No
A.25	Logical Name Group	NTCIP 1103 - A.6	O	Yes / No
A.26	Trap Management Group	NTCIP 1103 - A.7-9	O	Yes / No
A.27	Security Group	NTCIP 1103-A.10	M	Yes
A.28	RS232 Group	rfc1317	O	Yes / No
A.29	HDLC Group	rfc1381	O	Yes / No
A.30	Interfaces Group	rfd 213	O	Yes / No
A.31	IP Group	rfc1213	O	Yes / No
A.32	ICMP Group	rfc1213	O	Yes / No
A.33	TCP Group	" rfd 213"	O	Yes / No
A.34	UDP Group	rfc1213	O	Yes / No
A.35	Ethernet Group	rfc1643	O	Yes / No

Actuated Signal Controller (ASC) devices shall adhere to the conformance requirements specified in the above as a minimum to claim compliance to this standard. If a device supports the functionality defined within a group, then the device shall implement the functionality in the standard format. Additional objects or groups may be supported without being non-compliant with ASC objects or NTCIP. Minimum and maximum ranges of objects that differ from the values of the object's SYNTAX field may be enforced by an application running on a device.

A device which enforces range limits within the bounds specified by the values of the object's SYNTAX field shall not be categorized as being non-compliant with ASC objects or NTCIP.

A device which supports a subset of objects with enumerated values shall not be categorized as being non-compliant with ASC objects or NTCIP.

A.3 PHASE CONFORMANCE GROUP

The Phase Conformance Group shall consist of the following objects:

PHASE CONFORMANCE GROUP

NTCIP 1202 Clause	Object Name	Object Type	Object Status	Object Support	Allowed Values	Supported Values
2.2	Phase Conformance Group	—	M	Yes	—	—
2.2.1	maxPhases	S	M	Yes	2-255	
2.2.2	phaseTable	--	M	Yes	—	—
	phaseEntry	--	M	Yes	—	—
2.2.2.1	phaseNumber	s	M	Yes	1..255	
2.2.2.2	phaseWalk	P	M	Yes	0-255	
2.2.2.3	phasePedestrianClear	P	M	Yes	0-255	
2.2.2.4	phaseMinimumGreen	p	M	Yes	0-255	
2.2.2.5	phasePassage	P	M	Yes	0-255	
2.2.2.6	phaseMaximum1	P	M	Yes	0-255	
2.2.2.7	phaseMaximum2	p	M	Yes	0-255	
2.2.2.8	phaseYellowChange	P	M	Yes	0-255	
2.2.2.9	phaseRedClear	P	M	Yes	0-255	

2.2.2.10	phase Red Revert	P	O	Yes / No	0-255	
2.2.2.11	phaseAddedInitial	P	M	Yes	0-255	
2.2.2.12	phaseMaximumInitial	P	M	Yes	0-255	
2.2.2.13	phase Time BeforeReduction	P	M	Yes	0-255	
2.2.2.14	phaseCarsBeforeReduction	P	O	Yes / No	0-255	
2.2.2.15	phaseTimeToReduce	P	M	Yes	0-255	
2.2.2.16	phaseReduceBy	P	O	Yes / No	0-255	
2.2.2.17	phaseMinimumGap	P	M	Yes	0-255	
2.2.2.18	phaseDynamicMaxLimit	P	O	Yes / No	0-255	
2.2.2.19	phase DynamicMaxStep	P	O	Yes / No	0-255	
2.2.2.20	phaseStartup	P2	M	Yes	1-6	
	other(1)	--	—	Yes / No	—	—
	phaseNotON(2)	--	—	Yes / No	—	—
	greenWalk(3)	--	—	Yes / No	—	—
	greenNoWalk(4)	--	—	Yes / No	—	—
	yellowChange(5)	--	—	Yes / No	—	—
	redClear(6)	--	—	Yes / No	—	—
2.2.2.21	phaseOptions	P2	M	Yes	0-65535	
	Bit 0 - Enabled Phase	--	—	Yes / No	—	—

	Bit 1 - Automatic Flash Entry Phase	--	—	Yes / No	—	—
	Bit 2 - Automatic Flash Exit Phase	--	—	Yes / No	—	—
	Bit 3 - Non-Actuated 1	--	—	Yes / No	—	—
	Bit 4 - Non-Actuated 2	--	—	Yes / No	—	—
	Bit 5 - Non-Locking Detector Memory	--	—	Yes / No	—	—
	Bit 6 - Min Vehicle Recall	--	—	Yes / No	—	—
	Bit 7 - Max Vehicle Recall	--	—	Yes / No	—	—
	Bit 8 - Ped Recall	--	—	Yes / No	—	—
	Bit 9 - Soft Vehicle Recall	--	—	Yes / No	—	—
	Bit 10 - Dual Entry Phase	--	—	Yes / No	—	—
	Bit 11 - Simultaneous Gap Disable	--	—	Yes / No	—	—
	Bit 12 - Guaranteed Passage	--	—	Yes / No	—	—
	Bit 13 - Actuated Rest In Walk	--	—	Yes / No	—	—
	Bit 14 - Conditional Service Enable	--	—	Yes / No	—	—
	Bit 15 - Added Initial Calculation	--	—	Yes / No	—	—
2.2.2.22	phaseRing	P2	M	Yes	0-255	

2.2.2.23	phaseConcurrency	P2	M	Yes	string	
2.2.3	maxPhaseGroups	S	M	Yes	1-255	
2.2.4	phaseStatusGroupTable	--	M	Yes	---	—
	phaseStatusGroupEntry	--	M	Yes	—	—
2.2.4.1	phaseStatusGroupNumber	S	M	Yes	1-255	
2.2.4.2	phaseStatusGroupReds	S	M	Yes	0-255	
2.2.4.3	phaseStatusGroupYellows	S	M	Yes	0-255	
2.2.4.4	phaseStatusGroupGreens	S	M	Yes	0-255	

Annex B
CONSISTENCY CHECKS
(Normative)

Consistency checks assure that certain critical objects are checked "in context" and treated as interrelated values rather than separate non-related data items.

When data is downloaded to a CU operating in the "transaction" mode, as defined by the dbCreateTransaction object defined in NTCIP 1201, consistency checks shall be performed on downloaded data when the "verify" state is commanded. The consistency checks that shall occur and corresponding error messages are described below. Error messages, if any, may be examined by reading the dbTransactionError object once the CU has entered the "done" mode.

B.1 CONSISTENCY CHECK RULES

The consistency check rule is stated first, followed by the corresponding error message(s).

- Concurrent Phases, as defined by the phaseConcurrency object, must be in a different ring from phaseNumber (phase being defined). The error message indicates one or more defined concurrent phases have the same ring assignment as phaseNumber. The value "xx" corresponds to phaseNumber.

"PHASE xx CONCURRENCY FAULT"

An example: phaseConcurrency.1 (Phase 1 concurrent phases) includes Phase 2 and phaseRing.1 (Phase 1 Ring) equals phaseRing.2 (Phase 2 Ring). An error message of "PHASE 01 CONCURRENCY FAULT" is provided.

- Concurrent Phases, as defined by the phaseConcurrency object, must be mutually concurrent with phaseNumber (phase being defined). The error message indicates one or more defined concurrent phases does not include phaseNumber as a concurrent phase. The value "xx" corresponds to phaseNumber.

"PHASE xx MUTUAL FAULT"

An example: phaseConcurrency.1 (Phase 1 concurrent phases) includes phase 5 and phaseConcurrency.5 (Phase 5 concurrent phases) does not include phase 1. An error message of "PHASE 01 MUTUAL FAULT" is provided.

- Phase Sequences, as defined by the sequenceData object, must include phases only once in a given phase sequence. The error message indicates a phase appears more than once in a phase sequence. The value "xx" corresponds to sequenceNumber for sequenceData.

"SEQ xx SAME PHASE FAULT"

An example: sequenceData.1.1 (Sequence 01 / Ring 1) is 01-02-03-04-01 (Phase 1 appears twice). An error message of "SEQ 01 SAME PHASE FAULT" is provided.

© 2005 AASHTO / ITE / NEMA
NTCIP 1202:2005 v02.19 Page 160

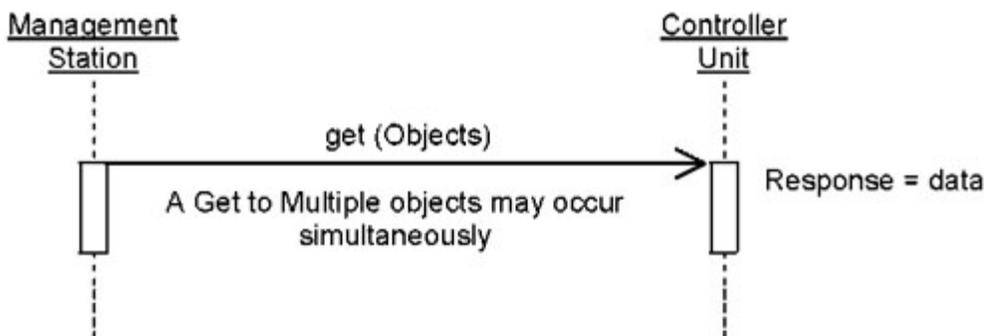
Annex C CONCEPT OF OPERATIONS (Informative)

This Annex provides:

1. Examples of how a management station may interface with an ASC complying with this standard as envisioned by the authors. Any ASC claiming conformance with the subject features depicted in these figures shall support the exchanges as shown. However, the flexible design of the NTCIP protocols allows a large number of other possibilities and these figures do not limit any other requirements of these standards. These diagrams are merely provided to promote a common understanding of how systems may be designed in order to increase the likelihood of interoperability in deployed systems.
2. Supplemental information on overlap sequences based on programming data for 'overlapIncludedPhases' and 'overlapModifierPhases'.

C.1 GET TYPE 'C' - 'P' - 'S' OBJECTS

This use case applies when getting Type 'C' - 'P' - 'S' objects.

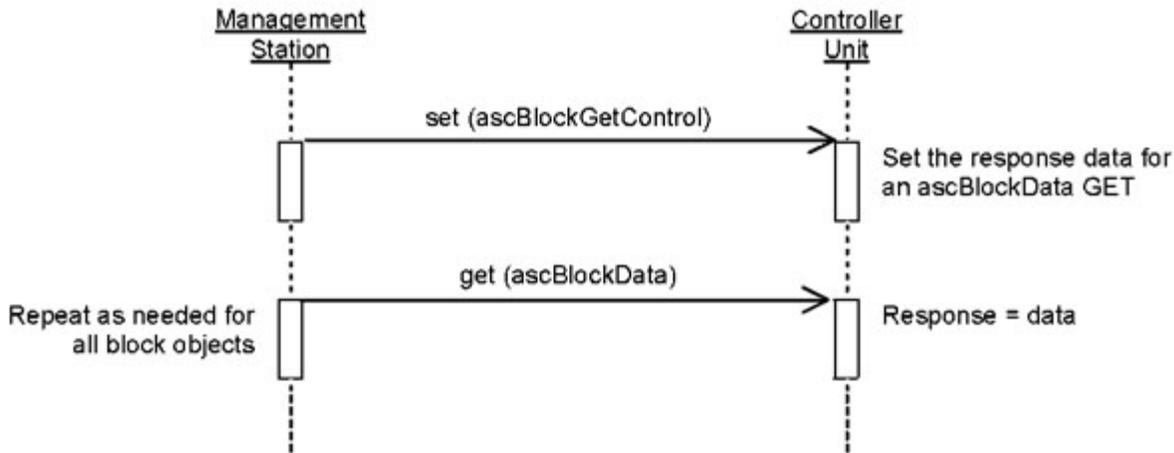


(Extended Text Description: This diagram illustrates the interaction between a "Management Station" on the left, and a "Controller Unit" on the right. From the "Management Station", "get(Objects)", A Get to Multiple objects may occur

simultaneously, is sent to the "Controller Unit", noted with "Response = data.")

C.2 GET BLOCK DATA

This use shall applies when getting block data via object 'ascBlockData'.



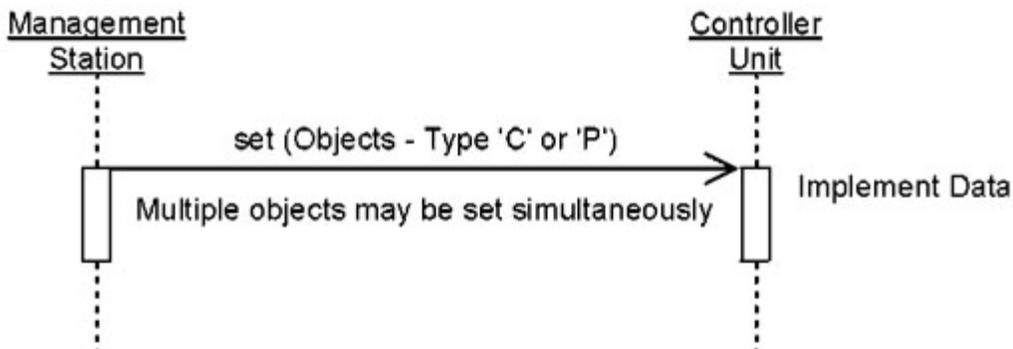
(*Extended Text Description:* This diagram illustrates the interaction between a "Management Station" on the left, and a "Controller Unit" on the right. From the "Management Station", two messages are sent to the "Controller Unit". The top message sent is "set(ascBlockGetControl)" is sent to the "Controller Unit", noted with "Set the response data for an ascBlockData GET". The bottom message sent from the "Management Station" noted with "Repeat as needed for all block objects" sends a "get (ascBlockData)" message to the "Controller Unit", noted with "Response = data".)

© 2005 AASHTO / ITE / NEMA

NTCIP 1202:2005 V02.19 Page 161

C.3 SET TYPE 'C' OR 'P' OBJECTS

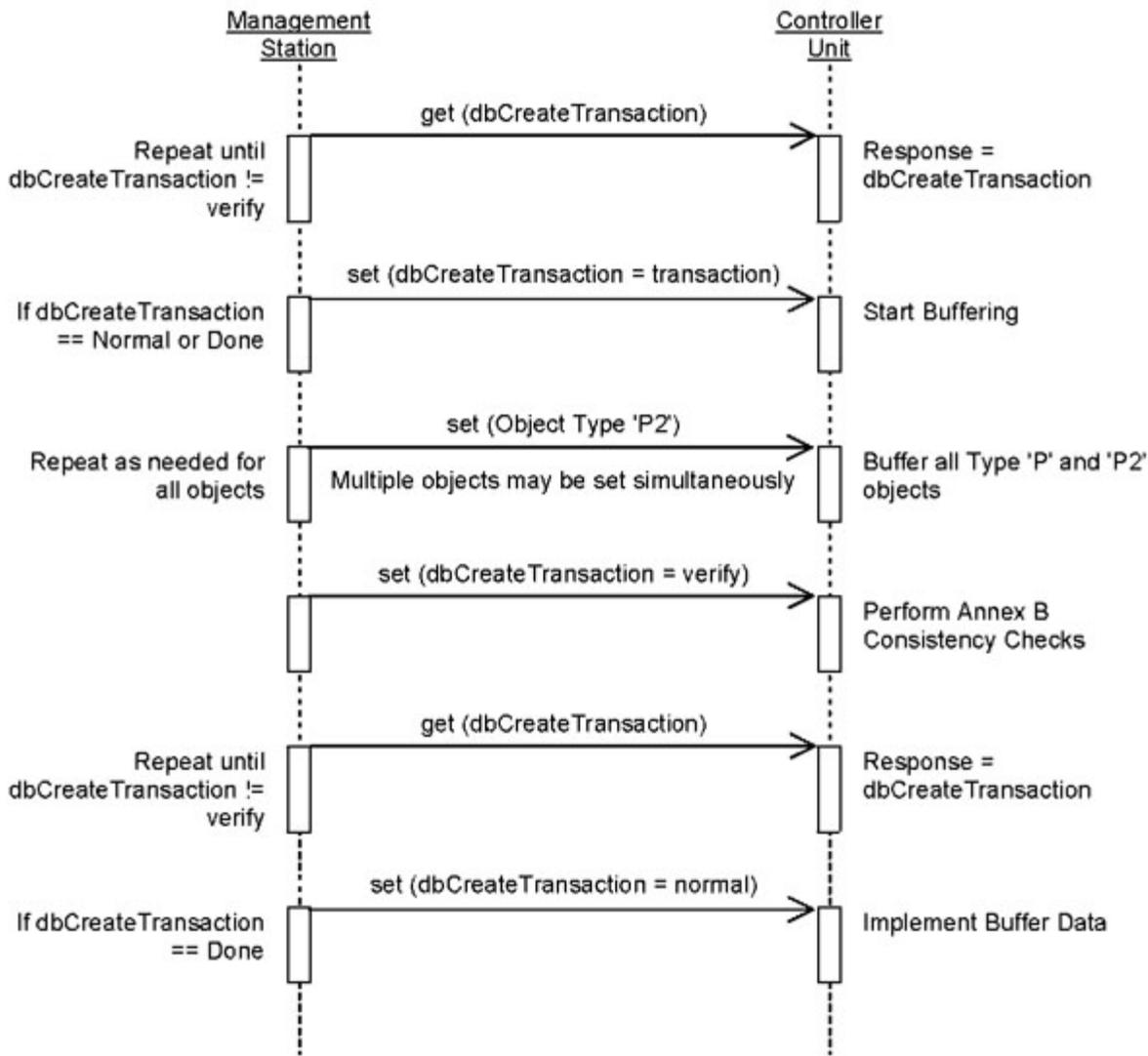
This use case applies when only Type 'C' or 'P' objects are included in the data to be set.



(*Extended Text Description:* This diagram illustrates the interaction between a "Management Station" on the left, and a "Controller Unit" on the right. From the "Management Station", a "set(Objects - Type 'C' or 'P')", where "multiple objects may be set simultaneously", message is sent to the "Controller Unit" noted with "Implement Data".)

C.4 SET TYPE 'P2' OBJECTS

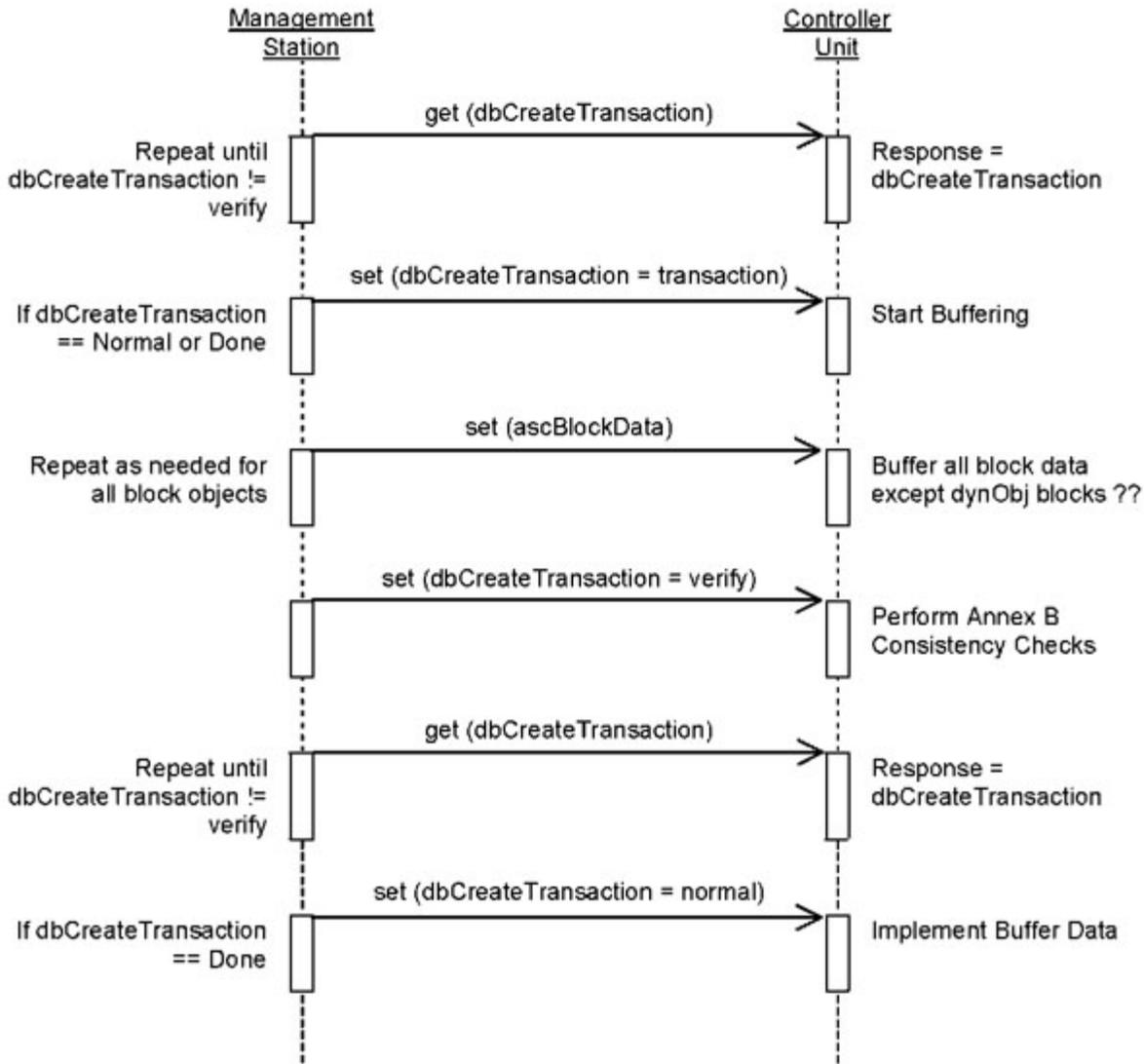
This use case applies when Type 'P2' objects are included in the data to be set.



(*Extended Text Description:* This diagram illustrates the interaction between a "Management Station" on the left, and a "Controller Unit" on the right. From the "Management Station", six messages are sent to the "Controller Unit", indicated by arrows going from the "Management Station" to the "Controller Unit". The first message, "get(dbCreateTransaction)", is noted with "Repeat until dbCreateTransaction !=verify" at the "Management Station", and noted with "Response=dbCreateTransaction" at the "Controller Unit". The second message, noted with "If dbCreateTransaction==Normal or Done" at the "Management Station", is "set(dbCreateTransaction=transaction)", and noted with "Start Buffering" at the "Controller Unit". The third message, "set(Object Type 'P2')", where "multiple objects may be set simultaneously", is noted with "Repeat as needed for all objects" at the "Management Station", and noted with "Buffer all Type 'P' and 'P2' objects" at the "Controller Unit". The fourth message, "set(dbCreateTransaction=verify)", is sent to the "Controller Unit", and noted with "Perform Annex B Consistency Checks". The fifth message, noted at the "Management Station" with "Repeat until dbCreateTransaction !=verify", is "get(dbCreateTransaction)", and is noted with "Response = dbCreateTransaction" and the "Controller Unit". The last message shown, noted with "If dbCreateTransaction ==Done", is "set(dbCreateTransaction = normal)", and noted with "Implement Buffer Data" at the "Controller Unit".)

C.5 SET BLOCK DATA

This use case applies when setting block data via object 'ascBlockData'.



(*Extended Text Description:* This diagram illustrates the interaction between a "Management Station" on the left, and a "Controller Unit" on the right. From the "Management Station", six messages are sent to the "Controller Unit", indicated by arrows going from the "Management Station" to the "Controller Unit". The first message, "get(dbCreateTransaction)", is noted with "Repeat until dbCreateTransaction !=verify" at the "Management Station", and noted with "Response=dbCreateTransaction" at the "Controller Unit". The second message, noted with "If dbCreateTransaction==Normal or Done" at the "Management Station", is "set(dbCreateTransaction=transaction)", and noted with "Start Buffering" at the "Controller Unit". The third message, "set(ascBlockData)", is noted with "Repeat as needed for all block objects" at the "Management Station", and noted with "Buffer all block data except dynObj blocks??" at the "Controller Unit". The fourth message, "set(dbCreateTransaction=verify)", is sent to the "Controller Unit", and noted with "Perform Annex B Consistency Checks". The fifth message, noted at the "Management Station" with "Repeat until dbCreateTransaction !=verify", is "get(dbCreateTransaction)", and is noted with "Response = dbCreateTransaction" and the "Controller Unit". The last message shown, noted with "If dbCreateTransaction ==Done", is "set(dbCreateTransaction = normal)", and noted with "Implement Buffer Data" at the "Controller Unit".)

© 2005 AASHTO / ITE / NEMA

References

ITS

- Standards website: <http://www.standards.its.dot.gov/> - includes a variety of information about ITS standards and the systems engineering process.
- NTCIP Website: <http://www.ntcip.org> - includes links to NTCIP standards, guides and other relevant information to the NTCIP standards. Check the document links. NTCIP standards are freely available. Select the shopping cart and provide the information required to receive free access.
- NTCIP 9001 - NTCIP Guide

IEEE

- IEEE Website: <http://www.ieee.org> - includes all sorts of information about IEEE standards.
- IEEE Std 1362 - Guide for Information Technology—System Definition —Concept of Operations (ConOps) Document
- IEEE Std. 1512.x This is a family of standards for message exchanges with emergency management centers for traffic incident management and includes message sets, data elements, and data frames.

FHWA

- FHWA Systems Engineering Website: <http://www.fhwa.dot.gov/cadiv/segb/>
Systems Engineering Guidebook for Intelligent Transportation Systems Version 3.0 (The "V" Systems Engineering Model): <http://ops.fhwa.dot.gov/publications/seitsguide/seguide.pdf>

ITE

- ITE Web site for TMDD and ATC standards: <http://www.ite.org/standards>

NEMA

- NEMA Web Site for Traffic Management Equipment Standards: <http://www.NEMA.org>
This site contains the TS2 and TS4 standards which can be purchased for a nominal cost.

SAE

- SAE web site for the Advanced Traveler Information Systems (ATIS) messages, data elements and data frames: <http://www.sae.org>. The ATIS standards developed for center-to-center communications, J2354, can be purchased for a nominal cost.