NTCIP 1211 v01

National Transportation Communications for ITS Protocol

Object Definitions for Signal Control and Prioritization

A Joint Standard of AASHTO, ITE, and NEMA

version 01.38

A Joint Standard of AASHTO, ITE, and NEMA

NTCIP 1211 version v01

National Transportation Communications for ITS Protocol

Object Definitions for Signal Control and Prioritization

published in May 2008

Published by

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

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

National Electrical Manufacturers Association (NEMA)
1300 North 17th Street, Suite 1752
Rosslyn, Virginia 22209-3806

file version 1211 v01.38p IHS-2008d

© 2008 AASHTO / ITE / NEMA. All rights reserved.

NOTICES

Copyright Notice

© 2008 by the American Association of State Highway and Transportation Officials (AASHTO), the Institute of Transportation Engineers (ITE), and the National Electrical Manufacturers Association (NEMA). All intellectual property rights, including, but not limited to, the rights of reproduction, translation and display are reserved under the laws of the United States of America, the Universal Copyright Convention, the Berne Convention, and the International and Pan American Copyright Conventions. Except as licensed or permitted, you may not copy these materials without prior written permission from AASHTO, ITE, or NEMA. Use of these materials does not give you any rights of ownership or claim of copyright in or to these materials.

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

PDF File License Agreement

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

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

Data Dictionary and MIB Distribution Permission

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

You may make and/or distribute unlimited copies, including derivative works, of the DD or MIB, including copies for commercial distribution, provided that:
(i) each copy you make and/or distribute contains the citation "Derived from NTCIP 0000 [insert the standard number]. Used by permission of AASHTO / ITE / NEMA.";

(ii) the copies or derivative works are not made part of the standards publications or works offered by other standards developing organizations or publishers or as works-for-hire not associated with commercial hardware or software products intended for field implementation;
(iii) use of the DD or MIB is restricted in that the syntax fields may be modified only to reflect a more restrictive subrange or enumerated values;
(iv) the description field may be modified but only to the extent that: (a) only those bit values or enumerated values that are supported are listed; and (b) the more restrictive subrange is expressed.

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

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

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

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

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

PRL and RTM Distribution Permission

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

(i) you may make and/or distribute unlimited copies, including derivative works of the PRL (then known as a Profile Implementation Conformance Statement ("PICS")) or the RTM, provided that each copy you make and/or distribute contains the citation "Based on NTCIP 0000 [insert the standard number] PRL or RTM. Used by permission. Original text (C) AASHTO / ITE / NEMA.";
(ii) you may not modify the PRL or the RTM except for the Project Requirements column, which is the only column that may be modified to show a products' implementation or the project-specific requirements; and
(iii) if the PRL or RTM excerpt is made from an unapproved draft, add to the citation "PRL (or RTM) excerpted from a draft document containing preliminary information that is subject to change."

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

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

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

Content and Liability Disclaimer

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

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

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

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

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

Trademark Notice

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

Do Not Copy Without Written Permission © 2008 AASHTO / ITE / NEMA

ACKNOWLEDGEMENTS

NTCIP 1211 v01 was prepared by the NTCIP Signal Control Priority Working Group (SCP WG), which is a subdivision of the Joint Committee on the NTCIP. The Joint Committee is organized under a Memorandum of Understanding among the American Association of State Highway and Transportation Officials (AASHTO), the Institute of Transportation Engineers (ITE), and the National Electrical Manufacturers Association (NEMA). The Joint Committee on the NTCIP consists of six representatives from each of the standards organizations, and provides guidance for NTCIP development.

At the time NTCIP 1211 v01 was prepared, the following individuals were active members of the NTCIP SCP WG:

- Ron Atherley (Co-chair)
- Kevin Balke
- Robert De Roche
- Gary Duncan
- Larry Head
- Bud Kent
- Duana Love (Co-chair)
- Paul Olson
- Peter Ragsdale
- Sonja Sun

Other individuals providing input to NTCIP 1211 v01 include:

- Robert Antonisse
- Bill Barlow
- Jack Brown
- James Cheeks
- George Chen
- Paul Chiu
- Paul Cho
- John Collura
- Ron De Leon
- Gerry DeCamp
- Casey Emoto
- Roelof Engelbrecht
- George Fraizier
- Curtis Herrick
- Mark Hillje
- Fred Housman
- Miomir Ivanovic
- Bill Kronenberger
- Irin Limargo
- Howard Long
- Alex Lopez
- Mark Madden
- Len Madsen
- Andrew Mao
- Steven Mitchell
- Bradley Mizuno
- Ken Montgomery
- Linda Mott
- Beth Neilson
- William Petit
- David Phillips
- Amad Rastegarpour
- Mike Roman
- Jeorg (Nu) Rosenbohm
- Willie Rotich
- Vicki Schofield
- Bruce Schopp
- Ed Seymour
- Ray Starr
- Bo Strickland
- Dale Thompson
- Tom Urbanik
- Susan Walker

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

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

- U.S. Department of Transportation, Research and Innovative Technology Administration
- 3M Intelligent Transportation Systems
- ARINC, Inc.
- BiTrans
- Caltrans
- City of Portland, Oregon
- City of Redmond
- City of Vancouver
- Community Transit (Snohomish County)
- Commuter Associates
- Consolidated Traffic Controls
- Eagle Traffic Control Systems
- Eberle Design, Inc.
- Econolite Control Products, Inc
- Florida DOT
- Gannett Fleming
- Georgia DOT
- Highway Rail Intersection WG of the Rail Transit Vehicle Interface Standards Committee
- Houston Metro
- Ice & Associates
- Institute of Electrical and Electronics Engineers, Inc.
- KDD & Associates
- King County DOT – Metro Transit
- Minnesota Department of Transportation
- Orbital TMS
- PB Farradyne, a Division of PBQD, Inc.
- Peek Traffic Systems, Inc.
- RTA Illinois
- Robert De Roche Consulting, Inc.
- Santa Clara Valley Transportation Authority
- Siemens ITS
- Teleride Innovative Systems, Inc.
- Texas Transportation Institute
- Trevilon Corp.
- University of Tennessee
- Wilbur Smith Associates

Do Not Copy Without Written Permission © 2008 AASHTO / ITE / NEMA

FOREWORD

This NTCIP standards publication uses only metric units.

This NTCIP standards publication defines the management information base for Signal Control and Prioritization (SCP) Systems. This NTCIP standard defines individual parameters that represent the configuration, status, and control information that is unique to an SCP. This NTCIP standard also defines specific groupings of these parameters and others to address the operational configuration, monitoring, and control of the device/entity in a baseline system configuration.

There are five annexes to this NTCIP standard. Annex A through Annex D are normative, and Annex E is informative.

This is an NTCIP Device Data Dictionary standards publication. NTCIP Device Data Dictionary standards define management information in terms of objects (data elements, data frames, and messages) for use within NTCIP systems. A Joint NTCIP Device Data Dictionary standards publication is equivalent to these document types at the standards organizations:

AASHTO – Standard Specification
ITE – Software Standard
NEMA – Standard

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

User Comment Instructions

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

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

A User Comment should be submitted to this address:

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

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

A User Comment should be submitted in the following form:

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

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

Approvals

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

AASHTO – Standard Specification; December 2005
ITE – Software Standard; October 2005
NEMA – Standard; November 2004

History

The technical specifications of NTCIP 1211 are identical to the former reference, except as noted in the development history below:

NTCIP 1211 version 01.25, July 2002 – Accepted as a User Comment Draft by the Joint Committee on the NTCIP. NTCIP Standards Bulletin B0079 sent v01.26 to AASHTO, ITE, and NEMA to solicit user comments.

NTCIP 1211 v01.36, February 2004 – Accepted as a Recommended Standard by the Joint Committee on the NTCIP. NTCIP Standards Bulletin B0096 sent v01.37b to AASHTO, ITE, and NEMA for balloting and approval. Approved by AASHTO in December 2005, approved by ITE in October 2005, and approved by NEMA in November 2004.

NTCIP 1211 v01.38, October 2006 to May 2007 – Prepared standard for publication; edited format, front matter, and style.

Compatibility of Versions

All NTCIP Standards Publications have a major and minor version number for configuration management. The version number syntax is "v00.00a," with the major version number before the period, and the minor version number and edition letter (if any) after the period.

Anyone using this standard should seek information about the version number that is of interest to them in any given circumstance. The MIB, the PRL, and the PICS should all reference the version number of the standards publication that was the source of the excerpted material.

Compliant systems based on later, or higher, version numbers MAY NOT be compatible with compliant systems based on earlier, or lower, version numbers. Anyone using this standard should also consult NTCIP 8004 for specific guidelines on compatibility.

Do Not Copy Without Written Permission © 2008 AASHTO / ITE / NEMA

INTRODUCTION

NTCIP 1211 v01 defines management information for SCP. SCP management information includes individual parameters that represent the configuration, status, and control information of such devices. SCP management information also includes data frames and message sets of these parameters, and others from different standards, to address the operational information exchanges between the devices in a baseline system configuration.

This NTCIP standard defines requirements that are applicable to an NTCIP environment that involves the control of traffic signal controllers. While the term Signal Control and Prioritization implies a logical implementation, the data concepts may be applicable to a physical device. By definition, SCP operates in the context of a "system" that includes: a priority requester (such as transit vehicle), traffic signal controllers, and the management centers (which configure and monitor them). NTCIP 1211 v01 imposes functional and communications requirements on all of these "system" components.

The following keywords apply to this standard: NTCIP, traffic signal control, transit, emergency responder, priority control, prioritization.

In 1992, the NEMA 3-TS Transportation Management Systems and Associated Control Devices Section began development of the NTCIP. The Transportation Section's purpose was in response to user needs to include standardized systems communication in the NEMA TS 2 standard, *Traffic Controller Assemblies*. Under the guidance of the Federal Highway Administration's NTCIP Steering Group, the NEMA effort was expanded to include the development of communications standards for all transportation field devices that could be used in an Intelligent Transportation Systems (ITS) network.

In September 1996, an agreement was executed among AASHTO, ITE, and NEMA to jointly develop, approve, and maintain the NTCIP standards. The Joint Committee on the NTCIP formed a working group to develop a common set of management information to control the roadside devices that act in supervisory capacity. The SCP WG first met in November 1999.

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

< This page is intentionally left blank. >

Do Not Copy Without Written Permission © 2008 AASHTO / ITE / NEMA

CONTENTS

Section 1 GENERAL 1-1
1.1 Scope 1-1
1.2 References 1-1
1.2.1 Normative References 1-1
1.2.2 Other References 1-2
1.2.3 Contact Information 1-2
1.3 Terms 1-2
1.4 Abbreviations and Acronyms 1-4
1.5 SCP Object Tree 1-5

Section 2 FUNCTIONAL REQUIREMENTS 2-1
2.1 Request for Priority Service Use Cases 2-1
2.2 Management Use Cases 2-11
2.3 Priority Request and Response Dialogs 2-12
2.3.1 Priority Request Generator – Priority Request Server Exchanges 2-13
2.3.2 Priority Request Server – Coordinator Exchanges 2-14
2.3.3 Configuration and Monitoring Dialogs 2-14
2.3.4 Traffic Management – Priority Request Server Exchanges 2-16
2.3.5 Traffic Management – Coordinator Exchanges 2-16
2.4 Priority Request Server Functions 2-16
2.4.1 Receipt of Priority Request 2-16
2.4.2 Receipt of Priority Update 2-16
2.4.3 Receipt of Priority Status Control 2-17
2.4.4 Receipt of Priority Status Buffer 2-17
2.4.5 Receipt of Priority Cancel 2-17
2.4.6 Receipt of Priority Clear 2-18
2.4.7 Automatic Priority Clear 2-18
2.4.8 Reservice Processing 2-18
2.4.9 Class Type Override Processing 2-18
2.4.10 Prioritization Processing 2-18
2.4.11 Priority Request Server – Coordinator Synchronization and Timing 2-19
2.4.12 State Transition Diagram 2-21
2.4.13 Clock Synchronization 2-23
2.5 Coordinator Functions 2-23
2.5.1 Get Service Request 2-23
2.5.2 Processing Service Request 2-23
2.5.3 Cancel Processing 2-24
2.5.4 Class Type Override Processing 2-24
2.5.5 Set Service Status 2-24
2.6 Event Log Processor 2-25
2.6.1 Priority Request Server Logging 2-25
2.6.2 Coordinator Logging 2-25
2.6.3 Fleet Management and Traffic Management Logging 2-25
2.7 DataBase Management Processing 2-25
2.7.1 Management of Priority Request Generator 2-25
2.7.2 Management of Priority Request Server 2-25
2.7.3 Management of Coordinator 2-25

Section 3 OBJECT DEFINITIONS FOR PRIORITY REQUEST SERVER 3-1
3.1 Priority Request Server MIB 3-1
3.1.1 Priority Request Table 3-1
3.1.2 Priority Request Server Busy 3-7

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

3.1.3 Priority Request Time to Live Value 3-7
3.1.4 Priority Request Reservice Timer 3-7
3.1.5 Priority Request Reservice Class 1 Time 3-8
3.1.6 Priority Request Reservice Class 2 Time 3-8
3.1.7 Priority Request Reservice Class 3 Time 3-8
3.1.8 Priority Request Reservice Class 4 Time 3-8
3.1.9 Priority Request Reservice Class 5 Time 3-9
3.1.10 Priority Request Reservice Class 6 Time 3-9
3.1.11 Priority Request Reservice Class 7 Time 3-9
3.1.12 Priority Request Reservice Class 8 Time 3-9
3.1.13 Priority Request Reservice Class 9 Time 3-10
3.1.14 Priority Request Reservice Class 10 Time 3-10
3.2 Priority Request Server Block Objects 3-10
3.2.1 Priority Request Message 3-10
3.2.2 Priority Update Message 3-10
3.2.3 Priority Status Control Message 3-11
3.2.4 Priority Status Buffer Message 3-11
3.2.5 Priority Cancel Message 3-11
3.2.6 Priority Clear Message 3-12
3.2.7 Program Data 3-12

Section 4 OBJECT DEFINITIONS FOR COORDINATOR 4-1
4.1 Coordinator MIB 4-1
4.2 Priority Strategy Parameters 4-1
4.2.1 Maximum Service Requests to Consider 4-2
4.2.2 Priority Strategy Requested Table 4-2
4.2.3 Coodinator Busy 4-5
4.2.4 Priority Strategies Max 4-5
4.2.5 Priority Strategies Table 4-5
4.2.6 Priority Strategy Extension to Split Table 4-7
4.2.7 Priority Strategy Default Coordination Pattern 4-8
4.2.8 Priority Service Request and Response Block Objects 4-8
4.3 SCP Block Objects 4-10
4.3.1 SCP Block Get Control 4-10
4.3.2 SCP Block Data 4-11
4.3.3 SCP Block Error Status 4-12
4.4 Coordination Processor Block Object Definitions 4-13
4.4.1 Coordinator Strategy Plan Block Definition 4-13
4.4.2 Coordinator Strategy Splits Block Object Definition 4-14

Annex A PRIORITY REQUEST SERVER INFORMATION PROFILE REQUIREMENTS LIST A-1
A.1 Introduction A-1
A.2 Notation A-2
A.2.1 Type Symbols A-2
A.2.2 Status Symbols A-2
A.2.3 Conditional Status Notation A-2
A.2.4 Support Column A-3
A.3 Priority Request Conformance Groups A-3
A.4 Priority Request Server Conformance Group A-4
A.4.1 Priority Request Server Message Conformance Group A-4
A.5 Configuration Conformance Group A-5
A.6 Database Management Conformance Group A-6
A.7 Report Conformance Group A-6
A.8 AuxIO Group A-7
A.9 PMPP Group A-7

 © 2008 AASHTO / ITE / NEMA

A.10 SNMP Group A-7
A.11 System Group A-8
A.12 SFMP Group A-8
A.13 STMP Group A-8
A.14 Logical Name Group A-9
A.15 Trap Management Group A-10
A.16 Security Group A-10
A.17 RS232 Group A-11
A.18 HDLC Group A-12
A.19 Interfaces Group A-13
A.20 IP Group A-13
A.21 ICMP Group A-14
A.22 TCP Group A-15
A.23 UDP Group A-16
A.24 Ethernet Group A-16
A.25 Time Management Conformance Group A-17

Annex B COORDINATOR INFORMATION PROFILE REQUIREMENTS LIST B-1
B.1 Introduction B-1
B.2 Notation B-2
B.2.1 Type Symbols B-2
B.2.2 Status Symbols B-2
B.2.3 Conditional Status Notation B-2
B.2.4 Support Column B-3
B.3 Coordinator Conformance Groups B-3
B.4 Priority Strategy Conformance Group B-4
B.5 SCP Block Object Conformance Group B-5

Annex C SCP DIALOGS C-1
C.1 Control Dialogs C-1
C.1.1 Priority Request Server Control C-1
C.1.2 Coordinator Control C-2
C.2 Configuration, Monitoring, and Logging Dialogs C-3
C.2.1 Priority Request Server Configuration, Monitoring, and Logging C-4
C.2.2 Coordinator Configuration and Monitoring C-7

Annex D CONSISTENCY CHECKS D-1
D.1 Consistency Check Rules D-1

Annex E SCP TUTORIAL E-1

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

< This page is intentionally left blank. >

Do Not Copy Without Written Permission © 2008 AASHTO / ITE / NEMA

Section 1
GENERAL

1.1 SCOPE

The scope of NTCIP 1211 v01 is the definition of management information and the operations related to Signal Control and Prioritization (SCP). It covers the management of multiple requests for priority (or preferential) treatment of different classes of vehicles (such as transit, emergency service, commercial fleet vehicles, etc.) and the implementation of special coordination operation within a Traffic Signal Controller. In order to affect a priority request, its scope includes the content of messages to be exchanged between a prioritizing entity and a controller, the sequence in which they are exchanged, and associated functions within a controller related to SCP. The scope includes the configuration and monitoring of a prioritizing entity and those aspects of coordination that relate to SCP. The process of generating of priority requests is defined in other standards.

NTCIP 1211 v01 primarily deals with granting priority while still maintaining coordination with adjacent intersections. The functionality expressed here, is intended to work in conjunction with the coordination object definitions and functions as defined in NTCIP 1202 – Object Definitions for Actuated Signal Controllers. The coordination objects and functions can also be invoked when an intersection is operating in a non-coordinated mode.

Granting absolute priority irrespective of coordination is considered pre-emption and is covered in NTCIP 1202 v02. The process of generating a priority request is also not covered here. For transit vehicles, generating a priority request is defined in APTA TCIP-S-01 v2, in the section on TCIP Signal Control and Priority.

1.2 REFERENCES

For revision information on this NTCIP standard, contact:

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

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

1.2.1 Normative References

The following standards (normative references) contain provisions that, through reference in this text, constitute provisions of this NTCIP standard. Other documents and standards (other references) which are referenced in NTCIP 1211 v01 might provide a complete understanding of the entire protocol and the relations between all parts of the protocol. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standard listed below.

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

AASHTO / ITE / NEMA NTCIP 1201 v02	*NTCIP Global Object (GO) Definitions – version 02* Version v02 Jointly Approved in 2005; published in October 2005.
AASHTO / ITE / NEMA NTCIP 1202 v02	*NTCIP Object Definitions for Actuated Traffic Signal Controller (ASC) Units – version 02* Version v02 Jointly Approved in 2005; published in November 2005.
AASHTO / ITE / NEMA NTCIP 2301 v01	*Application Profile for Simple Transportation Management Framework* Version v01 Jointly Approved in 2001; published in December 2001.
APTA TCIP-S-01 version 2 Volumes I, II, III, and IV	*Transit Communications Interface Profiles (TCIP)* (the former NTCIP 1400-series)

1.2.2 Other References

AASHTO / ITE / NEMA NTCIP 1103 v01	*Transportation Management Protocols*
AASHTO / ITE / NEMA NTCIP 8004 v01	*Structure and Identification of Management Information*
AASHTO / ITE / NEMA NTCIP 9001 v03	*The NTCIP Guide – version 03*
NEMA TS 2-2003 v02	*Traffic Controller Assemblies with NTCIP Requirements*

1.2.3 Contact Information

1.2.3.1 APTA Standards

APTA standards may be obtained from:

American Public Transportation Association
1666 K Street, N.W., Suite 1100
Washington, DC 20006
(202) 496-4800
www.aptastandards.com

1.2.3.2 NEMA Standards

NEMA standards may be obtained from:

National Electrical Manufacturers Association
1300 North 17th Street, Suite 1752
Rosslyn, VA 22209-3806
www.nema.org

Do Not Copy Without Written Permission © 2008 AASHTO / ITE / NEMA

1.3 TERMS

The SCP WG is viewing the development of NTCIP 1211 v01 as an opportunity to also create a standard set of terms for SCP related equipment that is currently labeled and defined differently by various users, depending upon their geographical location or functional domain. It is hoped that the proposed standard terminology would be used in the development of other standards, which are currently needed, but have not been addressed.

For the purpose of NTCIP 1211 v01, the following terms and definitions apply. For terms not defined in this section, English words are used in accordance with their definitions in the latest edition of *Webster's New Collegiate Dictionary*. Electrical and electronic terms not defined in this section or in *Webster's New Collegiate Dictionary* are used in accordance with their definitions in IEEE Std 100-1992, *The Authoritative Dictionary of IEEE Standards Terms*.

coordination

The control (operation) of (traffic signal) controller units in a manner to provide a relationship between specific green indications at adjacent intersections in accordance with a time schedule to permit continuous operation (movement) of groups of vehicles along the street at a planned speed. The parenthetical expressions are added to the definition as it appears in NTCIP 1202 v02.

data element

A single unit of information of interest (such as a fact, proposition, observation, etc.) about some (entity) class of interest (e.g., a person, place, process, property, concept, association, state, event). A data element is considered indivisible in a particular context. [Per NTCIP 8004 version v01 and as defined in ISO 14817]

NOTE — A data element is represented by an object class, a property of the represented object class and a value domain.
NOTE — In the context of NTCIP 1211, the SNMP Object Type Macro is used to define the entity type, property, and an explicit value domain term.

data frame

A grouping of data elements primarily for the purpose of referring to the group with a single name, and thereby efficiently reusing groups of data elements that commonly appear together (e.g., as an ASN.1 SEQUENCE, SEQUENCE OF, SET, SET OF or CHOICE) in a message specification. [Per NTCIP 8004 v01 and as defined in ISO 14817]

NOTE — This data concept type may be used to specify groups of data elements for other purposes as well.
NOTE — In the context of SNMP, a data frame consists of only the information (variable bindings) to be exchanged. It does not include the application layer header.

fleet management

The supervisory functions related to planning, operation, control, and maintenance of fleet vehicles such as buses.

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

Management Information Base	A structured collection or database of related managed objects defined using ASN.1. A Management Information Base may refer to all managed objects within an implementation or a subset that are specifically to some type of functionality.
message	A grouping of data elements and/or data frames, as well as associated message metadata, that is used to convey a complete unit of information. [Per NTCIP 8004 Version 1 and as defined in ISO 14817] NOTE — For the purposes of NTCIP 1211, a message is an abstract description; it is not a specific instance. NOTE — In the context of SNMP, a message consists of the application layer header and the information (variable binding) to be exchanged.
preemption	The transfer of the normal control (operation) of traffic signals to a special signal control mode for the purpose of servicing railroad crossings, emergency vehicle passage, mass transit vehicle passage, and other special tasks, the control of which requires terminating normal traffic control to provide the service needs of the special task. [Per NTCIP 1202 v02]
priority	The preferential treatment of one vehicle class (such as a transit vehicle, emergency service vehicle or a commercial fleet vehicle) over another vehicle class at a signalized intersection without causing the traffic signal controllers to drop from coordinated operations. Priority may be accomplished by a number of methods including the beginning and end times of greens on identified phases, the phase sequence, inclusion of special phases, without interrupting the general timing relationship between specific green indications at adjacent intersections.
priority request	The information that describes a need for priority service based upon user-defined criteria (such as the number of minutes behind schedule, vehicle occupancy levels, vehicle class, etc.). A priority request is sent from a Priority Request Generator to a Priority Request Server.
Priority Request Generator	A logical or physical entity that initiates a priority request.
Priority Request Server	A logical or physical entity that manages and prioritizes one or more priority requests and generates one or more service requests.
service request	The information that describes a priority service that is to be processed by the Coordinator within a Traffic Signal Controller. A service request is sent from a Priority Request Server to a Traffic Signal Controller.
traffic management	The supervisory functions related to planning, operation, control, and maintenance of traffic control devices.

Do Not Copy Without Written Permission © 2008 AASHTO / ITE / NEMA

traffic control assembly	The collection of components that reside in a traffic signal control cabinet. In addition to a traffic signal controller, a typical assembly includes vehicle detection and other input/output devices.
coordinator	A device or program or routine that provides coordination.

1.4 ABBREVIATIONS AND ACRONYMS

The acronyms used in NTCIP 1211 v01 are defined as follows:

ASC	Actuated Signal Controller
ASN.1	Abstract Syntax Notation One
CO	Coordinator
GLO	Global (Object Definitions)
MIB	Management Information Base
NTCIP	National Transportation Communications for Intelligent Transportation Systems (ITS) Protocol
OER	Octet Encoding Rules
PICS	Protocol (or Profile) Implementation Conformance Specification
PRG	Priority Request Generator
PRS	Priority Request Server
RFC	Request for Comments
SCP	Signal Control and Prioritization
TCIP	Transit Communications Interface Profiles
TED	Time of Estimated Departure
TMC	Traffic or Transit Management Center
TSD	Time of Service Desired

NTCIP 1211 v01 references object definitions that appear in other standards. In those cases, the object name is preceded by the entity acronym followed by a period. For example, the coordForceMode object that is defined in NTCIP 1202 v02, *Object Definitions for Actuated Signal Controllers*, would be referred to

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

as SCP.coordForceMode. The controllerLocalTime object that is defined in NTCIP 1201 v02, *Global Object Definitions*, would be referred to as GLO.controllerLocalTime.

1.5 SCP OBJECT TREE

The Object Definitions for Signal Control and Prioritization are defined under the nema.transportation node of the ISO global naming tree to uniquely identify all management information (object definitions). It also references numerous objects in the SCP and Global subtrees of the global naming tree.

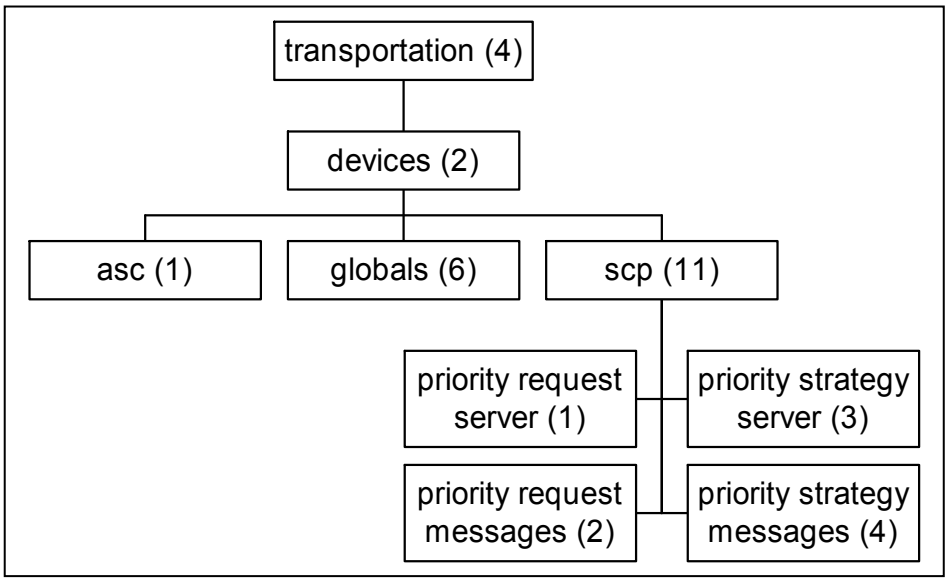


Figure 1-1
Naming Tree Organization

 © 2008 AASHTO / ITE / NEMA

Section 2
FUNCTIONAL REQUIREMENTS

2.1 REQUEST FOR PRIORITY SERVICE USE CASES

Every implementation of a Signal Control and Prioritization system shall have two primary components: a Priority Request Generator (PRG) and a Priority Request Server (PRS). The primary functions of the Priority Request Generator are as follows:

- To determine whether a vehicle is in need of preferential treatment (priority) at a signalized intersection according to series of user-defined criteria (i.e., lateness, occupancy level, etc.)
- To produce an estimate of the vehicles estimated time for service desired at the signalized intersection. This estimate, measured in seconds, is intended to represent the vehicles arrival time at the intersection and can range from zero (0) (representing a request for immediate service) to some time in the future.
- To communicate the vehicle's request for priority and its time of service desired to the Priority Request Server.
- To produce a log of all priority requests for processing by a fleet management agency.

Elements of the Priority Request Generator can be physically located in many different locations and multiple Fleet Vehicles from different Fleet operations can all produce requests for priority. Generally, it is the responsibility of the Fleet Management entities to install, operate, and maintain their equipment. Standards and specifications relating to the design of the priority request generator for Transit Vehicles may be found in the the APTA TCIP v2.

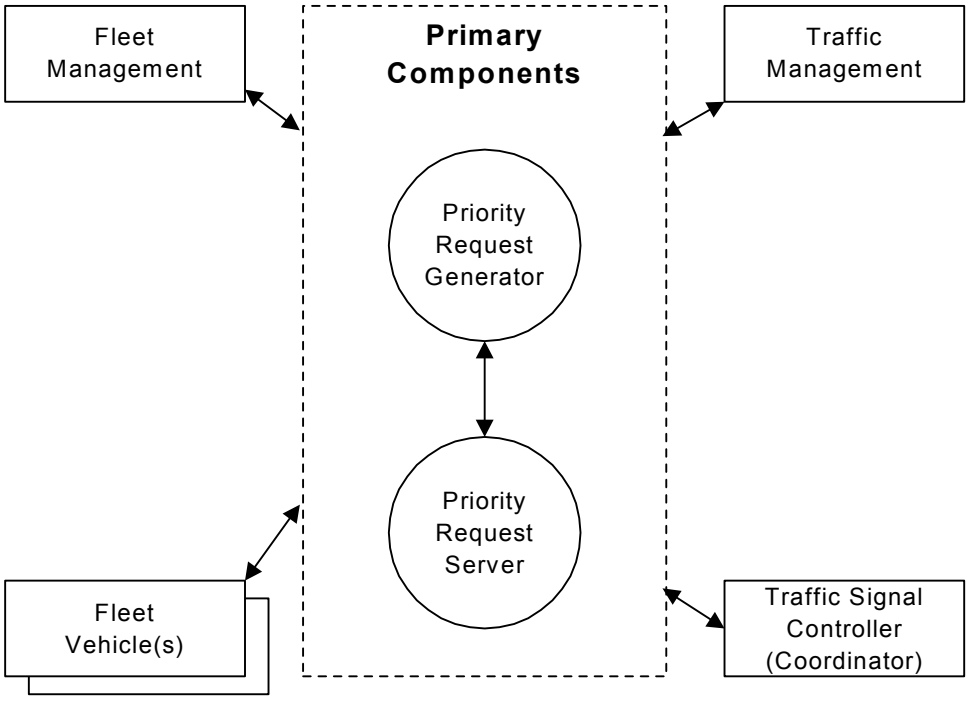


Figure 2-1
SCP System

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

The primary functions of the Priority Request Server are as follows:

- To receive multiple Priority Requests from different Priority Request Generators,
- To prioritize all the different Priority Requests for priority based on the requests vehicle class, vehicle level, and time of service desired.
- To generate a Service Request that defines the strategy to be used by the Traffic Signal Controller to provide priority to the Fleet Vehicle.
- To communicate the Service Request to the Traffic Signal Controller to be processed by the Coordinator.
- To produce a log of all the Priority Requests received and Service Requests generated by the Priority Request Server.

In most implementations, the Priority Request Server is located in the traffic signal cabinet and can be implemented as either a separate component installed in the traffic signal cabinet or incorporated internally in the traffic signal controller logic. The Traffic Management entity has the primary responsibility for installing, operating, and maintaining a Priority Request Server. The Priority Request Server should be capable of being configurable remotely through the Traffic Management Entity. NTCIP 1211 v01 primarily describes the standards and specification relating to the design of the Priority Request Server.

Many different physical implementation scenarios and use cases are associated with the activation of Signal Control and Prioritization. Figure 2-2 through Figure 2-9 illustrate several of the most likely. Note that these scenarios are based upon logical processes and may result in different physical implementations.

Do Not Copy Without Written Permission © 2008 AASHTO / ITE / NEMA

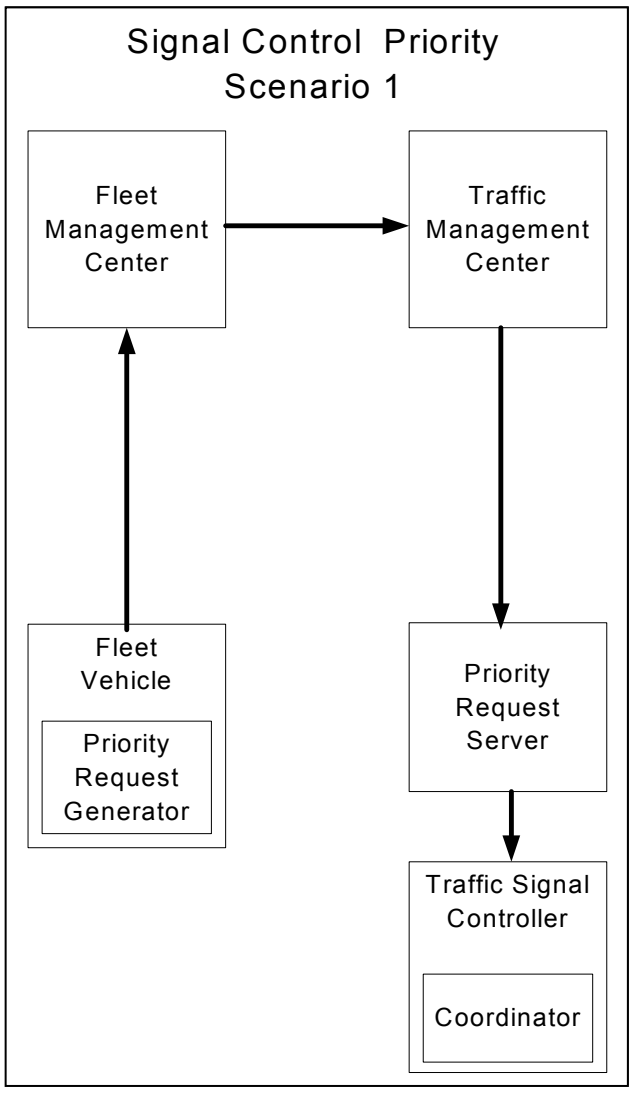


Figure 2-2
Scenario 1

The first scenario and use case (Figure 2-2 and Figure 2-3) represents a case in which the Priority Request Generator resides in the Fleet Vehicle but there is no direct communications path between the Fleet Vehicle and the Traffic Signal Controller. In this scenario, the Priority Request Generator generates a "priority request." Because there is no direct communications path between the Fleet Vehicle and the Traffic Signal Control, the priority request is routed through the Fleet Management Center to the Traffic Management Center, which in turn, forwards the priority request to the Priority Request Server. The Priority Request Server is a logical entity that receives all "priority requests" from multiple Priority Request Generators, and processes them to generate "service requests" that are passed to the Coordinator, the logical entity in the Traffic Signal Controller that executes the service request. In this scenario, the Priority Request Server can physically reside, either partially or in total, in either of Traffic Management Center or the Traffic Signal Cabinet.

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

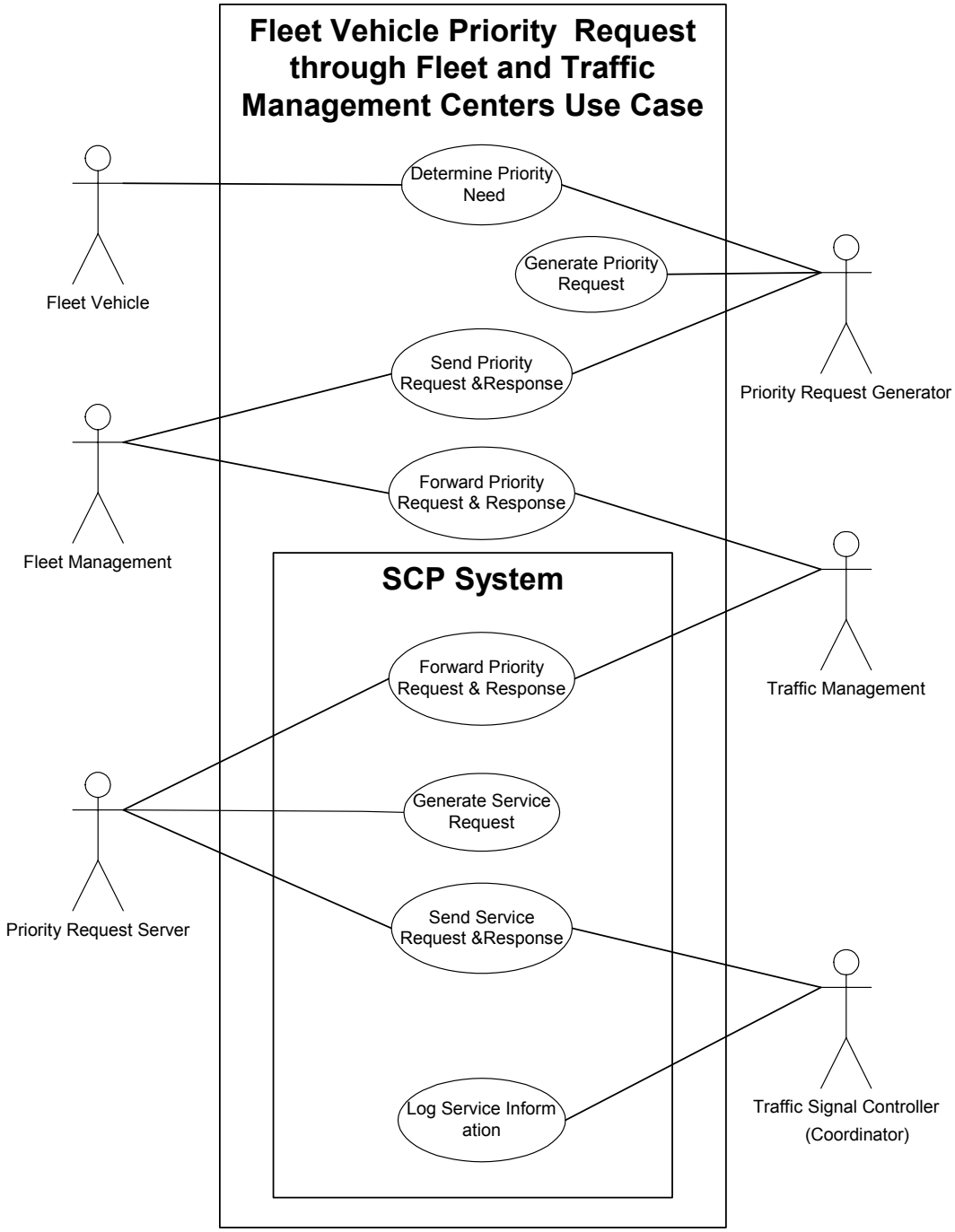


Figure 2-3
Scenario 1 – System Component Diagram

There are several slight variations of this scenario and use case. As shown in Figure 2-4 and Figure 2-6 respectively, the Priority Request Generator could be physically located in the Fleet Management or Traffic Management Center. In both of these cases, information needed to generate a priority request would be forwarded along until it reached the Priority Request Generator. As in the Figure 2-4 and Figure 2-5 (the second scenario and use case), the Priority Request Generator, located in the Fleet

 © 2008 AASHTO / ITE / NEMA

Management Center, would determine the need for priority and route the request for priority to the Priority Request Server through the Traffic Management Center. As in the first scenario, the Priority Request Server could be physically located, in part or in whole, in either the Traffic Management Center or in the Traffic Signal Cabinet.

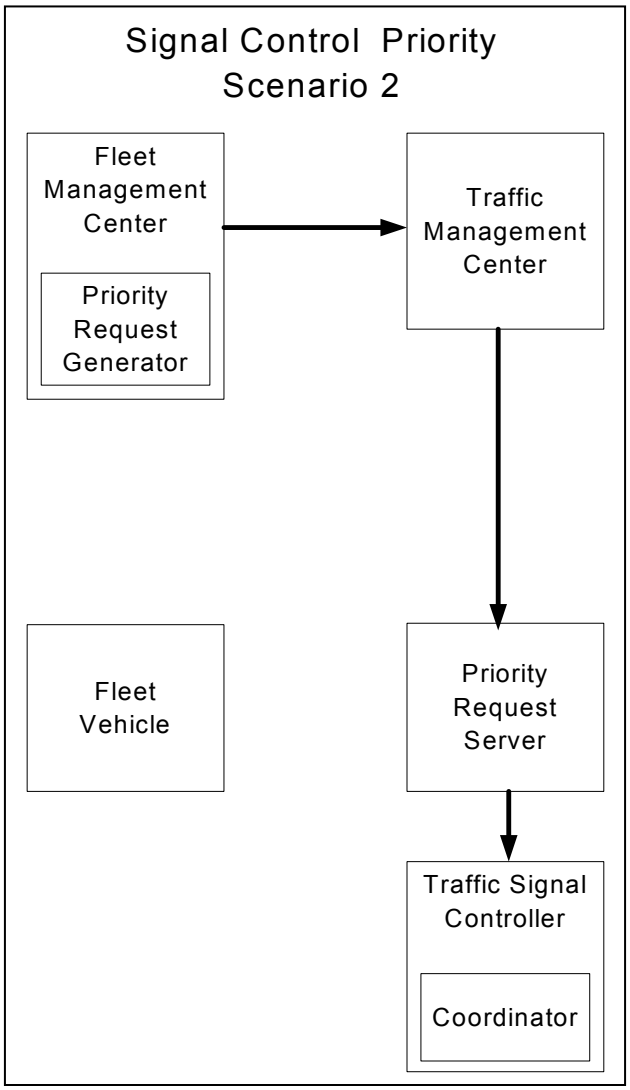


Figure 2-4
Scenario 2

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

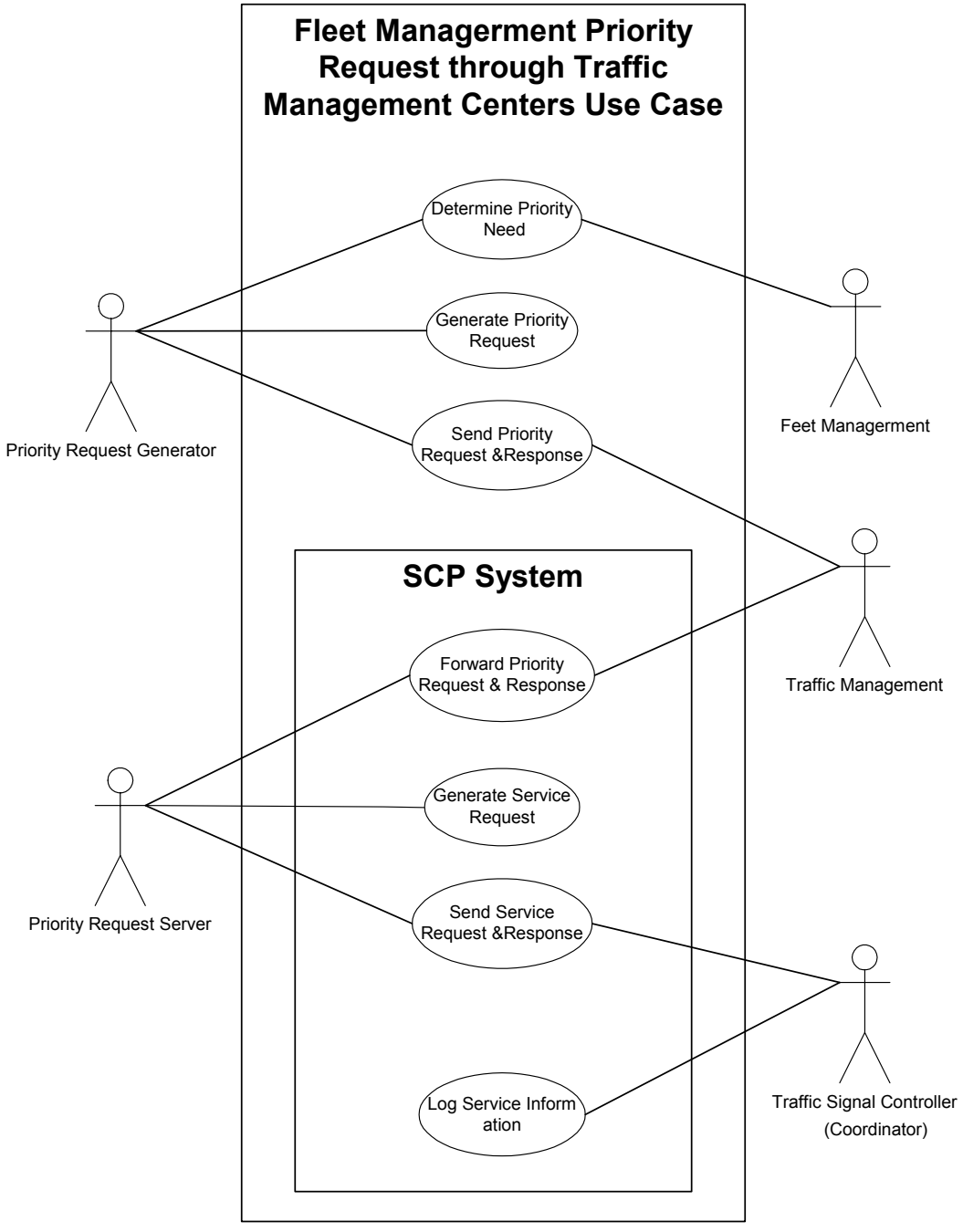


Figure 2-5
Scenario 2 – System Component Diagram

 © 2008 AASHTO / ITE / NEMA

The third scenario and use case is one in which the Priority Request Generator resides either as a logical or physical entity in the Traffic Management Center and there is not a direct communications path between the Fleet Management Centers. In this case, information needed to generate a request for priority is channeled directly to the Priority Request Generator in the Traffic Management Center. The Priority Request Generator determines the need for priority, initiates a priority request and sends the request to the Priority Request Server, which again, can reside in part or in whole in the Traffic Management Center or the Traffic Signal Cabinet.

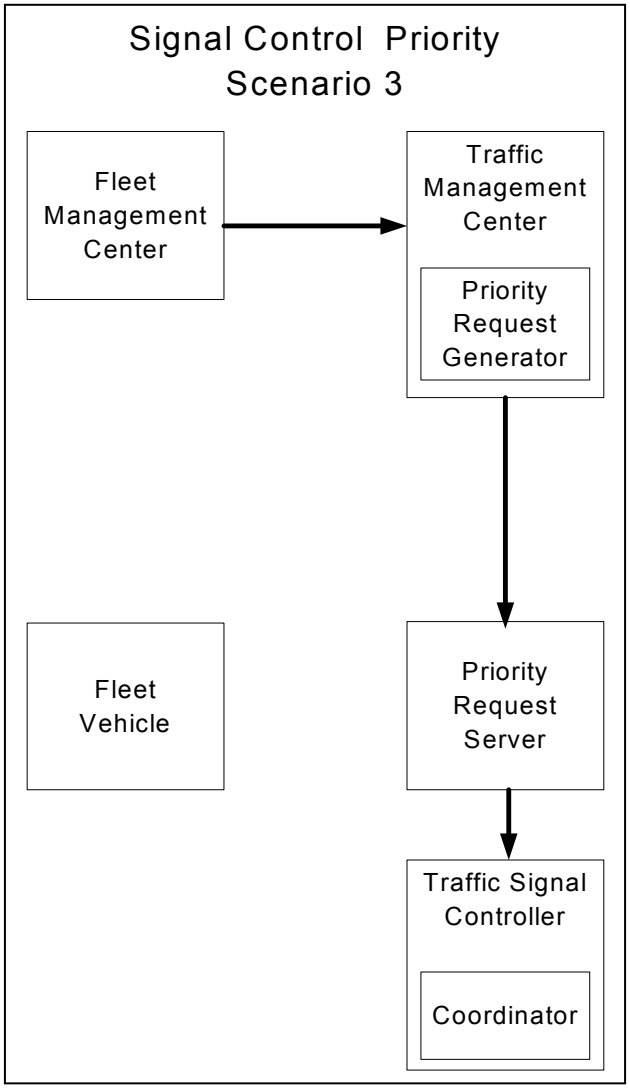


Figure 2-6
Scenario 3

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

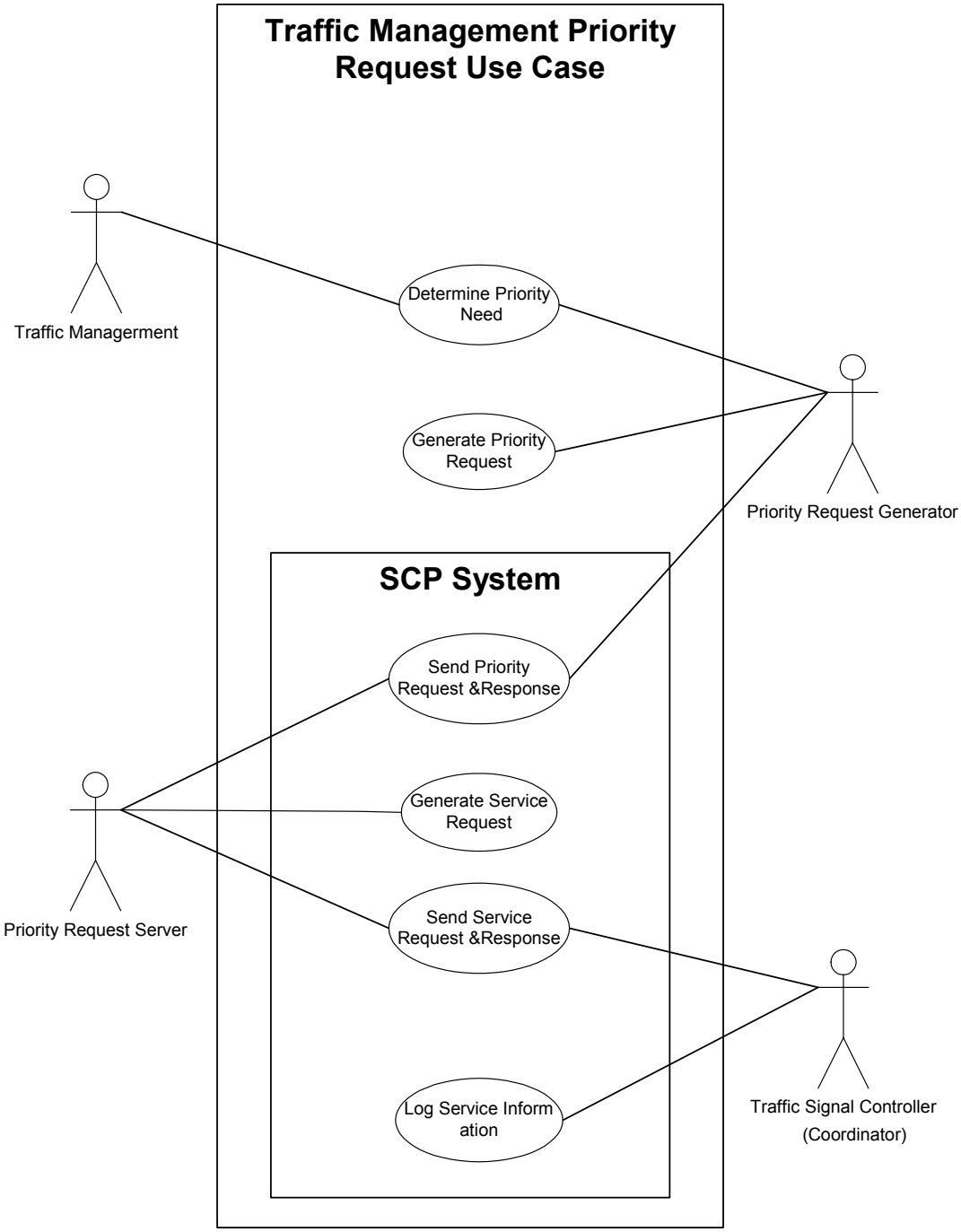


Figure 2-7
Scenario 3 – System Component Diagram

The fourth scenario and use case is one in which there is a direct communications link between the Fleet Vehicle and the Traffic Signal Cabinet. As in the first case, the Priority Request Generator resides within a Fleet Vehicle and use predefined criteria to generate a priority request. In this case, however, the Priority Request Generator sends the priority request directly to a Priority Request Server, which is a logical or physical entity located in the Traffic Signal Controller cabinet. The Priority Request Server that receives priority service requests, processes them, and resolves them to one or more priority service requests that are passed to the Traffic Signal Controller to be processed by the Coordinator.

Do Not Copy Without Written Permission © 2008 AASHTO / ITE / NEMA

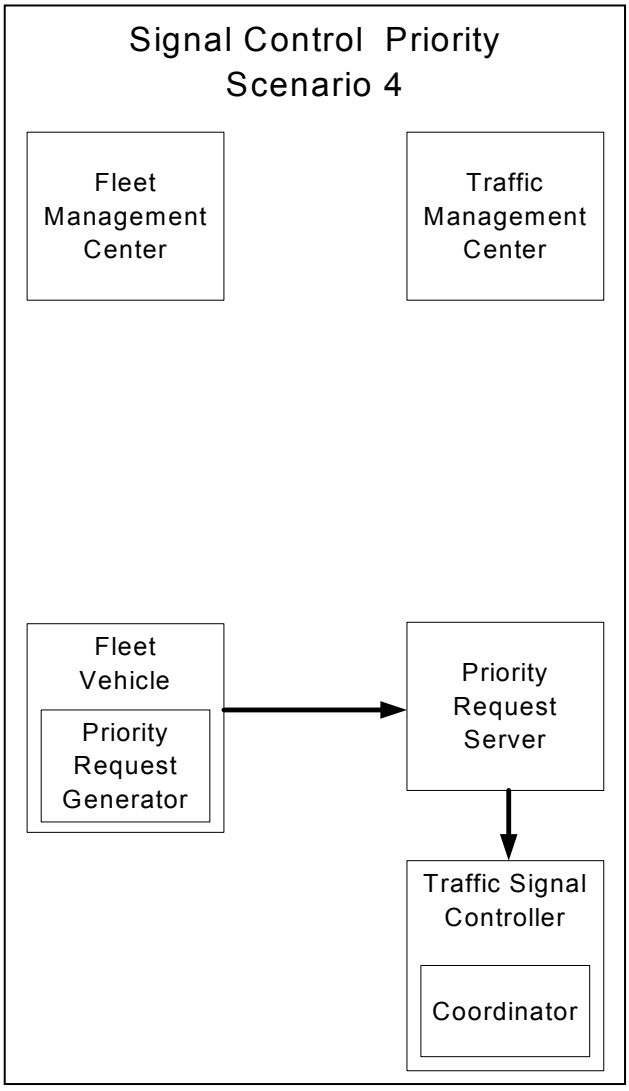


Figure 2-8
Scenario 4

© 2008 AASHTO / ITE / NEMA
Do Not Copy Without Written Permission

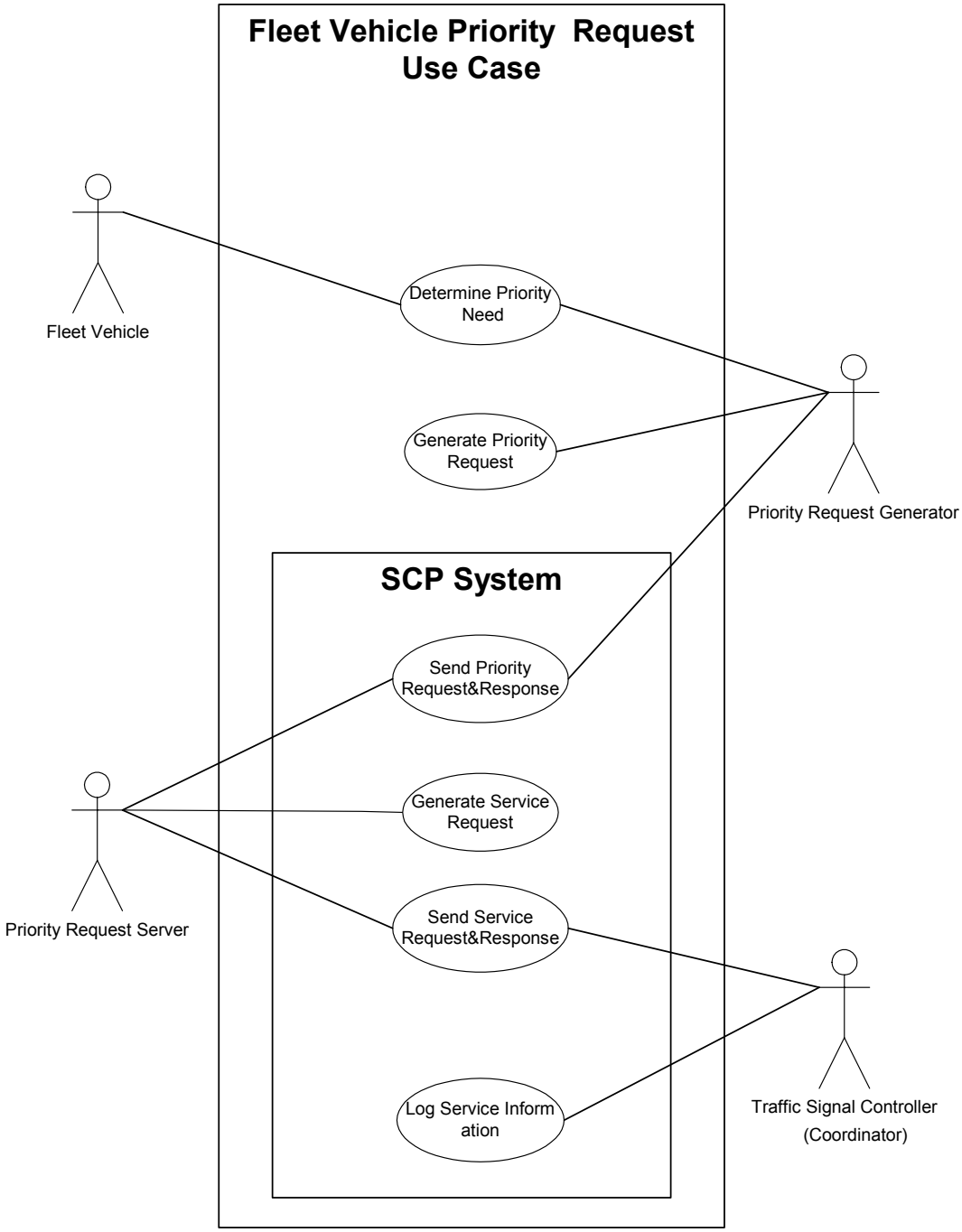


Figure 2-9
Scenario 4 – System Component Diagram

These four scenarios and use cases also assume that the Priority Request Server is implemented as a physical entity. It is possible that a traffic signal controller design integrates all the functions of a Priority Request Server as a logical entity within the controller.

The scope of NTCIP 1211 v01, in regard to request for priority service, is:

1. The priority request *messages* that would be exchanged between a Priority Request Generator and a Priority Request Server.

 © 2008 AASHTO / ITE / NEMA

2. The priority request *messages* that would be exchanged between Traffic Management and a Priority Request Server.
3. The service request *messages* that would be exchanged between a Priority Request Server and a Traffic Signal Controller to be processed by the Coordinator (assuming the Priority Request Server is implemented as a separate entity).

2.2 MANAGEMENT USE CASES

A Signal Control and Prioritization System also needs to be configured and monitored. Figure 2-10 illustrates the use cases.

Except where the Priority Request Generator is located in Traffic Management, Fleet Management shall be responsible for configuration and monitoring of the Priority Request Generator. The functionality associated with configuring and monitoring the Priority Request Generator is outside the scope of NTCIP 1211 v01.

Traffic Management shall be responsible for configuration and monitoring of any Priority Request Server and the Coordinator that resides in a Traffic Signal Controller.

The scope of NTCIP 1211 v01, in regard to configuration and monitoring, is:

1. The configuration and monitoring *messages* that would be exchanged between Traffic Management and a Priority Request Server.
2. The configuration and monitoring *messages* that would be exchanged between Traffic Management and a Coordinator that resides in a Traffic Signal Controller.

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

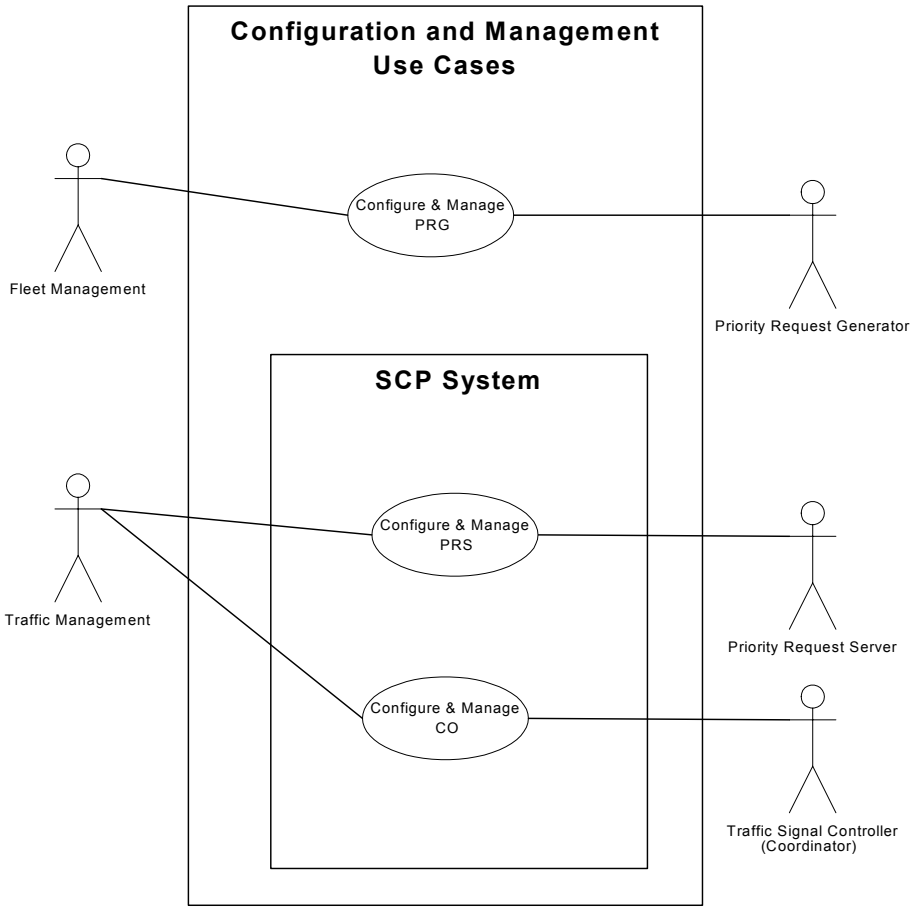


Figure 2-10
Configuration and Monitoring Use Cases

2.3 PRIORITY REQUEST AND RESPONSE DIALOGS

The basic priority request and response dialogs between a Priority Request Generator (PRG), a Priority Request Server (PRS), and a Coordinator (CO) are illustrated in Figure 2-11.

 © 2008 AASHTO / ITE / NEMA

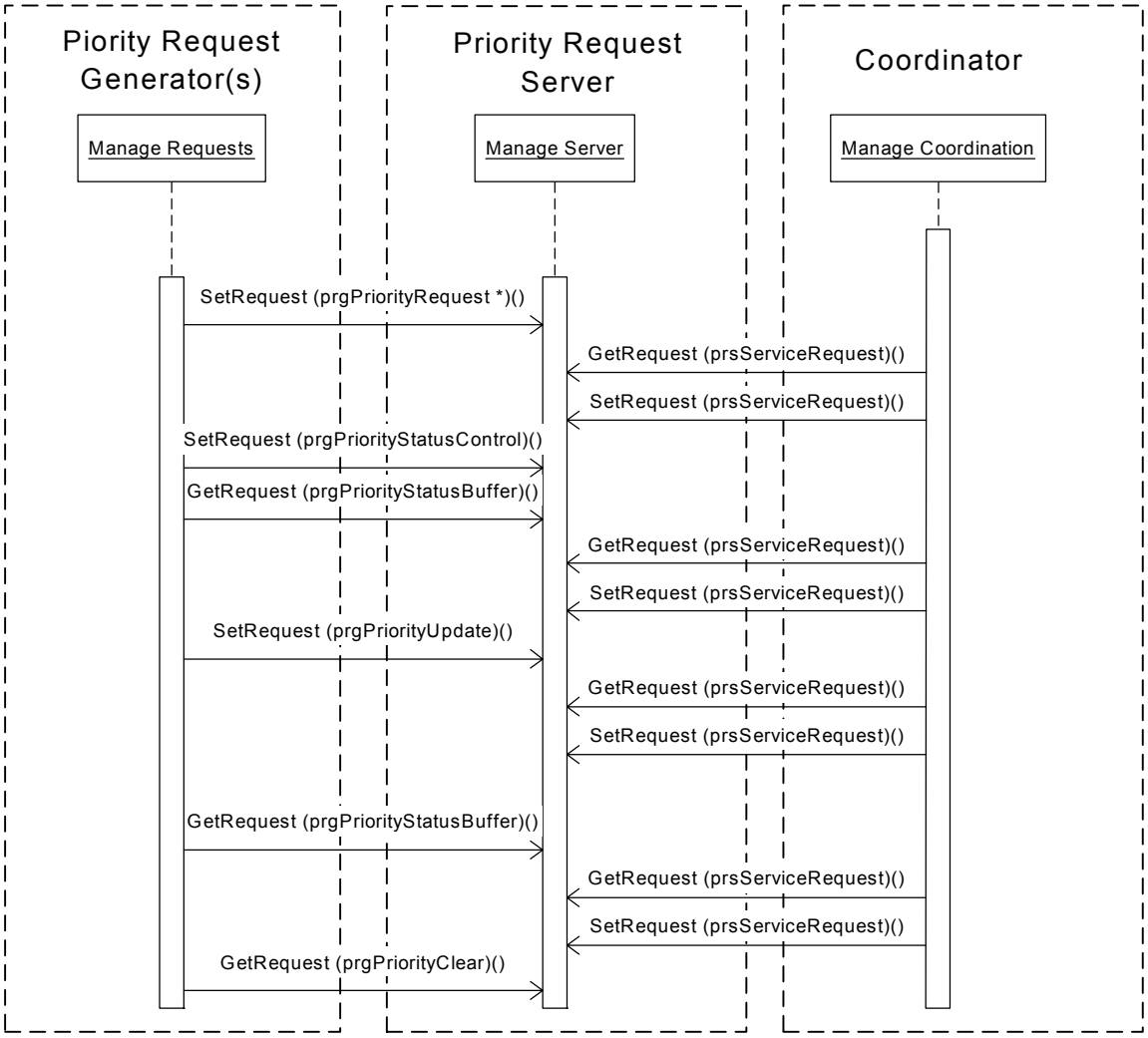


Figure 2-11
Basic Priority Request Sequence Diagram

In the right hand side of Figure 2-11, the CO is assumed to act as the manager and the PRS to act as an agent. If the reverse were true, the arrows would point in the opposite direction and the opcode pairs would be reversed. Annex C provides more detail on specific scenarios and the contents of the data frames that are exchanged.

2.3.1 Priority Request Generator – Priority Request Server Exchanges

The PRG shall send the PRS a "priority request " message, as an SNMP SetRequest consisting of a single variable binding as defined in section 3.2.1, to initiate a new priority request.

The PRG shall send the PRS a "priority update" message, as an SNMP SetRequest consisting of a single variable binding as defined in section 3.2.2, to update a priority request.

The PRG shall send the PRS a "priority status control" message, as an SNMP SetRequest consisting of a single variable binding as defined in section 3.2.3, to point to the information about a specific priority request.

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

The PRG shall send the PRS a "priority status buffer" message, as an SNMP GetRequest consisting of a single variable binding as defined in section 3.2.4, to retrieve information about a priority request.

The PRG shall send the PRS a "priority cancel" message, as an SNMP SetRequest consisting of a single variable binding as defined in section 3.2.5, to cancel a priority request.

> NOTE — Figure 2-11 does not show a "priority cancel". However, it may be sent after a "priority request" has been sent.

The PRG shall send the PRS a "priority clear" message, as an SNMP SetRequest consisting of a single variable binding as defined in section 3.2.6, to remove all information about priority request.

> NOTE — The term "variable binding" is used to represent a sequence of identifiers and/or values for the objects of interest (See NTCIP 1103 v01 for more information).

> NOTE — The above exchanges require the PRG to act as a manager and the PRS to act as an agent.

2.3.2 Priority Request Server – Coordinator Exchanges

If the PRS is implemented as an integral part of the traffic signal controller, the method by which the PRS and CO exchange information is implementation specific**.**

If the PRS is implemented as a separate entity and is located within the traffic control assembly, the assumption is that communications are such that the CO acts as a manager and the PRS acts as an agent (i.e. intra-cabinet communications). In this case, the CO shall send the PRS a "service request" message, as an SNMP GetRequest consisting of a single variable binding as defined in Section 4.2.8, to retrieve information about priority service requests. The CO shall also send the PRS the same "service request " message, but as an SNMP SetRequest to provide status about priority service requests.

If the PRS is implemented as a separate entity but is NOT located within the traffic control assembly, the assumption is that communications are such that the PRS acts as a manager and the CO acts as an agent (i.e. systems communications). In this case, the PRS shall send the CO a "service request" message, as an SNMP SetRequest consisting of a single variable binding as defined in Section 4.2.8, to retrieve information about priority service requests. The PRS shall also send the CO the same "service request " message, but as an SNMP GetRequest to provide status about priority service requests.

2.3.3 Configuration and Monitoring Dialogs

The basic configuration and monitoring dialogs between Traffic Management (TM) and a PRS and between TM and a CO that resides in Traffic Signal Controller are illustrated in the following in Figure 2-12.

 © 2008 AASHTO / ITE / NEMA

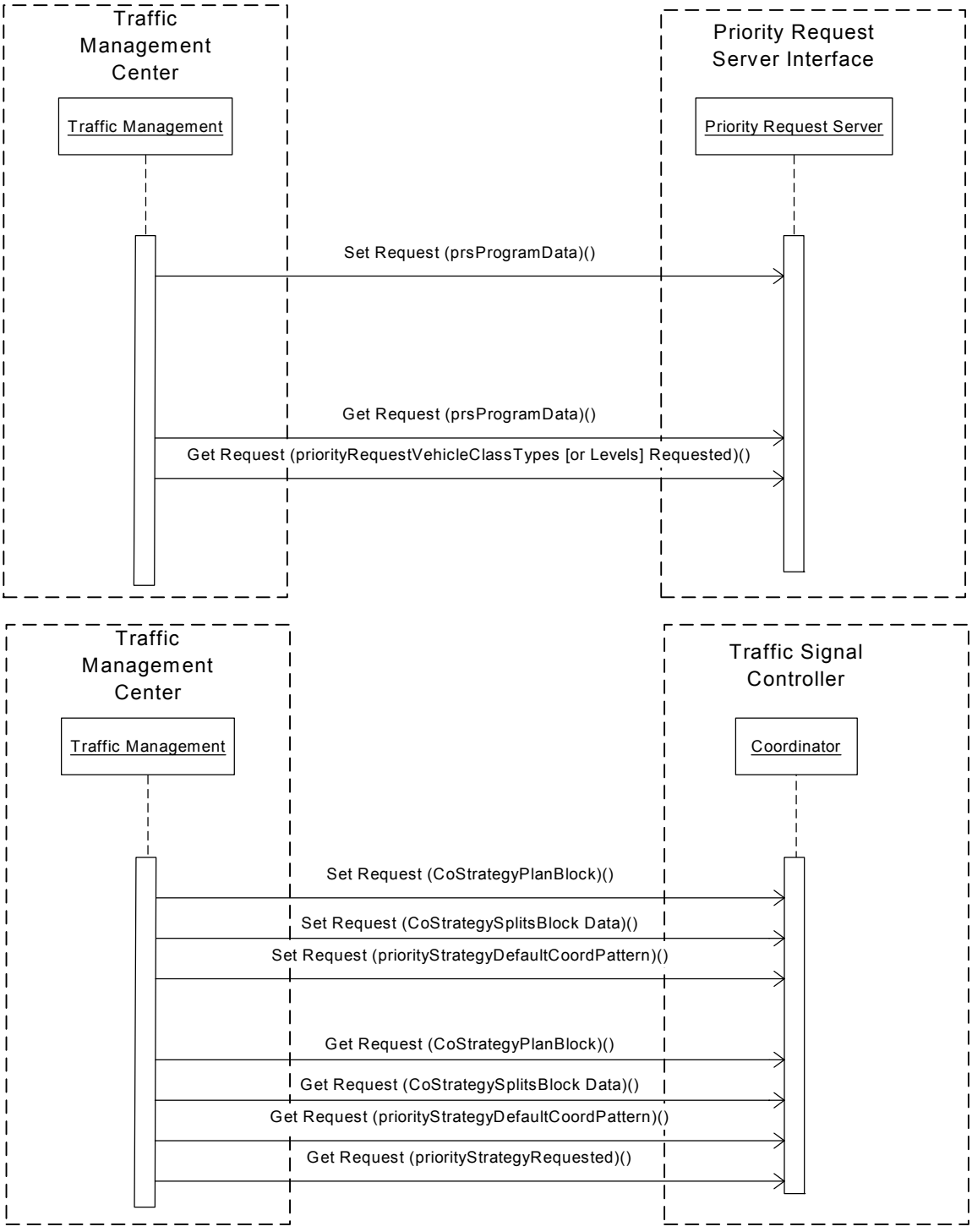


Figure 2-12
Basic Configuration and Monitoring
Sequence Diagrams

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

2.3.4 Traffic Management – Priority Request Server Exchanges

TM shall send the PRS database information about specific parameters related to prioritization configuration. These operations shall be carried out as a SetRequest with a single variable binding (prsProgramData) as defined in Clause 3.2.7.

2.3.5 Traffic Management – Coordinator Exchanges

TM shall send the CO configuration information about the database parameters related to strategy definition and modifiers for split timing during prioritization. These operations shall be carried out as sequence of SetRequests using the block control objects defined Clause 4.3 and the block objects defined in Clause 4.4.

TM may ask for status from the CO about service requests. In Figure 2-12, priorityStrategyRequested (Section 4.2.2.1.2) is shown as one of the objects that may be of interest. Other objects of interest are defined in Sections 4.2.2.1.3, 4.2.2.1.4, and 4.2.2.1.5. This information can be retrieved by the use of a GetRequest or sequence of GetRequests. The specific content of these messages is not defined.

2.4 PRIORITY REQUEST SERVER FUNCTIONS

2.4.1 Receipt of Priority Request

Upon receipt of a "priority request" message, the PRS shall perform several checks before responding with an acknowledgement. The message shall be evaluated by the PRS for validity and an Error Status of badValue (3) be returned for the following conditions:

a. if the SET length is NOT 25, or
b. if the contents of any message OID field cannot be parsed to fit the SYNTAX defined for the referenced object.

In order to enter priority request information in the priorityRequestTable, at least one row in the table shall have a priorityRequestStatusInPRS of "idleNotValid". If not, a "noSuchName" error-status is returned and no further processing takes place. (This error-status can be considered as a buffer full error.)

If the above checks pass, the following functions are performed by the PRS:

a. The information is stored in the appropriate priorityRequestTableEntry,
b. The priorityRequestTimeOfMessage is recorded,
c. The priorityRequestTimeToLive is set,
d. The priorityRequestStatusInPRS is set equal to "readyQueued", and
e. A "noError" error-status is returned.

2.4.2 Receipt of Priority Update

Upon receipt of a "priority update" message, the PRS shall perform several checks before responding with an acknowledgement. The message shall be evaluated by the PRS for validity and an Error Status of badValue (3) shall be returned for the following conditions:

a. If the SET length is NOT 25, or
b. if the contents of any message OID field cannot be parsed to fit the SYNTAX defined for the referenced object.

The message shall also be evaluated by the PRS for a matching entry in the priorityRequestTable and if not, an Error Status of noSuchName(2) shall be returned when a match cannot be found.

A matching entry requires a priorityRequestStatusInPRS of NOT "idleNotQueued" and all of the following to have the same values:

a. priorityRequestID,
b. priorityRequestVehicleID,
c. priorityRequestVehicleClassType,

Do Not Copy Without Written Permission © 2008 AASHTO / ITE / NEMA

d. priorityRequestVehicleClassLevel, and
e. priorityRequestServiceStrategyNumber

If the above checks pass, the following functions are performed by the PRS:
a. The values for priorityRequestTimeOfServiceDesired and priorityRequestTimeOfEstimatedDeparture are set equal to the values received, and
b. A "noError" error-status is returned.

2.4.3 Receipt of Priority Status Control

Upon receipt of a "priority status control" message, the PRS shall perform several checks before responding with an acknowledgement. The message shall be evaluated by the PRS for validity and an Error Status of badValue (3) shall be returned for the following conditions:
a. if the SET length is NOT 21, or
b. if the contents of any message OID field cannot be parsed to fit the SYNTAX defined for the referenced object.

The message shall also be evaluated by the PRS for a matching entry in the priorityRequestTable and an Error Status of noSuchName (2) shall be returned when a match cannot be found (see 2.4.2 for the definition of a matching entry).

If the above checks pass, the following functions are performed by the PRS:
a. The information as stored in the appropriate priorityRequestTableEntry is placed in the prgPriorityStatusBuffer when a "priority status buffer" message is sent, and
b. A "noError" error-status is returned.

2.4.4 Receipt of Priority Status Buffer

Upon receipt of a "priority status buffer" message, the PRS shall use values currently in prgPriorityStatusControl to define the data to be returned in the response. When prgPriorityStatusControl has invalid data (i.e., it has not been Set yet), an Error Status of badValue (3) shall be returned.

2.4.5 Receipt of Priority Cancel

Upon receipt of a "priority cancel" message, the PRS shall perform several checks before responding with an acknowledgement. The message shall be evaluated by the PRS for validity and an Error Status of badValue (3) shall be returned for the following conditions:
a. if the SET length is NOT 21, or
b. if the contents of any message OID field cannot be parsed to fit the SYNTAX defined for the referenced object.

The message shall also be evaluated by the PRS for a matching entry in the priorityRequestTable and an Error Status of noSuchName (2) shall be returned when a match cannot be found (see 2.4.2 for the definition of a matching entry).

If these checks pass, then a "noError" error-status is returned.

If priorityRequestStatusInPRS is equal to "readyQueued" or "readyOverridden", then priorityRequestStatusInPRS is set equal to "closedCanceled".

If priorityRequestStatusInPRS is equal to "activeProcessing" or “activeAdjustNotNeeded”, the PRS sets priorityRequestStatusInPRS equal to "activeCancel".

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

2.4.6 Receipt of Priority Clear

Upon receipt of a "priority clear" message, the PRS shall perform several checks before responding with an acknowledgement. The message shall be evaluated by the PRS for validity and an Error Status of badValue (3) shall be returned for the following conditions:

a. if the SET length is NOT 21, or
b. if the contents of any message OID field cannot be parsed to fit the SYNTAX defined for the referenced object.

The message shall also be evaluated by the PRS for a matching entry in the priorityRequestTable and an Error Status of noSuchName (2) shall be returned when a match cannot be found (see 2.4.2 for the definition of a matching entry).

If priorityRequestStatusInPRS is not equal to "closedCanceled", "reserviceError", "closedTimeToLiveError", "closedTimeError", "closedStategyError", or "closedCompleted", a "genError" error-status is returned and no further processing takes place.
If the above checks pass, the following functions are performed by the PRS:

a. All information is set to its default value state,
b. priorityRequestStatusInPRS is changed to "idleNotValid", and
c. A "noError" error-status is returned.

At this point, any information related to the priority request ceases to exist.

2.4.7 Automatic Priority Clear

When prsBusy is TRUE, the PRS shall scan all priorityRequestTable entries to determine if one should be cleared automatically. The priorityRequestTimeToLive for each request that has a priorityRequestStatusInPRS of "readyX", or "closedX" is checked to see if the time has been exceeded.

If the time has been exceeded, the following functions are performed by the PRS:

a. All other information is set to its default value state, and
b. priorityRequestStatusInPRS is changed to "idleNotValid"

At this point, any information related to the priority request ceases to exist.

2.4.8 Reservice Processing

The appropriate priorityRequestReserviceClass"X"Time shall be compared against priorityRequestReserviceTimer to check whether the reservice time has been exceeded. If priorityRequestReserviceTimer is not greater than the appropriate priorityRequestReserviceClass"X"Time, any request with a priorityRequestStatusInPRS of "readyQueued" shall have the priorityRequestStatusInPRG set to "reserviceError".

2.4.9 Class Type Override Processing

If a priority request with a higher Class Type becomes "readyQueued" when a lower Class Type is "activeProcessing" or “activeAdjustNotNeeded”, the PRS sets priorityRequestStatusInPRS of the lower Class Type equal to "activeOverride".

2.4.10 Prioritization Processing

If the value of priorityRequestTimeOfServiceDesiredInPRS is more than priorityRequestTimeToLive then priorityRequestStatusInPRS set equal to "closedTimeToLiveError".

The PRS shall scan all priorityRequestTable entries to determine if any of the entries in the priorityRequestTable has a priorityRequestStatusInPRS of "activeX". If none do then:

Do Not Copy Without Written Permission © 2008 AASHTO / ITE / NEMA

a. Rows with priorityRequestStatusInPRS equal to "readyQueued" shall be prioritized according to highest priorityRequestVehicleClassType, highest priorityRequestVehicleClassLevel, and then soonest priorityRequestTimeOfServiceDesiredInPRS. The row with the highest value of each becomes priorityRequestEntryNumber "1". The row with the next highest of each shall becomes priorityRequestEntryNumber "2", etc.
b. These are followed by any row having priorityRequestStatusInPRS equal to "readyOverriden".
c. These are followed by any row having priorityRequestStatusInPRS equal to "closedX".
d. These are then followed by any row having priorityRequestStatusInPRS equal to "idleNotValid".
e. The value of prsBusy is set to FALSE
f. No further processing takes place until the CO performs a SetRequest of prsServiceRequest (and, therefore, setting prsBusy to TRUE.

The priorityRequestEntryNumber of these rows follows in order and does not necessarily reflect a priority.

2.4.11 Priority Request Server, Coordinator Synchronization and Timing

2.4.11.1 Synchronization – CO is Manager

In the case where the CO acts as a manager and the PRS acts as an agent, a "busy" signal is used to synchronize the processing of status changes in PRS. Upon PRS initialization, the PRS sets its "busy" signal (prsBusy). When the CO sends a SetRequest (prsServiceRequest), this action also sets its "busy" signal. When the PRS completes it initialization and/or prioritization processes (basically any change in priorityRequestStatusInPRS), the PRS resets the "busy" signal and it is passed to the CO so that the CO knows that the PRS has completed its prioritization process of requests in its queue. The CO can act upon the information it received when the PRS "busy" signal is reset. When the CO sends a subsequent SetRequest (prsServiceRequest) the PRS "busy" signal is raised and the PRS can perform its prioritization processes again. Also, see Section 2.4.11.3.

2.4.11.2 Synchronization – PRS is Manager

In the case where the PRS acts as a manager and the CO acts as an agent, a "busy" signal is used to synchronize the processing of status changes in CO. Upon CO initialization, the CO sets its "busy" signal (coBusy). When the PRS sends a SetRequest (prsServiceRequest), this action also sets its "busy" signal. When the CO completes it initialization and/or service request processing (basically any change in priorityRequestStatusInCO), the CO resets the "busy" signal and it is passed to the PRS so that the PRS knows that the CO has completed its processing of service requests. The PRS can act upon the information it received when the CO "busy" signal is reset. When the PRS sends a subsequent SetRequest (prsServiceRequest) the CO "busy" signal is raised again and the CO can perform its processing of service requests again. Also see Section 2.4.11.4.

2.4.11.3 Timing – CO is Manager

The definition of prsServiceRequest (See Section 4.2.8) is unique in that it is a definition of a block object for both the PRS and CO. If the CO is the primary station and the PRS is an agent, the CO would issue a GetRequest to the PRS with the value field set to "null". In response, the PRS would issue a GetResponse to the CO with the value field set to the values of the objects contained within the PRS:

priorityRequestServiceStrategyNumber
priorityRequestTimeOfServiceDesiredInPRS
priorityRequestTimeOfEstimatedDepartureInPRS
priorityRequestStatusInPRS
prsBusy (= NOT Busy indicating the information is valid as explained in Section 2.4.11.1)

When the CO processes the GetResponse, it stores the values in the equivalent objects within the CO:

priorityRequestServiceStrategyNumber
priorityStrategyRequestedTimeOfServiceDesired
priorityStrategyRequestedTimeOfEstimatedDeparture

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

priorityStrategyRequestStatusInCO

The CO then evaluates the state of any current priority service and the information it just received. It may or may not change the status of priority service. It would also only raise it "busy" signal when it completes its evaluation. Regardless of whether or not it completes it evaluation, the CO "may" issue a SetRequest of prsServiceRequest to the PRS with the value field set to the values contained within it:

priorityRequestServiceStrategyNumber
priorityStrategyRequestedTimeOfServiceDesired
priorityStrategyRequestedTimeOfEstimatedDeparture
priorityStrategyRequestStatusInCO
coBusy (= Busy or NOT Busy indicating the validity of the information)

The word "may" is used to reflect the fact that the communications channel might impose timing constraints on how often a message is sent rather than when a massage is ready to be sent.

When the PRS processes the SetRequest it just received, it checks the "busy" signal and, if set, the PRS stores the values in the objects within it. Following the rules of SMNP, the PRS issues a GetResponse to the CO with the value field set to the values it just received. When the CO processes the GetResponse, the values are, basically, discarded. Since PRS now knows that the CO has processed the information it had previously sent it (coBusy = Busy), the PRS would raise it own "busy" signal and perform its prioritization processes again. The CO would then send a GetRequest to start the process over again.

2.4.11.4 Timing – PRS is Manager

If the PRS is the primary station and the CO is an agent, the PRS would issue a SetRequest to the CO with the value field set to the values of the objects contained within the PRS:

priorityRequestServiceStrategyNumber
priorityRequestTimeOfServiceDesiredInPRS
priorityRequestTimeOfEstimatedDepartureInPRS
priorityRequestStatusInPRS
prsBusy (= NOT Busy indicating the information is valid as explained in Section 2.4.11.2)

In response, the CO would issue a GetResponse to the PRS with the value field set to the values that it just received in the message. Assuming that the prsBusy signal is not set, the CO processes the SetRequest and stores the values in the objects within the CO:

priorityRequestServiceStrategyNumber
priorityStrategyRequestedTimeOfServiceDesired
priorityStrategyRequestedTimeOfEstimatedDeparture
priorityStrategyRequestStatusInCO

The PRS then issues a GetRequest of prsServiceRequest to the CO with the value field set to "null". In response, the CO would issue a GetResponse with the value field set to the values of the objects contained within the CO:

priorityRequestServiceStrategyNumber
priorityStrategyRequestedTimeOfServiceDesired
priorityStrategyRequestedTimeOfEstimatedDeparture
priorityStrategyRequestStatusInCO
coBusy (= Busy or NOT Busy indicating the validity of the information)

The PRS would continue to issue GetRequests until the CO's coBusy signal was not set indicating the information was valid.

Once it received valid information for the CO, it would raise its own "busy" signal and perform its prioritization processes again. Once it was ready (not busy), it sends a SetRequest of prsServiceRequest to the CO to repeat the process over again.

Do Not Copy Without Written Permission © 2008 AASHTO / ITE / NEMA

2.4.12 State Transition Diagram

Figure 2-13 is an UML State Transition Diagram for priorityRequestStatusInPRS and priorityStrategyRequestStatusInCO. See Sections 3.1.1.1.9 and 4.2.2.1.5 for more about what states an entity can set and what states an entity just processes.

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

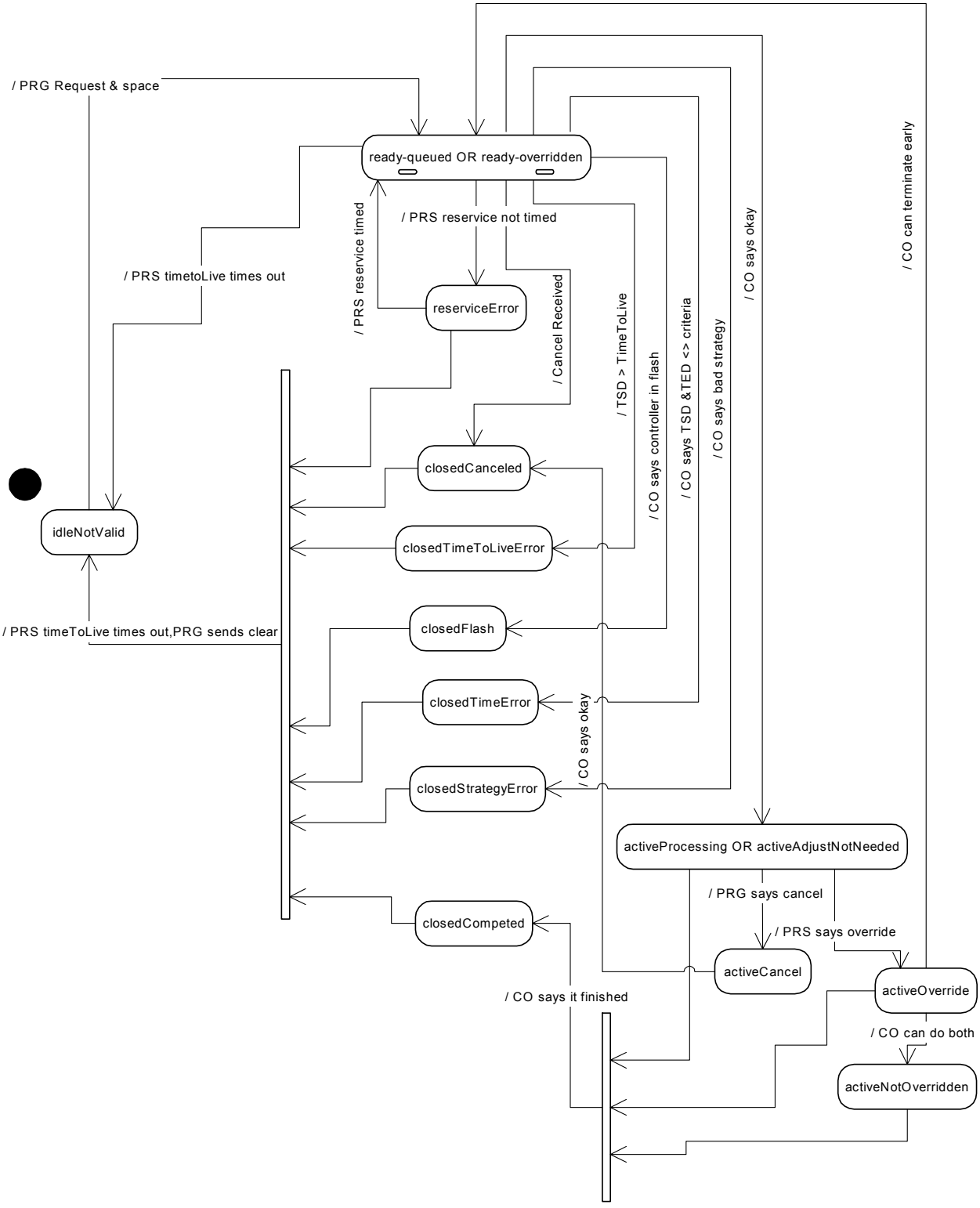


Figure 2-13
State Transition Diagram

© 2008 AASHTO / ITE / NEMA

2.4.13 Clock Synchronization

The functional design of the PRS requires a real-time clock in order to ensure that values of time expressed in message exchanges between the PRS and CO are synchronized. The specifications for the clock used by the CO are defined in the NEMA TS 2-2003 Standard.

2.5 COORDINATOR FUNCTIONS

The CO shall always act upon priorityRequestedEntryNumber "1". This does not preclude a CO from acting upon or considering other requests.

2.5.1 Get Service Request

The CO performs a "service request" GetRequest to transfer the following information from the PRS:

priorityRequestServiceStrategyNumber	>	priorityStrategyRequested
priorityRequestTimeOfServiceDesiredInPRS	>	priorityStrategyRequestedTimeOfServiceDesired
priorityRequestTimeOfEstimatedDepartureInPRS	>	priorityStrategyRequestedTimeOfEstimatedDeparture
priorityRequestStatusInPRS	>	priorityStrategyRequestStatusInCO

NOTE — All ten rows are always transferred and the transfer includes prsBusy.

If the priorityStrategyRequestStatusInCO is equal to "readyQueued", the CO checks
a. priorityStrategyRequestedTimeOfServiceDesired
b. priorityStrategyRequestedTimeOfEstimatedDeparture

If the values cannot meet some criteria defined by the CO, the CO set priorityStrategyRequestStatusInCO equal to "closedTimeError".

If not, the CO checks:
a. priorityStrategyRequested

If the coordination values associated with the priorityStrategyRequested cannot meet some criteria defined by the CO, the CO shall set priorityStrategyRequestStatusInCO equal to "closedStrategyError".

If the CO determines that a request can be met by current timing, the CO shall set priorityStrategyRequestStatusInCO equal to "activeAdjustNotNeeded”.

If the CO determines that the controller unit is operating in flash, the CO shall set priorityStrategyRequestStatusInCO equal to "closedFlash".

If the request passes these checks, the CO shall set priorityStrategyRequestStatusInCO equal to "activeProcessing".

At this point, the CO sends a "service request" to update the corresponding status in the PRS.

2.5.2 Processing Service Request

Once the "activeProcessing" or “activeAdjustNotneeded”state is reached, the CO shall be responsible for implementing the service request as a strategy defined by various parameters in the priorityStrategyTable (see Section 4.2.5) and the current running coordination pattern (see NTCIP 1202 v02).

The exact nature of how a CO responds is dependent upon its ability to respond to one or more than one service request at a time. The CO shall implement the strategy(s) as defined by the following:

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

The SCPs coordPatternStatus variable (or priorityStrategyDefaultCoordPattern if not coordinated) shall indicate that the following shall apply:

a. priorityStrategyMaximumReductionTime
b. priorityStrategyMaximumExtensionTime

The priorityStrategyRequested shall indicate that the following shall apply:

a. priorityStrategyPhaseOmits
b. priorityStrategyPedOmits

The CO shall then consider:

a. priorityStrategyRequestedTimeOfServiceDesired
b. priorityStrategyRequestedTimeOfEstimatedDeparture

Based upon all of these parameters, the CO shall adjust the "split" timing of the phases to give priority to the fleet vehicle while maintaining coordination. The controller unit has 'maintained coordination' following completion of a priority service request with the resumption of normal split timings at the appropriate points in the cycle. The intent is that the controller cycles back to or remain in the coordinated phase(s) such that they are green at the zero point in the cycle following completion of the priority service request and then time the allotted split time from that point. Skipping the coordinated phase(s) in any one cycle is not permitted.

A service request is complete when [TSD and TED have passed OR priority phase(s) have timed the max split extension OR priority has been cancelled] AND non-priority split reduction(s) equal priority split extension(s). Once the service request is complete, priority omits (phase and pedestrian) shall be removed.

Once the CO has completed the service request, the CO shall set priorityStrategyRequestStatusInCO equal to "closedCanceled" or "closedCompleted".

2.5.3 Cancel Processing

If a priority request's priorityStrategyRequestStatusInCO is equal to "activeCancel", the CO shall complete the active strategy as though the TSD and TED have passed. When the CO completes the strategy, priorityStrategyRequestStatusInCO is set equal to "closedCanceled".

2.5.4 Class Type Override Processing

Upon receipt of a service request with priorityStrategyRequestStatusInCO equal to "activeOverride", the CO may be able to complete the current request without affecting the need to cancel the CO. If it can, the CO would return priorityStrategyRequestStatusInCO set to "activeNotOverridden". If not, it completes the active strategy as thought the TSD and TED have passed and then sets priorityStrategyRequestStatusInCO set to "readyOverridden".

2.5.5 Set Service Status

The CO basically performs a "service status" SetRequest to transfer the following information back to the PRS:

priorityStrategyRequested	>	priorityRequestServiceStrategyNumber
priorityStrategyRequestedTimeOfServiceDesired	>	priorityRequestTimeOfServiceDesiredInPRS
priorityStrategyRequestedTimeOfEstimatedDeparture	>	priorityRequestTimeOfEstimatedDepartureInPRS
priorityStrategyRequestStatusInCO	>	priorityRequestStatusInPRS

NOTE — All ten rows are always transferred and the transfer includes coBusy.

 © 2008 AASHTO / ITE / NEMA

2.6 EVENT LOG PROCESSOR

The requirements for logging of priority service events are contingent upon support of a logging mechanism or function within the entity in question.

> NOTE — Logging can only take place on the change of the value of an object. For example, priorityRequestStatusInPRS or priorityStrategyRequestStatusInCO could be used to trigger an event log.

2.6.1 Priority Request Server Logging

If a PRS implementation supports the globalReport conformance group, logging shall be as defined in NTCIP 1201 v02 and usage is at the discretion of the user.

If the global report mechanism is used, a change of the value for priorityRequestStatusInPRS might be used to trigger an entry into the log.

2.6.2 Coordinator Logging

If a Traffic Signal Controller implementation supports the globalReport conformance group, logging shall be as defined in NTCIP 1201 v02 and usage is at the discretion of the user.

If the global report mechanism is used, a change of the value for priorityStrategyRequestStatusInCO might be used to trigger an entry into the log.

2.6.3 Fleet Management and Traffic Management Logging

The requirements for event logging in Fleet Management and TM are not defined in NTCIP 1211 v01. There is no current NTCIP standard that defines this functionality.

2.7 DATABASE MANAGEMENT PROCESSING

2.7.1 Management of Priority Request Generator

The functionality of Fleet Management in regard to configuration and monitoring of the PRG is outside the scope of NTCIP 1211 v01. Requirements may be covered in the proposed addition to the APTA TCIP.

2.7.2 Management of Priority Request Server

The functionality of the TM with respect to configuration and monitoring the PRS shall be as follows.

It shall be responsible for defining the values of:

a. priorityRequestTimeToLiveValue
b. priorityRequestReserviceClass"X"Times

2.7.3 Management of Coordinator

The functionality of the TM with respect to configuration and monitoring the SCP functions within a CO shall be as follows.

It shall be responsible for configuring the values of:

a. priorityStrategyServicePhases
b. priorityStrategyPhaseOmits
c. priorityStrategyPedOmits
d. priorityStrategyMaximumReductionTime
e. priorityStrategyMaximumExtensionTime
f. priorityStrategyDescription

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

There are no specific requirements related to monitoring other than TM shall use standard SNMP GetRequests to retrieve status information.

Do Not Copy Without Written Permission © 2008 AASHTO / ITE / NEMA

Section 3
OBJECT DEFINITIONS FOR PRIORITY REQUEST SERVER

This section defines the information requirements of the Priority Request Server portion of an SCP System.

The clauses below present the objects in lexicographical order of their OBJECT IDENTIFIERS, which correspond to their physical location within the global naming tree. Most of the objects defined in NTCIP 1211 v01 reside under the "scp" node of the global naming tree. To aid in object management, the SCP node has been subdivided into logical categories; each defined by a node under the SCP 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 may contain both "global" and "scp" specific objects.

3.1 PRIORITY REQUEST SERVER MIB

```
PRS-MIB1 DEFINITIONS ::= BEGIN

-- This group of objects represents the data elements that would reside in
-- a Priority Request Server.

IMPORTS

      EntryStatus, OwnerString, OerString,
      Byte, UByte, Short, Ushort, Long, Ulong
      FROM NEMA-SMI

      BITMAP16, scp
      FROM NEMA-SMI2

      OBJECT-TYPE
      FROM RFC-1212

      priorityRequestServer ::=  {scp 1}

      TrueFalse ::= INTEGER (0 | 255)
      FALSE ::= 0
      TRUE ::= NOT FALSE
```

3.1.1 Priority Request Table

```
priorityRequestTable   OBJECT-TYPE
   SYNTAX   SEQUENCE OF PriorityRequestEntry
   ACCESS   not-accessible
   STATUS   optional
   DESCRIPTION
           "A static table containing the parameters associated with each
           signal control priority request. The number of rows in this table is
           equal to 10."
::= { priorityRequestServer 2 }
```

© 2008 AASHTO / ITE / NEMA

3.1.1.1 Priority Request Table Entry

```
priorityRequestTableEntry   OBJECT-TYPE
      SYNTAX   PriorityRequestTableEntry
      ACCESS   not-accessible
      STATUS   optional
      DESCRIPTION
            "This object defines the parameters that are associated with
            priority requests."
      INDEX   { priorityRequestEntryNumber }
::= { priorityRequestTable 1 }

PriorityRequestTableEntry ::= SEQUENCE {
      priorityRequestEntryNumber               INTEGER,
      priorityRequestID                        INTEGER,
      priorityRequestVehicleID                 OCTET STRING (SIZE 17),
      priorityRequestVehicleClassType          INTEGER,
      priorityRequestVehicleClassLevel         INTEGER,
      priorityRequestServiceStrategyNumber     INTEGER,
      priorityRequestTimeOfServiceDesired      INTEGER,
      priorityRequestTimeOfEstimatedDeparture  INTEGER,
      priorityRequestStatusInPRS               INTEGER,
      priorityRequestTimeOfMessage             INTEGER,
      priorityRequestTimeToLive                INTEGER,
      priorityRequestTimeOfServiceDesiredInPRS     INTEGER,
      priorityRequestTimeOfEstimatedDepartureInPRS INTEGER
  }
```

3.1.1.1.1 Priority Request Entry Number

```
priorityRequestEntryNumber OBJECT-TYPE
      SYNTAX   INTEGER (1..10)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object represents the row number in the priority request
          table.  It is used by a management application for identification
          purposes."
::= { priorityRequestTableEntry 1 }
```

3.1.1.1.2 Priority Request ID

```
priorityRequestID OBJECT-TYPE
      SYNTAX   INTEGER (1..255)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object is the 'PRG requested' priority request identifier. It
          is assigned by the priority request generator so that further
          information related to a priority request can be identified. It
          shall be unique for this intersection from a vehicle ID of vehicle
          type."
      DEFVAL { 1 }
::= { priorityRequestTableEntry 2 }
```

 © 2008 AASHTO / ITE / NEMA

3.1.1.1.3 Priority Request Vehicle ID

```
 priorityRequestVehicleID     OBJECT-TYPE
      SYNTAX    OCTET STRING (SIZE 17)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object is the 'PRG requested' globally unique identifier of
          the entity requesting priority. For fleet vehicles, this is assumed
          to be the Vehicle Identification Number (VIN).  For management
          centers, the value is not defined but shall still be unique to
          differentiate the source of the priority request. "
      DEFVAL { "INVALID-VEH-ID-##" }      -- considered invalid
::= { priorityRequestTableEntry 3 }
```

3.1.1.1.4 Priority Request Vehicle Class Type

```
priorityRequestVehicleClassType     OBJECT-TYPE
      SYNTAX   INTEGER (1..10)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object is the 'PRG requested' class type (relative priority of
          a request).  The order of precedence is by class type with
            1     highest;
                     …
            10    lowest
          A request with a higher class type will override a lower class
          type."
      DEFVAL { 10 }
::= { priorityRequestTableEntry 4 }
```

3.1.1.1.5 Priority Request Vehicle Class Level

```
priorityRequestVehicleClassLevel    OBJECT-TYPE
      SYNTAX   INTEGER (1..10)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
            "This object is the 'PRG requested' class level (relative priority
            of a request within each class of request). The order of
            precedence is by class type and then class level.
            1     highest;
                     …
            10    lowest

            A request with a higher class level does NOT override a lower
            class level."
      DEFVAL { 10 }
::= { priorityRequestTableEntry 5 }
```

3.1.1.1.6 Priority Request Service Strategy Number

```
priorityRequestServiceStrategyNumber OBJECT-TYPE
      SYNTAX    INTEGER (0..255)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object is the 'PRG requested' strategy (1..255) or 0
          indicating the PRS has reset the the value to none.
```

© 2008 AASHTO / ITE / NEMA

```
            The PRS only checks for non-zero value in a request.  The CO
            determines whether a specific value represents a valid strategy.  A
            PRG shall never set this to a value of zero (0).  If it does, a PRS
            shall return a badValue error."
        DEFVAL { 0 }
::= { priorityRequestTableEntry 6 }
```

3.1.1.1.7 Priority Request Time of Service Desired

```
priorityRequestTimeOfServiceDesired OBJECT-TYPE
        SYNTAX   INTEGER (1..65535)
        ACCESS   read-only
        STATUS   optional
        DESCRIPTION
            "This object is the 'PRG requested' desired time in seconds to
            arrive at the intersection's stopping point relative to the receipt
            of the message.  A near side stopping point is assumed to be
            sufficiently close to the intersection's stop bar that regular
            queues frequently back up across the stopping point.  In this case,
            advance queue clearance prior to the arrival of fleet vehicle will
            be normally required.  For all practical purposes, arrival at the
            stopping point is the same as arrival at the stop bar.
            This is a relative time.

            Note that it is the responsibility of the PRG to take into account
            any communications delays between the PRG and the PRS."
        DEFVAL { 1 }
::= { priorityRequestTableEntry 7 }
```

3.1.1.1.8 Priority Request Time of Estimated Departure

```
priorityRequestTimeOfEstimatedDeparture   OBJECT-TYPE
        SYNTAX   INTEGER (1..65535)
        ACCESS   read-only
        STATUS   optional
        DESCRIPTION
            "This object is the 'PRG requested' estimated time in seconds of
            departure from the intersection's stopping point relative to the
            receipt of the message.
            This is a relative time.

            Note that it is the responsibility of the PRG to take into account
            any communications delays between the PRG and the PRS."
        DEFVAL { 1 }
::= { priorityRequestTableEntry 8 }
```

3.1.1.1.9 Priority Request Status

```
priorityRequestStatusInPRS    OBJECT-TYPE
        SYNTAX    INTEGER { idleNotValid (1),
                            readyQueued (2),
                            readyOverridden (3),
                            activeProcessing (4),
                            activeCancel (5),
                            activeOverride (6),
                            activeNotOverridden (7),
                            closedCanceled (8),
                            ReserviceError (9),
                            closedTimeToLiveError (10),
```

 © 2008 AASHTO / ITE / NEMA

```
                    closedTimerError (11),
                    closedStrategyError (12),
                    closedCompleted (13),
                    activeAdjustNotNeeded (14),
                    closedFlash (15) }

ACCESS   read-only
STATUS   optional
DESCRIPTION
    "This object provides status information about requests in the PRG.
    It is also passed to the CO as control information.
      idleNotValid        PRG determined that row does not contain valid
                          data
      readyQueued         PRG has validated the request but is waiting
                          for the CO to activate
      readyOverridden     CO has overridden the request
      activeProcessing    CO is processing the requested strategy
      activeCancel        PRS has asked that request be canceled
      activeOverride      PRS has asked that reuest be overriddden
      activeNotOverridden CO did not process the requested override
      closedCanceled      CO has canceled the request
      reserviceError      PRS determined that the request came too
                              soon after a previous request
      closedTimeToLiveError   PRG determined that TSD exceeds the time
                              to live
      closedTimerError    CO indicated that the requested times could
                          not be met
      closedStrategyError CO indicated that the requested strategy was
                          not valid.
      closedCompleted     CO has completed the requested
                          strategyprevious request
      activeAdjustNotNeeded   CO indicated that the request can be met
                              by the current timing and no adjustment is
                              necessary
      closedFlash         CO indicated that the controller is in flash

      Upon receipt of a activeProcessing or a activeAdjustNotNeeded from
      the CO, the PRS may change the status to:
             activeCancel   - upon reciept of a prgPriorityCancel
             activeOverride - upon reciept of a prgPriorityRequest with
                              higher class

      Upon receipt of a closedTimerError,  closedStrategyError,
      closedCanceled, or closedCompleted, or closedFlash from the CO,
      the PRS may change the status to:
             idleNotValid   - upon reciept of a prgPriorityClear or a
                              timeout of TimetToLive

      Upon receipt of a activeNotOverridden from the CO, the PRS may
      change the status to:
             activeCancel   - upon reciept of a prgPriorityCancel

      Upon receipt of a readyOverridden from the CO, the PRS may change
      the status to:
             readyQueued    - upon completion of overriding request
             closedCanceled - upon reciept of a prgPriorityCancel
             closedTimeToLiveError - a timeout of TimeToLive
```

© 2008 AASHTO / ITE / NEMA

```
            When the status is idleNotValid, the PRS may change the status to:
                   readyQueued    - upon reciept of a valid rgPriorityRequest

            When the status is readyQueued, the PRS may change the status to:
                   closedCanceled - upon reciept of a prgPriorityCancel
                   closedTimeToLiveError - a timeout of TimeToLive
                   reserviceError  - the request came too soon after a
                                     previous request

            Note: A change of status is predicated on the 'busy' signal being
            set."

      DEFVAL { idleNotValid }
::= { priorityRequestTableEntry 9 }
```

3.1.1.1.10 Priority Request Time of Message

```
priorityRequestTimeOfMessage OBJECT-TYPE
      SYNTAX    INTEGER --(0..4294967295)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
         "This object represents the initial time of receipt of the priority
         request.  The value of GLO.globalTime is copied to this object when
         new priority request is first entered in the priorityRequestTable."
      DEFVAL { 0 }
::= { priorityRequestTableEntry 10 }
```

3.1.1.1.11 Priority Request Time to Live

```
priorityRequestTimeToLive OBJECT-TYPE
      SYNTAX   INTEGER -- (0..4294967295)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
         "This object represents the initial time of receipt of the priority
         request priorityRequestTimeOfMessage plus
         priorityRequestTimeToLiveValue. When GLO.globalTime is equivalent
         to this value, a priority request that has a
         priorityRequestStatusInPRS of 'readyX' or 'closedX' shall have the
         priorityRequestStatusInPRS set to 'idleNotValid'. "
::= { priorityRequestTableEntry 11 }
```

3.1.1.1.12 Priority Request Time of Service Desired (CO)

```
priorityRequestTimeOfServiceDesiredInPRS OBJECT-TYPE
      SYNTAX    INTEGER -- (0..4294967295)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
         "This object is the 'PRS requested' desired time (GLO.globalTime) to
         arrive at the intersection's stopping point.  It is derived by
         adding priorityRequestTimeOfServiceDesired to
         priorityRequestTimeOfMessage."
::= { priorityRequestTableEntry 12 }
```

 © 2008 AASHTO / ITE / NEMA

3.1.1.1.13 Priority Request Time of Estimated Departure (PRS)

```
priorityRequestTimeOfEstimatedDepartureInPRS    OBJECT-TYPE
      SYNTAX   INTEGER -- (0..4294967295)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object is the 'PRS requested' time (GLO.globalTime) of
          departure from the intersection's stopping point. It is derived by
          adding priorityRequestTimeOfEstimatedDeparture to
          priorityRequestTimeOfMessage."
::= { priorityRequestTableEntry 13 }
```

3.1.2 Priority Request Server Busy

```
prsBusy OBJECT-TYPE
      SYNTAX   TrueFalse
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object is used to indicate to a CO that the PRS is
          initializing or otherwise processing and, when TRUE, the information
          included with it in a prsServiceRequest response is not valid.  It
          is used to synchonize exchanges between a PRS and CO.

          When the CO is acting as manager and the PRS is acting as an agent,
          prsBusy is set TRUE when the CO performs a SetRequest
          prsServiceRequest.

          When the PRS is acting as a manager and the CO is acting as the
          agent, the value shall always be set to FALSE.

::= { priorityRequestServer 2 }
```

3.1.3 Priority Request Time to Live Value

```
priorityRequestTimeToLiveValue      OBJECT-TYPE
      SYNTAX   INTEGER (0..65535)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object represents the value (expressed in seconds) that is
          added to the current value of GLO.globalTime and loaded into
          priorityRequestTimeToLive when a priority request is initially
          entered in the table."
      DEFVAL { 0 }
::= { priorityRequestServer 3 }
```

3.1.4 Priority Request Reservice Timer

```
priorityRequestReserviceTimer OBJECT-TYPE
      SYNTAX    Gauge (0..65535)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object represents a timer that is reset to zero at the
          completion of any strategy (priorityRequestStatusInPRS is initially
          set equal to 'closedCompleted'). It is incremented every second up
          to its maximum value of 65535 where it latches.
```

© 2008 AASHTO / ITE / NEMA

```
          This timer is compared to Priority Request Reservice Class X Times
          to determine whether an priority request that comes right after a
          previous will be honored."
::= { priorityRequestServer 4 }
```

3.1.5 Priority Request Reservice Class 1 Time

```
priorityRequestReserviceClass1Time  OBJECT-TYPE
      SYNTAX    INTEGER (0..65535)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object represents the time in seconds between the end of ANY
          strategy and the start of any new Class Type 1 priority request.
          The value of the priorityRequestReserviceTimer shall be equal to or
          greater than this value for priority request to be honored."
      DEFVAL { 0 }
::= { priorityRequestServer 5 }
```

3.1.6 Priority Request Reservice Class 2 Time

```
priorityRequestReserviceClass2Time  OBJECT-TYPE
      SYNTAX    INTEGER (0..65535)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object represents the time in seconds between the end of ANY
          strategy and the start of any new Class Type 2 priority request.
          The value of the priorityRequestReserviceTimer shall be equal to or
          greater than this value for priority request to be honored."
      DEFVAL { 0 }
::= { priorityRequestServer 6 }
```

3.1.7 Priority Request Reservice Class 3 Time

```
priorityRequestReserviceClass3Time  OBJECT-TYPE
      SYNTAX    INTEGER (0..65535)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object represents the time in seconds between the end of ANY
          strategy and the start of any new Class Type 3 priority request.
          The value of the priorityRequestReserviceTimer shall be equal to or
          greater than this value for priority request to be honored. "
      DEFVAL { 0 }
::= { priorityRequestServer 7 }
```

3.1.8 Priority Request Reservice Class 4 Time

```
priorityRequestReserviceClass4Time  OBJECT-TYPE
      SYNTAX    INTEGER (0..65535)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object represents the time in seconds between the end of ANY
          strategy and the start of any new Class Type 4 priority request.
          The value of the priorityRequestReserviceTimer shall be equal to or
          greater than this value for priority request to be honored. "
```

 © 2008 AASHTO / ITE / NEMA

```
      DEFVAL { 0 }
::= { priorityRequestServer 8 }
```

3.1.9 Priority Request Reservice Class 5 Time

```
priorityRequestReserviceClass5Time  OBJECT-TYPE
      SYNTAX    INTEGER (0..65535)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object represents the time in seconds between the end of ANY
          strategy and the start of any new Class Type 5 priority request.
          The value of the priorityRequestReserviceTimer shall be equal to or
          greater than this value for priority request to be honored."
      DEFVAL { 0 }
::= { priorityRequestServer 9 }
```

3.1.10 Priority Request Reservice Class 6 Time

```
priorityRequestReserviceClass6Time  OBJECT-TYPE
      SYNTAX    INTEGER (0..65535)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object represents the time in seconds between the end of ANY
          strategy and the start of any new Class Type 6 priority request.
          The value of the priorityRequestReserviceTimer shall be equal to or
          greater than this value for priority request to be honored."
      DEFVAL { 0 }
::= { priorityRequestServer 10 }
```

3.1.11 Priority Request Reservice Class 7 Time

```
priorityRequestReserviceClass7Time  OBJECT-TYPE
      SYNTAX    INTEGER (0..65535)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object represents the time in seconds between the end of ANY
          strategy and the start of any new Class Type 7 priority request.
          The value of the priorityRequestReserviceTimer shall be equal to or
          greater than this value for priority request to be honored."
      DEFVAL { 0 }
::= { priorityRequestServer 11 }
```

3.1.12 Priority Request Reservice Class 8 Time

```
priorityRequestReserviceClass8Time  OBJECT-TYPE
      SYNTAX    INTEGER (0..65535)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object represents the time in seconds between the end of ANY
          strategy and the start of any new Class Type 8 priority request.
          The value of the priorityRequestReserviceTimer shall be equal to or
          greater than this value for priority request to be honored."
      DEFVAL { 0 }
::= { priorityRequestServer 12 }
```

© 2008 AASHTO / ITE / NEMA Copy Per MIB Distribution Notice

3.1.13 Priority Request Reservice Class 9 Time

```
PriorityRequestReserviceClass9Time  OBJECT-TYPE
      SYNTAX    INTEGER (0..65535)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object represents the time in seconds between the end of ANY
          strategy and the start of any new Class Type 9 priority request.
          The value of the priorityRequestReserviceTimer shall be equal to or
          greater than this value for priority request to be honored."
      DEFVAL { 0 }
::= { priorityRequestServer 13 }
```

3.1.14 Priority Request Reservice Class 10 Time

```
priorityRequestReserviceClass10Time OBJECT-TYPE
      SYNTAX    INTEGER (0..65535)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object represents the time in seconds between the end of ANY
          strategy and the start of any new Class Type 10 priority request.
          The value of the priorityRequestReserviceTimer shall be equal to or
          greater than this value for priority request to be honored."
      DEFVAL { 0 }
::= { priorityRequestServer 14 }
```

3.2 PRIORITY REQUEST SERVER BLOCK OBJECTS

```
priorityRequestMessages ::=  { scp 2 }
```

3.2.1 Priority Request Message

```
prgPriorityRequest   OBJECT-TYPE
   SYNTAX   OCTET STRING {SIZE(25)}
   ACCESS   write-only
   STATUS   optional
   DESCRIPTION
      "<Definition> An OER encoded string of reference parameters to initiate
       a new priority request.

       The parameter values in this string when sent from a PRG to a PRS are:
         priorityRequestID                       INTEGER (1..255)
         priorityRequestVehicleID                OCTET STRING (SIZE 17)
         priorityRequestVehicleClassType         INTEGER (1..10)
         priorityRequestVehicleClassLevel        INTEGER (1..10)
         priorityRequestServiceStrategyNumber    INTEGER (1..255)
         priorityRequestTimeOfServiceDesired     INTEGER (1..65535)
         priorityRequestTimeOfEstimatedDeparture INTEGER (1..65535)"

::= { priorityRequestMessages 1 }
```

3.2.2 Priority Update Message

```
prgPriorityUpdate   OBJECT-TYPE
   SYNTAX   OCTET STRING {SIZE(25)}
   ACCESS   write-only
   STATUS   optional
```

 © 2008 AASHTO / ITE / NEMA

```
   DESCRIPTION
      "<Definition> An OER encoded string of reference parameters to update
       a priority request.

       The parameter values in this string when sent from a PRG to a PRS are:
         priorityRequestID                       INTEGER (1..255)
         priorityRequestVehicleID                OCTET STRING (SIZE 17)
         priorityRequestVehicleClassType         INTEGER (1..10)
         priorityRequestVehicleClassLevel        INTEGER (1..10)
         priorityRequestServiceStrategyNumber    INTEGER (1..255)
         priorityRequestTimeOfServiceDesired     INTEGER (1..65535)
         priorityRequestTimeOfEstimatedDeparture INTEGER (1..65535)"
::= { priorityRequestMessages 2 }
```

3.2.3 Priority Status Control Message

```
prgPriorityStatusControl   OBJECT-TYPE
   SYNTAX   OCTET STRING {SIZE(21)}
   ACCESS   write-only
   STATUS   optional
   DESCRIPTION
      "<Definition> An OER encoded string of reference parameters used to
       retrieve the status of a priority request.

       The parameter values in this string when sent from a PRG to a PRS are:
         priorityRequestID                       INTEGER (1..255)
         priorityRequestVehicleID                OCTET STRING (SIZE 17)
         priorityRequestVehicleClassType         INTEGER (1..10)
         priorityRequestVehicleClassLevel        INTEGER (1..10)
         priorityRequestServiceStrategyNumber    INTEGER (1..255)"
::= { priorityRequestMessages 3 }
```

3.2.4 Priority Status Buffer Message

```
prgPriorityStatusBuffer   OBJECT-TYPE
   SYNTAX   OCTET STRING {SIZE(23)}
   ACCESS   read-only
   STATUS   optional
   DESCRIPTION
       "<Definition> An OER encoded string defining the status of a
       priority request.

       The parameter values in this string when retrieved from a PRG are:
         priorityRequestID                       INTEGER (1..255)
         priorityRequestVehicleID                OCTET STRING (SIZE 17)
         priorityRequestVehicleClassType         INTEGER (1..10)
         priorityRequestVehicleClassLevel        INTEGER (1..10)
         priorityRequestServiceStrategyNumber    INTEGER (1..255)
         priorityRequestStatusInPRS              INTEGER (1..255)"
::= { priorityRequestMessages 4 }
```

3.2.5 Priority Cancel Message

```
prgPriorityCancel   OBJECT-TYPE
   SYNTAX   OCTET STRING {SIZE(21)}
   ACCESS   write-only
   STATUS   optional
   DESCRIPTION
      "<Definition> An OER encoded string of reference parameters to cancel
```

© 2008 AASHTO / ITE / NEMA

```
       a priority request.

       The parameter values in this string when sent from a PRG to a PRS are:
         priorityRequestID                       INTEGER (1..255)
         priorityRequestVehicleID                OCTET STRING (SIZE 17)
         priorityRequestVehicleClassType         INTEGER (1..10)
         priorityRequestVehicleClassLevel        INTEGER (1..10)
         priorityRequestServiceStrategyNumber    INTEGER (1..255)"
::= { priorityRequestMessages 5 }
```

3.2.6 Priority Clear Message

```
prgPriorityClear   OBJECT-TYPE
   SYNTAX   OCTET STRING {SIZE(21)}
   ACCESS   write-only
   STATUS   optional
   DESCRIPTION
      "<Definition> An OER encoded string of reference parameters to clear
       a priority request.

       The parameter values in this string when sent from a PRG to a PRS are:
         priorityRequestID                       INTEGER (1..255)
         priorityRequestVehicleID                OCTET STRING (SIZE 17)
         priorityRequestVehicleClassType         INTEGER (1..10)
         priorityRequestVehicleClassLevel        INTEGER (1..10)
         priorityRequestServiceStrategyNumber    INTEGER (1..255)"
::= { priorityRequestMessages 6 }
```

3.2.7 Program Data

```
prsProgramData   OBJECT-TYPE
   SYNTAX   OCTET STRING {SIZE(23)}
   ACCESS   read-write
   STATUS   optional
   DESCRIPTION
      "<Definition> An OER encoded string of reference parameters for
       exchanges between Traffic Management and a PRS.

       The parameter values in this string are:
          priorityRequestTimeToLiveValue       INTEGER (0..65535)
          priorityRequestReserviceClass1Time   INTEGER (0..65535)
          priorityRequestReserviceClass2Time   INTEGER (0..65535)
          priorityRequestReserviceClass3Time   INTEGER (0..65535)
          priorityRequestReserviceClass4Time   INTEGER (0..65535)
          priorityRequestReserviceClass5Time   INTEGER (0..65535)
          priorityRequestReserviceClass6Time   INTEGER (0..65535)
          priorityRequestReserviceClass7Time   INTEGER (0..65535)
          priorityRequestReserviceClass8Time   INTEGER (0..65535)
          priorityRequestReserviceClass9Time   INTEGER (0..65535)
          priorityRequestReserviceClass10Time  INTEGER (0..65535)"
::= { priorityRequestMessages 7 }

END -- PRS-MIB1
```

 © 2008 AASHTO / ITE / NEMA

Section 4
OBJECT DEFINITIONS FOR COORDINATOR

This section defines the information requirements of the Coordinator portion an SCP System.

The clauses below present the objects in lexigraphical order of their OBJECT IDENTIFIERS, which correspond to their physical location within the global naming tree. Most of the objects defined in NTCIP 1211 v01 reside under the "scp" node of the global naming tree. To aid in object management, the SCP node has been subdivided into logical categories; each defined by a node under the SCP node. The individual objects are then located under the appropriate node.

Nodes should not be confused with conformance groups, which are defined in Annex B. 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 may contain both "global" and "scp" specific objects.

4.1 COORDINATOR MIB

```
CO-MIB1 DEFINITIONS ::= BEGIN

-- The following Management Information Base represents the data elements that
-- would reside in a Coordinator that implements SCP.

IMPORTS

      EntryStatus, OwnerString, OerString,
      Byte, UByte, Short, Ushort, Long, Ulong
      FROM NEMA-SMI

      scp
      FROM NEMA-SMI2

      OBJECT-TYPE
      FROM RFC-1212

      splitNumber, splitPhase
      FROM  SCP-MIB1

      TrueFalse ::= INTEGER (0 | 255)
```

4.2 PRIORITY STRATEGY PARAMETERS

```
-- This group of object represents the output from the prioritization process
-- and the parameters that Coordinator acts upon.

priorityStrategyServer ::= { scp 3 }
```

© 2008 AASHTO / ITE / NEMA

4.2.1 Maximum Service Requests to Consider

```
maxStrategyRequestsToConsider OBJECT-TYPE
      SYNTAX    INTEGER (1..10)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
            " This object represents the number of rows in the
            priorityStrategyRequestedTable that the CO will consider when it
            trys to determine which strategies to execute.  The number of rows
            in the table is fixed at 10 but an implemention is only requred to
            act upon the first row."
::= { priorityStrategyServer 1 }
```

4.2.2 Priority Strategy Requested Table

```
priorityStrategyRequestedTable   OBJECT-TYPE
      SYNTAX   SEQUENCE OF priorityStrategyRequestedEntry
      ACCESS   not-accessible
      STATUS   optional
      DESCRIPTION
          "A static table containing the parameters associated with executing
          a priority strategy. The number of rows in this table is equal to
          10."
::= { priorityStrategyServer 2 }
```

4.2.2.1 Priority Strategy Requested Entry

```
priorityStrategyRequestedEntry   OBJECT-TYPE
      SYNTAX   PriorityStrategyRequestedEntry
      ACCESS   not-accessible
      STATUS   optional
      DESCRIPTION
            "This object defines the parameters that are associated with a
            service request."
      INDEX   { priorityStrategyRequestedNumber }
::= { priorityStrategyRequestedTable 1 }

priorityStrategyRequestedEntry ::= SEQUENCE {
      priorityStrategyRequestNumber                      INTEGER,
      priorityStrategyRequested                          INTEGER,
      priorityStrategyRequestedTimeOfServiceDesired      INTEGER,
      priorityStrategyRequestedTimeOfEstimatedDeparture INTEGER,
      priorityStrategyRequestStatusInCO                  INTEGER}
```

4.2.2.1.1 Priority Strategy Request Number

```
priorityStrategyRequestNumber OBJECT-TYPE
      SYNTAX    INTEGER (1..10)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
              " This object represents the index used to reference rows in
              the priorityStrategyRequestedTable. "
::= { priorityStrategyRequestedEntry 1 }
```

 © 2008 AASHTO / ITE / NEMA

4.2.2.1.2 Priority Strategy Requested

```
priorityStrategyRequested OBJECT-TYPE
      SYNTAX    INTEGER (0..255)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          " This object represents a strategy being requested by the PRS. A
          value of zero (0) shall indicate that none is request."
::= { priorityStrategyRequestedEntry 2 }
```

4.2.2.1.3 Priority Strategy Requested Time of Service Desired

```
priorityStrategyRequestedTimeOfServiceDesired   OBJECT-TYPE
      SYNTAX INTEGER -- (0..4294967295)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object represents the estimated time of service desired
          expressed as global time.  This value is received from the PRS via
          the prsServiceRequest message's
          priorityRequestTimeOfServiceDesiredInPRS."
::= { priorityStrategyRequestedEntry 3 }
```

4.2.2.1.4 Priority Strategy Requested Time of Estimated Departure

```
priorityStrategyRequestedTimeOfEstimatedDeparture     OBJECT-TYPE
      SYNTAX   INTEGER -- (0..4294967295)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object represents the estimated time of departure expressed as
          global time.  This value is received from the PRS via the
          prsServiceRequest message's
          priorityRequestTimeOfServiceDesiredInPRS."
::= { priorityStrategyRequestedEntry 4 }
```

4.2.2.1.5 Priority Strategy Request Status (CO)

```
priorityStrategyRequestStatusInCO   OBJECT-TYPE
      SYNTAX    INTEGER { idleNotValid (1),
                          readyQueued (2),
                          readyOverridden (3),
                          activeProcessing (4),
                          activeCancel (5),
                          activeOverride (6),
                          activeNotOverridden (7),
                          closedCanceled (8),
                          reserviceError (9),
                          closedTimeToLiveError (10),
                          closedTimerError (11),
                          closedStrategyError (12),
                          closedCompleted (13),
                          activeAdjustNotneeded (14),
                          closedFlash (15) }
```

© 2008 AASHTO / ITE / NEMA

```
ACCESS   read-only
STATUS   optional
DESCRIPTION
    "This object provides an image of priorityRequestStatusInPRS that
    was sent by the PRS or a response to readyQueued, activeCancel, or
    activeOverride.
      idleNotValid        PRG determined that the request does not
                          contain valid data
      readyQueued         PRG has validated the request but is waiting
                          for the CO to activate
      readyOverridden     CO has overridden the request
      activeProcessing    CO is processing the requested strategy
      activeCancel        PRS has asked that request be canceled
      activeOverride      PRS has asked that request be overriddden
      activeNotOverridden CO did not process the requested override
      closedCanceled      CO has canceled the request
      reserviceError        PRG determined that the request came too
                            soon after a previous request
      closedTimeToLiveError PRG determined that TSD exceeds the time to
                            live
      closedTimerError      CO determined that the requested times could
                            not be met
      closedStrategyError   CO determined that the requested strategy
                            was not valid.
      closedCompleted       CO has completed the requested strategy
                            request
      activeAdjustNotNeeded CO determined that the request could be met
                            with current timing and adjustment was not
                            needed
      closedFlash           CO determined that the controller was
                            operating in flash.

      Upon receipt of a readyQueued, the CO may change the status to:
            activeProcessing
            closedTimerError
            closedStrategyError
            activeAdjustNotNeeded
            closedFlash

      Upon receipt of a activeCancel, the CO may change the status to:
            closedCanceled
            closedCompleted

      Upon receipt of a activeOverride, the CO may change the status to:
            activeNotOverridden
            readyOverridden
            closedCompleted

      The CO does NOT take any action when one of the following status
      is received from the PRS:
            idleNotValid
            readyOverridden
            activeProcessing
            activeNotOverridden
            closedCanceled
            reserviceError
            closedTimeToLiveError
            closedTimerError
```

 © 2008 AASHTO / ITE / NEMA

```
                    closedStrategyError
                    closedCompleted
                    activeAdjustNotNeeded
                    closedFlash

                Note: A change of status is predicated on the "busy" signal being
                set."

      DEFVAL { idleNotValid }
::= { priorityStrategyRequestedEntry 5}
```

4.2.3 Coodinator Busy

```
coBusy OBJECT-TYPE
      SYNTAX    TrueFalse
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object is used to indicate to a PRS that the CO is
          initializing or otherwise processing and when TRUE, the information
          included with it in a prsServiceRequest response is not valid.  It
          is used to synchonize exchanges between a PRS and CO.

          When the PRS is acting as manager and the CO is acting as an agent,
          coBusy is set TRUE when the PRS performs a SetRequest
          prsServiceRequest.

          When the CO is acting as a manager and the PRS is acting as the
          agent, the value shall always be set to False."

::= { priorityStrategyServer 3 }

-- The following set of objects defines a table of priority strategy
-- parameters and a list of per unit parameters.  In the table, each row in
-- the table represents data associated with a particular strategy.
```

4.2.4 Priority Strategies Max

```
priorityStrategiesMax   OBJECT-TYPE
      SYNTAX    INTEGER (1..255)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "The maximum number of priority strategies supported by this
          entity."
::= { priorityStrategyServer 4}
```

4.2.5 Priority Strategies Table

```
priorityStrategyTable   OBJECT-TYPE
      SYNTAX   SEQUENCE OF PriorityStrategyEntry
      ACCESS   not-accessible
      STATUS   optional
      DESCRIPTION
          "A static table containing the parameters associated with each
          priority strategy. The number of rows in this table is equal to the
          priorityStrategiesMax object."
```

© 2008 AASHTO / ITE / NEMA

```
::= { priorityStrategyServer 5 }
```

4.2.5.1 Priority Strategy Table Entry

```
priorityStrategyEntry   OBJECT-TYPE
      SYNTAX   PriorityStrategyEntry
      ACCESS   not-accessible
      STATUS   optional
      DESCRIPTION
          "Parameters associated with a specific priority strategy."
      INDEX   { priorityStrategyNumber }
::= { priorityStrategyTable 1 }

PriorityStrategyEntry ::= SEQUENCE {
            priorityStrategyNumber         INTEGER,
            priorityStrategyServicePhases  OCTET STRING,
            priorityStrategyPhaseOmits     OCTET STRING,
            priorityStrategyPedOmits       OCTET STRING,
            priorityStrategyDescription    OCTET STRING }
```

4.2.5.1.1 Priority Strategy Number

```
priorityStrategyNumber OBJECT-TYPE
      SYNTAX    INTEGER (1..255)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object serves as the index in the priorityStrategyTable."
::= { priorityStrategyEntry 1}
```

4.2.5.1.2 Priority Strategy Service Phases

```
priorityStrategyServicePhases OBJECT-TYPE
      SYNTAX    OCTET STRING
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object defines the phase or phases that represent the movement
          on which priority service is desired. Each octet (binary value)
          within the octet string contains an SCP.phaseNumber.

          There can only be one phase per ring. All defined Service Phases
          shall be able to time concurrently. It is assumed that any possible
          concurrent phase are allowed to be serviced in order with no backing
          up unless otherwise permitted."
      DEFVAL { "" }
::= { priorityStrategyEntry 2}
```

4.2.5.1.3 Priority Strategy Phase Omits

```
priorityStrategyPhaseOmits OBJECT-TYPE
      SYNTAX    OCTET STRING
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object represents the phase(s) that are to be omitted during
          the priority service. Each octet (binary value) within the octet
          string contains an SCP.phaseNumber.  The CO shall ignore any omit
          applied to the coordinated phase(s). These phase omits are logically
          "OR'ed with any other phase omits that may be active."
```

 © 2008 AASHTO / ITE / NEMA

```
      DEFVAL { "" }
::= { priorityStrategyEntry 3}
```

4.2.5.1.4 Priority Strategy Ped Omits

```
priorityStrategyPedOmits OBJECT-TYPE
      SYNTAX    OCTET STRING
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object represents the pedestrian movement(s) that are to be
          omitted during the priority service.  Each octet (binary value)
          within the octet string contains an SCP.phaseNumber.  These ped
          omits are logically "OR'ed with any other ped omits that may be
          active.  These ped omits do not apply to non-actuated phases."
      DEFVAL { "" }
::= { priorityStrategyEntry 4}
```

4.2.5.1.5 Priority Strategy Description

```
priorityStrategyDescription   OBJECT-TYPE
      SYNTAX    OCTET STRING (SIZE (0..40))
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object represents a text description of the strategy.  It is
          intended to convey to priority service requesters what signals or
          movements are given priority.  It should be defined in language that
          is domain neutral."
::= { priorityStrategyEntry 5}

-- The following set of objects represent per unit parameters that apply to
-- all strategies. It includes a set of database variables that relate to
-- an extension of the splitTable as defined in NTCIP 1202.
```

4.2.6 Priority Strategy Extension to Split Table

```
priorityStrategyExtensionToSplitTable     OBJECT-TYPE
      SYNTAX    SEQUENCE OF PriorityStrategyExtensionToSplitEntry
      ACCESS   not-accessible
      STATUS   optional
      DESCRIPTION
          "A table containing priority strategy parameters that are associated
          with each split and each phase of that split.

          This is an extension to the splitTable as defined in the NTCIP 1202,
          Object Definitions for Actuated Signal Controllers. "
::= { priorityStrategyServer 6 }
```

4.2.6.1 Priority Strategy Extension to Split Entry

```
priorityStrategyExtensionToSplitEntry     OBJECT-TYPE
      SYNTAX    PriorityStrategyExtensionToSplitEntry
      ACCESS   not-accessible
      STATUS   optional
      DESCRIPTION
          "A table containing priority strategy parameters that are associated
          with each split and each phase of that split."
```

© 2008 AASHTO / ITE / NEMA

```
      INDEX {splitNumber, splitPhase }
::= { priorityStrategyExtensionToSplitTable 1}

PriorityStrategyExtensionToSplitEntry ::= SEQUENCE {
      priorityStrategyMaximumReductionTime INTEGER,
      priorityStrategyMaximumExtensionTime INTEGER }
```

4.2.6.1.1 Priority Strategy Maximum Reduction Time

```
priorityStrategyMaximumReductionTime      OBJECT-TYPE
      SYNTAX   INTEGER (0..255)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object represents the maximum number of seconds that the split
          of a phase can be reduced by during a priority strategy.

          This effective value may be reduced by the CO based on the operating
          phase minimum service requirements.

      NOTE - The definition of a phases minimum service is dependent on
      whether it is actuated or non-actuated, whether the ped clear terminates
      before the yellow or not, whether variable initial is active or not,
      whether it is a coord phase or not, and whether it is in a single or
      multi-ring configuration. "

      DEFVAL { 0 }
::= { priorityStrategyExtensionToSplitEntry 1 }
```

4.2.6.1.2 Priority Strategy Maximum Extension Time

```
priorityStrategyMaximumExtensionTime      OBJECT-TYPE
      SYNTAX   INTEGER (0..255)
      ACCESS   read-only
      STATUS   optional
      DESCRIPTION
          "This object represents the maximum number of seconds that the split
          of a phase can be extended during a priority strategy.

          The effective value may be modified by the CO based on time gained
          from one cycle of reductions to the respective non-priority phases."

      DEFVAL { 0 }
::= { priorityStrategyExtensionToSplitEntry 2 }
```

4.2.7 Priority Strategy Default Coordination Pattern

```
priorityStrategyDefaultCoordPattern       OBJECT-TYPE
      SYNTAX   INTEGER (1..253)
      ACCESS   read-write
      STATUS   optional
      DESCRIPTION
            "This object represents the coordination pattern timing parameters
            to use if the controller is not operating in a coordinated mode. "
::= { priorityStrategyServer 7 }
```

 © 2008 AASHTO / ITE / NEMA

4.2.8 Priority Service Request and Response Block Objects

```
--The following block object definition is defined for exchanges between a PRS
--and a CO.

priorityStrategyMessages ::= { scp 4}
```

4.2.8.1 Service Request

```
prsServiceRequest   OBJECT-TYPE
   SYNTAX   OCTET STRING {SIZE(110)}
   ACCESS   read-write
   STATUS   optional
   DESCRIPTION
      "<Definition> An OER encoded string of reference parameters for
       exchanges between a CO and a PRS.

       The parameter values of this OID when transfering information from a
       PRS to a CO are:
        priorityRequestServiceStrategyNumber.1          INTEGER (1..255)
        priorityRequestTimeOfServiceDesiredInPRS.1      INTEGER (0..4294967295)
        priorityRequestTimeOfEstimatedDepartureInPRS.1  INTEGER (0..4294967295)
        priorityRequestStatusInPRS.1                    INTEGER (1..255)
        priorityRequestServiceStrategyNumber.2          INTEGER (1..255)
        priorityRequestTimeOfServiceDesiredInPRS.2      INTEGER (0..4294967295)
        priorityRequestTimeOfEstimatedDepartureInPRS.2  INTEGER (0..4294967295)
        priorityRequestStatusInPRS.2                    INTEGER (1..255)
        priorityRequestServiceStrategyNumber.3          INTEGER (1..255)
        priorityRequestTimeOfServiceDesiredInPRS.3      INTEGER (0..4294967295)
        priorityRequestTimeOfEstimatedDepartureInPRS.3  INTEGER (0..4294967295)
        priorityRequestStatusInPRS.3                    INTEGER (1..255)
        priorityRequestServiceStrategyNumber.4          INTEGER (1..255)
        priorityRequestTimeOfServiceDesiredInPRS.4      INTEGER (0..4294967295)
        priorityRequestTimeOfEstimatedDepartureInPRS.4  INTEGER (0..4294967295)
        priorityRequestStatusInPRS.4                    INTEGER (1..255)
        priorityRequestServiceStrategyNumber.5          INTEGER (1..255)
        priorityRequestTimeOfServiceDesiredInPRS.5      INTEGER (0..4294967295)
        priorityRequestTimeOfEstimatedDepartureInPRS.5  INTEGER (0..4294967295)
        priorityRequestStatusInPRS.5                    INTEGER (1..255)
        priorityRequestServiceStrategyNumber.6          INTEGER (1..255)
        priorityRequestTimeOfServiceDesiredInPRS.6      INTEGER (0..4294967295)
        priorityRequestTimeOfEstimatedDepartureInPRS.6  INTEGER (0..4294967295)
        priorityRequestStatusInPRS.6                    INTEGER (1..255)
        priorityRequestServiceStrategyNumber.7          INTEGER (1..255)
        priorityRequestTimeOfServiceDesiredInPRS.7      INTEGER (0..4294967295)
        priorityRequestTimeOfEstimatedDepartureInPRS.7  INTEGER (0..4294967295)
        priorityRequestStatusInPRS.7                    INTEGER (1..255)
        priorityRequestServiceStrategyNumber.8          INTEGER (1..255)
        priorityRequestTimeOfServiceDesiredInPRS.8      INTEGER (0..4294967295)
        priorityRequestTimeOfEstimatedDepartureInPRS.8  INTEGER (0..4294967295)
        priorityRequestStatusInPRS.8                    INTEGER (1..255)
        priorityRequestServiceStrategyNumber.9          INTEGER (1..255)
        priorityRequestTimeOfServiceDesiredInPRS.9      INTEGER (0..4294967295)
        priorityRequestTimeOfEstimatedDepartureInPRS.9  INTEGER (0..4294967295)
        priorityRequestStatusInPRS.9                    INTEGER (1..255)
        priorityRequestServiceStrategyNumber.10         INTEGER (1..255)
        priorityRequestTimeOfServiceDesiredInPRS.10     INTEGER (0..4294967295)
        priorityRequestTimeOfEstimatedDepartureInPRS.10 INTEGER (0..4294967295)
        priorityRequestStatusInPRS.10                   INTEGER (1..255)
        prsBusy                                         INTEGER (0..1)

       The parameter values for this OID when transfering information from a
       CO to a PRS are:
        priorityRequestServiceStrategyNumber.1                INTEGER (1..255)
        priorityStrategyRequestedTimeOfServiceDesired.1       INTEGER (0..4294967295)
        priorityStrategyRequestedTimeOfEstimatedDeparture.1   INTEGER (0..4294967295)
        priorityStrategyRequestStatusInCO.1                   INTEGER(1..255)
        priorityRequestServiceStrategyNumber.2                INTEGER (1..255)
```

© 2008 AASHTO / ITE / NEMA

```
          priorityStrategyRequestedTimeOfServiceDesired.2        INTEGER (0..4294967295)
          priorityStrategyRequestedTimeOfEstimatedDeparture.2    INTEGER (0..4294967295)
          priorityStrategyRequestStatusInCO.2                    INTEGER(1..255)
          priorityRequestServiceStrategyNumber.3                 INTEGER (1..255)
          priorityStrategyRequestedTimeOfServiceDesired.3        INTEGER (0..4294967295)
          priorityStrategyRequestedTimeOfEstimatedDeparture.3    INTEGER (0..4294967295)
          priorityStrategyRequestStatusInCO.3                    INTEGER(1..255)
          priorityRequestServiceStrategyNumber.4                 INTEGER (1..255)
          priorityStrategyRequestedTimeOfServiceDesired.4        INTEGER (0..4294967295)
          priorityStrategyRequestedTimeOfEstimatedDeparture.4    INTEGER (0..4294967295)
          priorityStrategyRequestStatusInCO.4                    INTEGER(1..255)
          priorityRequestServiceStrategyNumber.5                 INTEGER (1..255)
          priorityStrategyRequestedTimeOfServiceDesired.5        INTEGER (0..4294967295)
          priorityStrategyRequestedTimeOfEstimatedDeparture.5    INTEGER (0..4294967295)
          priorityStrategyRequestStatusInCO.5                    INTEGER(1..255)
          priorityRequestServiceStrategyNumber.6                 INTEGER (1..255)
          priorityStrategyRequestedTimeOfServiceDesired.6        INTEGER (0..4294967295)
          priorityStrategyRequestedTimeOfEstimatedDeparture.6    INTEGER (0..4294967295)
          priorityStrategyRequestStatusInCO.6                    INTEGER(1..255)
          priorityRequestServiceStrategyNumber.7                 INTEGER (1..255)
          priorityStrategyRequestedTimeOfServiceDesired.7        INTEGER (0..4294967295)
          priorityStrategyRequestedTimeOfEstimatedDeparture.7    INTEGER (0..4294967295)
          priorityStrategyRequestStatusInCO.7                    INTEGER(1..255)
          priorityRequestServiceStrategyNumber.8                 INTEGER (1..255)
          priorityStrategyRequestedTimeOfServiceDesired.8        INTEGER (0..4294967295)
          priorityStrategyRequestedTimeOfEstimatedDeparture.8    INTEGER (0..4294967295)
          priorityStrategyRequestStatusInCO.8                    INTEGER(1..255)
          priorityRequestServiceStrategyNumber.9                 INTEGER (1..255)
          priorityStrategyRequestedTimeOfServiceDesired.9        INTEGER (0..4294967295)
          priorityStrategyRequestedTimeOfEstimatedDeparture.9    INTEGER (0..4294967295)
          priorityStrategyRequestStatusInCO.9                    INTEGER(1..255)
          priorityRequestServiceStrategyNumber.10                INTEGER (1..255)
          priorityStrategyRequestedTimeOfServiceDesired.10       INTEGER (0..4294967295)
          priorityStrategyRequestedTimeOfEstimatedDeparture.10   INTEGER (0..4294967295)
          priorityStrategyRequestStatusInCO.10                   INTEGER(1..255)"
::= { priorityStrategyMessages 2 }
```

4.3 SCP BLOCK OBJECTS

```
scpBlock  OBJECT IDENTIFIER ::= { scp 5 }

-- This object is an identifier used to group all objects for
-- support of SCP Block Upload and Download activities.
```

4.3.1 SCP Block Get Control

```
scpBlockGetControl   OBJECT-TYPE
   SYNTAX   OCTET STRING (SIZE(2..12))
   ACCESS   read-write
   STATUS   optional
   DESCRIPTION
      "<Definition> An OER encoded string of reference
       parameters for SCP Block Uploads. The parameter
       values in this string are:
         scpBlockDataType    INTEGER (0..255)
         scpBlockDataID      INTEGER (0..255)
         scpBlockIndex1      INTEGER (0..255) if needed
         scpBlockQuantity1   INTEGER (0..255) if needed
         scpBlockIndex2      INTEGER (0..255) if needed
         scpBlockQuantity2   INTEGER (0..255) if needed
         scpBlockIndex3      INTEGER (0..255) if needed
         scpBlockQuantity3   INTEGER (0..255) if needed
         scpBlockIndex4      INTEGER (0..255) if needed
         scpBlockQuantity4   INTEGER (0..255) if needed
```

 © 2008 AASHTO / ITE / NEMA

```
          scpBlockIndex5      INTEGER (0..255) if needed
          scpBlockQuantity5   INTEGER (0..255) if needed

        A GET of scpBlockData shall use values currently
        in this object to define the data to be returned.

        A SET of this object shall be evaluated for validity
        and Error Status of badValue(3) be returned for the
        following conditions:
         1) scpBlockDataType is not supported
         2) scpBlockDataID is not supported
         3) scpBlockIndex1 is zero or not supported
         4) scpBlockQuantity1 is zero or
            scpBlockIndex1 + scpBlockQuantity1 - 1 is not
            supported
         5) scpBlockIndex2 is zero or not supported
         6) scpBlockQuantity2 is zero or
            scpBlockIndex2 + scpBlockQuantity2) - 1 is not
            supported
         7) scpBlockIndex3 is zero or not supported
         8) scpBlockQuantity3 is zero or
            scpBlockIndex3 + scpBlockQuantity3) - 1 is not
            supported
         9) scpBlockIndex4 is zero or not supported
        10) scpBlockQuantity4 is zero or
            scpBlockIndex4 + scpBlockQuantity4) - 1 is not
            supported
        11) scpBlockIndex5 is zero or not supported
        12) scpBlockQuantity5 is zero or
            scpBlockIndex5 + scpBlockQuantity5) - 1 is not
            supported
        13) if the SET length is zero or incorrect for
            scpBlockDataType & scpBlockDataID
        14) if the GetResponse length for a GET on
            scpBlockData using maximum data field sizes
            would exceed a local limitation
        When this validity check fails, scpBlockErrorStatus
        shall be set equal to the Bullet Value above that
        generated the error.
        <DescriptiveName> NTCIP-1211::SCP.scpBlockGetControl
        <DataConceptType> Data Frame
        <Unit> "
::= { scpBlock 1 }
```

4.3.2 SCP Block Data

```
scpBlockData   OBJECT-TYPE
   SYNTAX   OCTET STRING (SIZE(2..484))
   ACCESS  read-write
   STATUS  optional
   DESCRIPTION
      "<Definition> An OER encoded string used for
       uploading and downloading SCP parameters. See
       SECTION 3 for encoding and decoding the block.

       A SET on this object shall require the use of
       'dbCreateTransaction' defined in NTCIP 1201
       Clause 2.3.1.
```

© 2008 AASHTO / ITE / NEMA Copy Per MIB Distribution Notice

```
        A SET of this object shall be evaluated for validity
        and Error Status of badValue(3) be returned for the
        following conditions:
         1) scpBlockDataType is not supported
         2) scpBlockDataID is not supported
         3) scpBlockIndex1 is zero or not supported
         4) scpBlockQuantity1 is zero or
            scpBlockIndex1 + scpBlockQuantity1 - 1 is not
            supported
         5) scpBlockIndex2 is zero or not supported
         6) scpBlockQuantity2 is zero or
            scpBlockIndex2 + scpBlockQuantity2) - 1 is not
            supported
         7) scpBlockIndex3 is zero or not supported
         8) scpBlockQuantity3 is zero or
            scpBlockIndex3 + scpBlockQuantity3) - 1 is not
            supported
         9) scpBlockIndex4 is zero or not supported
        10) scpBlockQuantity4 is zero or
            scpBlockIndex4 + scpBlockQuantity4) - 1 is not
            supported
        11) scpBlockIndex5 is zero or not supported
        12) scpBlockQuantity5 is zero or
            scpBlockIndex5 + scpBlockQuantity5) - 1 is not
            supported
        13) if the SET length is zero or incorrect for
            scpBlockDataType & scpBlockDataID
        14) if the SET (SEQUENCE OF) value is incorrect.
        When this validity check fails, scpBlockErrorStatus
        shall be set equal to the Bullet Value above that
        generated the error.

        A SET that includes an unsupported value for a
        supported data element shall return an Error Status
        of badValue(3) and scpBlockErrorStatus shall be set
        equal to:
           (data Sequence # * 100) + data Element #

        A SET that includes a non-zero or non-null value in
        the position of an unsupported data element shall
        return an Error Status of badValue(3) and
        scpBlockErrorStatus shall be set equal to:
           (data Sequence # * 100) + data Element #

        A GET on this object shall use values currently in
        scpBlockGetControl to define the data to be returned.
        When scpBlockGetControl has invalid data, an Error
        Status of badValue(3) shall be returned.

        A GET shall return a zero or null value in the position
        of an unsupported object.
        <DescriptiveName> NTCIP-1211::SCP.scpBlockData
        <DataConceptType> Data Frame
        <Unit> "
::= { scpBlock 2 }
```

 © 2008 AASHTO / ITE / NEMA

4.3.3 SCP Block Error Status

```
scpBlockErrorStatus   OBJECT-TYPE
   SYNTAX   INTEGER (0..65535)
   ACCESS   read-only
   STATUS   optional
   DESCRIPTION
      "<Definition> This object defines the data element
       within scpBlockGetControl or scpBlockData that
       caused a badValue(3) ErrorStatus.

       This object should equal zero after any successful
       SET to scpBlockGetControl or scpBlockData.
       <DescriptiveName> NTCIP-1211::SCP.scpBlockErrorStatus
       <DataConceptType> Data Element"
::= { scpBlock 3 }



END -- CO-MIB1
```

4.4 COORDINATION PROCESSOR BLOCK OBJECT DEFINITIONS

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

The Data Type octet (scpBlockDataType) provides for the definition of both NTCIP Standard and Device Proprietary data blocks. NTCIP Standard Data Blocks shall use an 'scpBlockDataType' of zero. Device Proprietary Data Blocks shall use an 'scpBlockDataType' 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 (scpBlockDataID) provides for definition of included data parameters. NCTIP Standard Data Blocks shall include an 'scpBlockDataID' as listed below:

scpBlockData-dataID Definitions		
DataID	Name	Description
0x00	CoStrategyPlanBlock	Defines Priority Phase, Omits, etc.
0x01	CoStrategySplitsBlock	Defines Max Extensions and Reductions
0x02–0x7F		Reserved For NTCIP SCP Usage

4.4.1 Coordinator Strategy Plan Block Definition

```
-- scpBlockData values for standard Block
-- coStrategyPlanBlock Data shall be as follows:

CoStrategyPlanBlock ::= SEQUENCE
   {
    scpBlockDataType     INTEGER (0..255),  -- 0x00 standard block
    scpBlockDataID       INTEGER (0..255),  -- 0x00 CoStrategyPlanBlock
    scpBlockIndex1       INTEGER (0..255),  -- priorityStrategyNumber
    scpBlockQuantity1    INTEGER (0..255),  -- ## of strategies
```

© 2008 AASHTO / ITE / NEMA

```
    --      for (
    --           x = scpBlockIndex1;
    --           x < (scpBlockIndex1 + scpBlockQuantity1);
    --           x++)

    data    SEQUENCE OF CoStrategyPlanBlockData
   }

CoStrategyPlanBlockData ::= SEQUENCE
  { priorityStrategyServicePhases.x             OCTET STRING,
    priorityStategyPhaseOmits.x                 OCTET STRING,
    priorityStategyPedOmits.x                   OCTET STRING,
    priorityStrategyDescription.x               OCTET STRING (SIZE 0..40) }
```

4.4.2 Coordinator Strategy Splits Block Object Definition

```
-- scpBlockData values for standard Block
-- CoStrategySplitsBlock Data shall be as follows:

CoStrategySplitsBlock ::= SEQUENCE
   {
    scpBlockDataType     INTEGER (0..255),  -- 0x00 standard block
    scpBlockDataID       INTEGER (0..255),  -- 0x01 CoStrategySplitsBlock
    scpBlockIndex1       INTEGER (0..255),  -- splitPhase
    scpBlockQuantity1    INTEGER (0..255),  -- ## of phases
    scpBlockIndex2       INTEGER (0..255),  -- splitNumber
    scpBlockQuantity2    INTEGER (0..255),  -- ## of splits

    -- for (
    --      y = scpBlockIndex2;
    --      y < (scpBlockIndex2 + scpBlockQuantity2);
    --      y++)
    --      for (
    --           x = scpBlockIndex1;
    --           x < (scpBlockIndex1 + scpBlockQuantity1);
    --           x++)

    data    SEQUENCE OF CoStrategySplitsBlockData
   }

CoStrategySplitsBlockData ::= SEQUENCE
   {
    priorityStrategyMaximumReductionTime.y.x          INTEGER,
    priorityStrategyMaximumExtensionTime.y.x          INTEGER
   }
```

 © 2008 AASHTO / ITE / NEMA

Annex A
PRIORITY REQUEST SERVER INFORMATION PROFILE REQUIREMENTS LIST

(Normative)

A.1 INTRODUCTION

This annex provides the Profile Requirements List (PRL) for implementations of the SCP – Priority Request Server Information in the form of proforma. A Profile Implementation Conformance Specification (PICS) for an implementation is generated by an implementer or supplier by indicating the appropriate level of support provided by an implementation.

To claim conformance with this profile, an implementation shall satisfy the mandatory conformance requirements of this profile.

An implementation's completed PRL is called the PICS. The PICS states which capabilities and options of the protocol have been implemented. The following can use the PICS:

a. The protocol implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight.

b. The supplier and user, as a detailed indication of the capabilities of the implementation.

c. The user, as a basis for initially checking the possibility of interworking with another implementation (note that, while interworking can never be guaranteed, failure to do so can often be predicted from incompatible PICSs).

d. A user, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

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 1200-series standards. 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. The device shall 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

© 2008 AASHTO / ITE / NEMA Copy Per PRL Reproduction Notice

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.

A.2 NOTATION

The following notations and symbols are used to indicate status and conditional status within NTCIP 1211 v01.

A.2.1 Type Symbols

The following symbols are used to indicate type:

Symbol	Type
C	Control Object – use of 'dbCreateTransaction' in NTCIP 1201 v02 Clause 2.3.1 shall NOT delay a SET to this object.
P	Parameter Object – use of 'dbCreateTransaction' in NTCIP 1201 v02 Clause 2.3.1 to SET this object is optional.
P2	Parameter Object – use of 'dbCreateTransaction' in NTCIP 1201 v02 Clause 2.3.1 to SET this object is mandatory.
S	Status / Information Object – this object is read only therefore a SET is not permitted.

A.2.2 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.2.3 Conditional Status Notation

The following predicate notations is used:

 © 2008 AASHTO / ITE / NEMA

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 is supported. In the simplest case, <predicate> is the identifying tag of a single item.

A.2.4 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 s upport column:

A.3 PRIORITY REQUEST CONFORMANCE GROUPS

The Conformance Group definitions for Priority Request Servers are defined in this clause. A Priority Request Server has multiple functions; thus, Conformance Groups are defined for each function.

The following table lists functional requirements for a Priority Request Server, and asks if the listed features have been implemented.

Ref	Areas	Clause of Profile	Status	Support
A.4	Priority Request Server Conformance Group	NTCIP 1211 – 3.1	M	Yes
A.4.1	Priority Request Server Message Conformance Group	NTCIP 1211 – 3.2	M	Yes
A.5	Configuration Conformance Group	NTCIP 1201 – 2.2	M	Yes
A.6	Database Management Conformance Group	NTCIP 1201 – 2.3	M	Yes
A.7	Report Conformance Group	NTCIP 1201 – 2.5	O	Yes / No
A.8	Auxiliary I/O Group	NTCIP 1201 – 2.5	N/A	No
A.9	PMPP Group	NTCIP 1201 – 2.6	O	Yes / No
A.10	SNMP Group	rfc1213	M	Yes
A.11	System Group	rfc1213	M	Yes
A.12	SFMP Group	NTCIP 1103 – A.4	N/A	No
A.13	STMP Group	NTCIP 1103 – A.5	N/A	No
A.14	Logical Name Group	NTCIP 1103 – A.6	O	Yes / No
A.15	Trap Management Group	NTCIP 1103 – A.7-9	N/A	No
A.16	Security Group	NTCIP 1103 – A.10	M	Yes
A.17	RS232 Group	rfc1317	O	Yes / No
A.18	HDLC Group	rfc1381	O	Yes / No
A.19	Interfaces Group	rfc1213	O	Yes / No
A.20	IP Group	rfc1213	O	Yes / No
A.21	ICMP Group	rfc1213	O	Yes / No
A.22	TCP Group	rfc1213	O	Yes / No
A.23	UDP Group	rfc1213	O	Yes / No
A.24	Ethernet Group	rfc1643	O	Yes / No
A.25	Time Management Group	NTCIP 1201 – 2.2	M	Yes

NOTE — If the PRS is logically integrated into an Actuated Signal Controller, a second set of the Conformance Groups (A.6 through A.25) would not be needed.

Priority Request Server devices shall adhere to the conformance requirements specified in the as a minimum to claim compliance to NTCIP 1211. Additional objects or groups may be supported without being non-compliant with Priority Request Server 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 Priority Request Server objects or NTCIP.

© 2008 AASHTO / ITE / NEMA Copy Per PRL Reproduction Notice

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

A.4 PRIORITY REQUEST SERVER CONFORMANCE GROUP

The Priority Request Server Conformance Group shall consist of the following objects:

PRIORITY REQUEST SERVER CONFORMANCE GROUP						
NTCIP 1211 Clause	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
3.1	**Priority Request Server Group**	---	M	Yes	---	---
3.1.1	priorityRequestTable	--	M	Yes	----	---
3.1.1.1	priorityRequestEntry	--	M	Yes	---	---
3.1.1.1.1	priorityRequestEntryNumber	S	M	Yes	1–10	
3.1.1.1.2	priorityRequestID	S	M	Yes	0–255	
3.1.1.1.3	priorityRequestVehicleID	S	M	Yes	string	
3.1.1.1.4	priorityRequestVehicleClassType	S	M	Yes	1–10	
3.1.1.1.5	priorityRequestVehicleClassLevel	S	M	Yes	1–10	
3.1.1.1.6	priorityRequestServiceStrategyNumber	S	M	Yes	1–255	
3.1.1.1.7	priorityRequestTimeOfServiceDesired	S	M	Yes	1–65535	
3.1.1.1.8	priorityRequestTimeOfEstimated Departure	S	M	Yes	1–65535	
3.1.1.1.9	priorityRequestStatusInPRS	S	M	Yes	1–15	
	idleNotValid	--	---	Yes	---	---
	readyQueued	--	---	Yes	---	---
	readyOverriden	--	---	Yes	---	---
	activeProcessing	--	---	Yes	---	---
	activeCancel	--	---	Yes	---	---
	activeOverride	--	---	Yes	---	---
	activeNotOverriden	--	---	Yes	---	---
	closedCanceled	--	---	Yes	---	---
	reserviceError	--	---	Yes	---	---
	closedTimerToLiveError	--	---	Yes	---	---
	closedTimerError	--	---	Yes	---	---
	closedStreategyError	--	---	Yes	---	---
	closedCompleted	--	---	Yes	---	---
	activeAdjustNotNeeded	--	---	Yes	---	---
	closedFlash	--	---	Yes	---	---
3.1.1.1.10	priorityRequestTimeOfMessage	S	M	Yes	0–4294967295	
3.1.1.1.11	priorityRequestTimeToLive	S	M	Yes	0–4294967295	
3.1.1.1.12	priorityRequestTimeOfServiceDesired InPRS	S	M	Yes	0–4294967295	
3.1.1.1.13	priorityRequestTimeOfEstimated DepartureInPRS	S	M	Yes	0–4294967295	
3.1.2	prsBusy	S	M	Yes	0 | 255	
3.1.3	priorityRequestTimeToLiveValue	P	M	Yes	0–65535	
3.1.4	priorityRequestReserviceTimer	P	M	Yes	0–65535	
3.1.5	priorityRequestReserviceClass1Time	P	M	Yes	0–65535	
3.1.6	priorityRequestReserviceClass2Time	P	M	Yes	0–65535	
3.1.7	priorityRequestReserviceClass3Time	P	M	Yes	0–65535	
3.1.8	priorityRequestReserviceClass4Time	P	M	Yes	0–65535	
3.1.9	priorityRequestReserviceClass5Time	P	M	Yes	0–65535	
3.1.10	priorityRequestReserviceClass6Time	P	M	Yes	0–65535	
3.1.11	priorityRequestReserviceClass7Time	P	M	Yes	0–65535	
3.1.12	priorityRequestReserviceClass8Time	P	M	Yes	0–65535	
3.1.13	priorityRequestReserviceClass9Time	P	M	Yes	0–65535	
3.1.14	priorityRequestReserviceClass10Time	P	M	Yes	0–65535	

A.4.1 Priority Request Server Message Conformance Group

The PRS Message Conformance Group shall consist of the following objects:

 © 2008 AASHTO / ITE / NEMA

MESSAGE CONFORMANCE GROUP						
NTCIP 1211 Clause	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
3.2.1	prgPriorityRequest	C	M	Yes	string	
3.2.2	prgPriorityUpdate	C	M	Yes	string	
3.2.3	prgPriorityStatusControl	C	M	Yes	string	
3.2.4	prgPriorityStatusBuffer	C	M	Yes	string	
3.2.5	prgPriorityCancel	C	M	Yes	string	
3.2.6	prgPriorityClear	C	M	Yes	string	
3.2.7	prsProgramData	C	M	Yes	string	

A.5 CONFIGURATION CONFORMANCE GROUP

The Configuration Conformance Group shall consist of the following objects:

CONFIGURATION CONFORMANCE GROUP						
NTCIP 1201 Clause	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
2.2	**Global Config Group**	--	O	Yes / No	-----	----
2.2.1	globalSetIDParmeter	S	2.2 : O	Yes / No	0–65535	
2.2.2	globalMaxModules	S	2.2 : M	Yes	0–255	
2.2.3	globalModuleTable	--	2.2 : M	Yes	---	---
	moduleTableEntry	--	2.2 : M	Yes	---	---
2.2.3.1	moduleNumber	S	2.2 : M	Yes	1–255	
2.2.3.2	moduleDeviceNode	S	2.2 : M	Yes	OID	
2.2.3.3	moduleMake	S	2.2 : M	Yes	String	
2.2.3.4	moduleModel	S	2.2 : M	Yes	String	
2.2.3.5	moduleVersion	S	2.2 : M	Yes	String	
2.2.3.6	moduleType	S	2.2 : M	Yes	1–3	
	other(1)	--	---	Yes / No	---	---
	hardware(2)	--	---	Yes / No	---	---
	software(3)	--	---	Yes / No	---	---

© 2008 AASHTO / ITE / NEMA
Copy Per PRL Reproduction Notice

A.6 DATABASE MANAGEMENT CONFORMANCE GROUP

The Database Management Conformance Group shall consist of the following objects:

DATABASE MANAGEMENT CONFORMANCE GROUP						
NTCIP 1201 Clause	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
2.3	**DB Management Conformance Group**	--	M	Yes	-----	----
2.3.1	dbCreateTransaction	C	M	Yes	1,2,3,6	
2.3.6	dbVerifyStatus	S	M	Yes	0–2	
2.3.7	dbVerifyError	S	M	Yes	string	

A.7 REPORT CONFORMANCE GROUP

The Report Conformance Group shall consist of the following objects:

REPORT CONFORMANCE GROUP						
NTCIP 1201 Clause	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
2.5	**Report Conformance Group**	--	O	Yes / No	-----	----
2.5.1	maxEventClasses	S	2.5 : M	Yes	0–255	
2.5.2	eventClassTable	--	2.5 : M	Yes	---	---
	eventClassEntry	--	2.5 : M	Yes	---	---
2.5.2.1	eventClassNumber	S	2.5 : M	Yes	1–255	
2.5.2.2	eventClassLimit	P	2.5 : M	Yes	0–255	
2.5.2.3	eventClassClearTime	P	2.5 : M	Yes	counter	
2.5.2.4	eventClassDescription	P	2.5 : O	Yes	string	
2.5.2.5	eventClassNumRowsInLog	S	2.5 : M	Yes	0–255	
2.5.3	maxEventLogConfigs	S	2.5 : M	Yes	0–65535	
2.5.4	eventLogConfigTable	--	2.5 : M	Yes	---	---
	eventLogConfigEntry	--	2.5 : M	Yes	---	---
2.5.4.1	eventConfigID	S	2.5 : M	Yes	1–65535	
2.5.4.2	eventConfigClass	P	2.5 : M	Yes	1–255	
2.5.4.3	eventConfigMode	P	2.5 : M	Yes	1–6	
	other(1)	--	---	Yes / No	---	---
	onChange(2)	--	---	Yes / No	---	---
	greaterThanValue(3)	--	---	Yes / No	---	---
	smallerThanValue(4)	--	---	Yes / No	---	---
	hystersisBound(5)	--	---	Yes / No	---	---
	periodic(6)	--	---	Yes / No	---	---
	andedWithValue(7)	--	---	Yes / No	---	---
2.5.4.4	eventConfigCompareValue	P	2.5 : M	Yes	INT	
2.5.4.5	eventConfigCompareValue2	P	2.5 : M	Yes	INT	
2.5.4.6	eventConfigCompareOID	P	2.5 : M	Yes	OID	
2.5.4.7	eventConfigLogOID	P	2.5 : O	Yes	OID	
2.5.4.8	eventConfigAction	P	2.5 : M	Yes	1–3	
2.5.5	maxEventLogSize	S	2.5 : M	Yes	0–65535	
2.5.6	eventLogTable	--	2.5 : M	Yes	---	---
	eventLogEntry	--	2.5 : M	Yes	---	---
2.5.6.1	eventLogClass	S	2.5 : M	Yes	0–255	
2.5.6.2	eventLogNumber	S	2.5 : M	Yes	1–255	
2.5.6.3	eventLogID	S	2.5 : M	Yes	0–65535	
2.5.6.4	eventLogTime	S	2.5 : M	Yes	counter	
2.5.6.5	eventLogValue	S	2.5 : M	Yes	opaque (1)	

(1) For any NTCIP Compliant SCP implementation the value of the 'eventLogValue' OID has been changed from Opaque to Opaque (SIZE(0..40)).

Copy Per PRL Reproduction Notice © 2008 AASHTO / ITE / NEMA

A.8 AUXIO GROUP

The AUXIO Group shall consist of the following objects:

SFMP GROUP						
NTCIP 2101 Clause	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
2.8	**AUXIO GROUP**	--	O	No	----	---

A.9 PMPP GROUP

The PMPP Group shall consist of the following objects:

PMPP GROUP						
NTCIP 1201 Clause	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
2.6	**PMPP GROUP**	--	O	Yes / No	----	---
2.6.1	maxGroupAddress	S	2.6 : M	Yes	0..255	
2.6.2	hdlcGroupAddressTable	--	2.6 : M	Yes	---	---
	hdlcGroupAddressEntry	--	2.6 : M	Yes	---	---
2.6.2.1	hdlcGroupAddressIndex	S	2.6 : M	Yes	0..255	
2.6.2.2	hdlcGroupAddress	P	2.6 : M	Yes	INT	deprecated
2.6.2.3	hdlcGroupAddressNumber	P	2.6 : M	Yes	INT	

A.10 SNMP GROUP

The SNMP Group shall consist of the following objects:

SNMP GROUP						
rfc 1213	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
snmp	**SNMP GROUP**	--	O	Yes / No	---	
snmp.1	snmpInPkts	S	snmp : M	Yes	Counter	
snmp.2	snmpOutPkts	S	snmp : M	Yes	Counter	
snmp.3	snmpInBadVersions	S	snmp : M	Yes	Counter	
snmp.4	snmpInBadCommunityNames	S	snmp : M	Yes	Counter	
snmp.5	snmpInBadCommunityUses	S	snmp : M	Yes	Counter	
snmp.6	snmpInASNParseErrs	S	snmp : M	Yes	Counter	
snmp.8	snmpInTooBigs	S	snmp : M	Yes	Counter	
snmp.9	snmpInNoSuchNames	S	snmp : M	Yes	Counter	
snmp.10	snmpInBadValues	S	snmp : M	Yes	Counter	
snmp.11	snmpInReadOnlys	S	snmp : M	Yes	Counter	
snmp.12	snmpInGenErrs	S	snmp : M	Yes	Counter	
snmp.13	snmpInTotalReqVars	S	snmp : O	Yes / No	Counter	
snmp.14	snmpInTotalSetVars	S	snmp : O	Yes / No	Counter	
snmp.15	snmpInGetRequests	S	snmp : M	Yes	Counter	
snmp.16	snmpInGetNexts	S	snmp : M	Yes	Counter	
snmp.17	snmpInSetRequests	S	snmp : M	Yes	Counter	
snmp.18	snmpInGetResponses	S	snmp : M	Yes	Counter	
snmp.19	snmpInTraps	S	snmp : M	Yes	Counter	
snmp.20	snmpOutTooBigs	S	snmp : M	Yes	Counter	
snmp.21	snmpOutNoSuchNames	S	snmp : M	Yes	Counter	
snmp.22	snmpOutBadValues	S	snmp : M	Yes	Counter	
snmp.24	snmpOutGenErrs	S	snmp : M	Yes	Counter	
snmp.25	snmpOutGetRequests	S	snmp : M	Yes	Counter	
snmp.26	snmpOutGetNexts	S	snmp : M	Yes	Counter	
snmp.27	snmpOutSetRequests	S	snmp : M	Yes	Counter	

© 2008 AASHTO / ITE / NEMA Copy Per PRL Reproduction Notice

SNMP GROUP						
rfc 1213	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
snmp.28	snmpOutGetResponses	S	snmp : M	Yes	Counter	
snmp.29	snmpOutTraps	S	snmp : M	Yes	Counter	
snmp.30	snmpEnableAuthenTraps	P	snmp : M	Yes	INT	

SNMP GROUP						
NTCIP 1103 Clause	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
A.3	snmp-maxPacketSize	S	snmp : M	Yes	484–65535	

A.11 SYSTEM GROUP

The System Group shall consist of the following objects:

SYSTEM GROUP						
rfc 1213	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
system	**SYSTEM GROUP**	--	O	Yes / No	---	
system 1	sysDescr	S	system : M	Yes	string	
system 2	sysObjectID	S	system : M	Yes	OID	
system 3	sysUpTime	S	system : M	Yes	TimeTicks	
system 4	sysContact	P	system : M	Yes	string	
system 5	sysName	P	system : M	Yes	atring	
system 6	sysLocation	P	system : M	Yes	string	
system 7	sysServices	S	system : M	Yes	0..127	

A.12 SFMP GROUP

The SFMP Group shall consist of the following objects:

SFMP GROUP						
NTCIP 1103 Clause	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
A.4	**Objects for SFMP**	--	O	No	----	---

A.13 STMP GROUP

The STMP Group shall consist of the following objects:

STMP GROUP						
NTCIP 1103 Clause	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
A.5	**Objects for STMP**	--	O	Yes / No	---	---
A.5.1	Dynamic Object Definition	--	A.5 : M	Yes	---	---
	dynObjDef	--	A.5 : M	Yes	---	---
	dynObjEntry	--	A.5 : M	Yes	---	---
	dynObjNumber	S	A.5 : M	Yes	1..13	

© 2008 AASHTO / ITE / NEMA

STMP GROUP						
NTCIP 1103 Clause	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
	dynObjIndex	S	A.5 : M	Yes	1..255	
	dynObjVariable	C	A.5 : M	Yes	OID	
	dynObjOwner	NA	NA	No	---	deprecated
	dynObjStatus	NA	NA	No	---	deprecated
A.5.2	Dynamic Object Data	--	NA	No	---	deprecated
A.5.3	Dynamic Object Configuration	--	A.5 : M	Yes	---	---
	dynObjConfigTable	--	A.5 : M	Yes	---	---
	dynObjConfigEntry	--	A.5 : M	Yes	---	---
	dynObjConfigOwner	C	A.5 : M	Yes	string	
	dynObjConfigStatus	C	A.5 : M	Yes	1..3	
A.5.4	STMP Statistics	--	A.5 : M	Yes	--	
.1	stmp-InPkts	S	A.5 : M	Yes	counter	
.2	stmp-OutPkts	S	A.5 : M	Yes	counter	
.6	stmp-InParseErrs	S	A.5 : M	Yes	counter	
.8	stmp-InTooBigs	S	A.5 : M	Yes	counter	
.9	stmp-InNoSuchNames	S	A.5 : M	Yes	counter	
.10	stmp-InBadValues	S	A.5 : M	Yes	counter	
.11	stmp-InReadOnlys	S	A.5 : M	Yes	counter	
.12	stmp-InGenErrs	S	A.5 : M	Yes	counter	
.15	stmp-InGetRequests	S	A.5 : M	Yes	counter	
.16	stmp-InGetNexts	S	A.5 : M	Yes	counter	
.17	stmp-InSetRequests	S	A.5 : M	Yes	counter	
.18	stmp-InGetResponses	S	A.5 : M	Yes	counter	
.20	stmp-OutTooBigs	S	A.5 : M	Yes	counter	
.21	stmp-OutNoSuchNames	S	A.5 : M	Yes	counter	
.22	stmp-OutBadValues	S	A.5 : M	Yes	counter	
.23	stmp-OutReadOnly	S	A.5 : M	Yes	counter	
.24	stmp-OutGenError	S	A.5 : M	Yes	counter	
.25	stmp-OutGetRequests	S	A.5 : M	Yes	counter	
.26	stmp-OutGetNexts	S	A.5 : M	Yes	counter	
.27	stmp-OutSetRequests	S	A.5 : M	Yes	counter	
.28	stmp-OutGetResponses	S	A.5 : M	Yes	counter	
.29	stmp-OutTrapResponses	S	A.5 : M	Yes	counter	
.31	stmp-InSetRequestsNoReply	S	A.5 : M	Yes	counter	
.32	stmp-InSetResponses	S	A.5 : M	Yes	counter	
.33	stmp-InErrorResponses	S	A.5 : M	Yes	counter	
.34	stmp-OutSetRequestsNoReply	S	A.5 : M	Yes	counter	
.35	stmp-OutSetResponses	S	A.5 : M	Yes	counter	
.36	stmp-OutErrorResponses	S	A.5 : M	Yes	counter	
A.5.5	STMP Configuration	--	A.5 : M	Yes	---	---
.1	dynamicObjectPersistence	P	A.5 : M	Yes	0–65535	
.2	dynamicObjectTable-ConfigID	S	A.5 : M	Yes	0–65535	

A.14 LOGICAL NAME GROUP

The Logical Name Group shall consist of the following objects:

LOGICAL NAME GROUP						
NTCIP 1103 Clause	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
A.6	**Objects for Logical Names**	--	O	Yes / No	----	---
	logicalNameTranslationTable-maxEntries	S	A.6 : M	Yes	0..255	
A.6.2	logicalNameTranslationTable	--	A.6 : M	Yes	---	---
	logicalNameTranslationEntry	--	A.6 : M	Yes	---	---
A.6.2.1	logicalNameTranslation-index	S	A.6 : M	Yes	INT	
A.6.2.2	logicalNameTranslation-logicalName	P	A.6 : M	Yes	string	
A.6.2.3	logicalNameTranslation-networkAddress	P	A.6 : M	Yes	string	

© 2008 AASHTO / ITE / NEMA Copy Per PRL Reproduction Notice

LOGICAL NAME GROUP						
NTCIP 1103 Clause	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
A.6.2.4	logicalNameTranslation-status	C	A.6 : M	Yes	INT	

A.15 TRAP MANAGEMENT GROUP

The Trap Management Group shall consist of the following objects:

TRAP MANAGEMENT GROUP						
NTCIP 1103 Clause	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
A.7	**Objects for Trap Management**	--	O	Yes / No	----	---
A.7.1	trapMgmt-managerName	P	A.7 : M	Yes	string	
A.7.2	trapMgmt-applicationProtocol	P	A.7 : M	Yes	INT	
A.7.3	trapMgmt-transportProfile	P	A.7 : M	Yes	INT	
A.7.4	trapMgmt-mode	P	A.7 : M	Yes	INT	
A.7.5	trapMgmt-repeatMin	P	A.7 : M	Yes	0..255	
A.7.6	trapMgmt-repeatDelta	P	A.7 : M	Yes	0..255	
A.7.7	trapMgmt-maxRetries	P	A.7 : M	Yes	0..254	
A.7.8	trapMgmt-tries	S	A.7 : M	Yes	0.255	
A.7.9	trapMgmt-time	S	A.7 : M	Yes	Counter	
A.7.10	trapMgmt-counter	S	A.7 : M	Yes	Counter	
A.8	Trap Data	--	A.7 : M	Yes	---	---
A.8.1	trapEvent	--	A.7 : M	Yes	---	---
A.8.2	trapMgmt-log	S	A.7 : M	Yes	string	
A.9	Generic Trap Objects	--	A.7 : M	Yes	---	---
A.9.1	agentHealth-coldStarts	S	A.7 : M	Yes	Counter	
A.9.2	agentHealth-warmStarts	S	A.7 : M	Yes	Counter	
A.9.3	agentHealth-linkDowns	S	A.7 : M	Yes	Counter	
A.9.4	agentHealth-linkUps	S	A.7 : M	Yes	Counter	
A.9.5	agentHealth-authenticationFailures	S	A.7 : M	Yes	Counter	
A.9.6	agentHealth-linkDownData	S	A.7 : M	Yes	INT	
A.9.7	agentHealth-linkUpData	S	A.7 : M	Yes	INT	

A.16 SECURITY GROUP

The Security Group shall consist of the following objects:

SECURITY GROUP						
NTCIP 1103 Clause	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
A.10	**SECURITY GROUP**	--	O	Yes / No	----	---
A.10.1	communityNameAdmin	P	A.10 : M	Yes	string	
A.10.2	communityNamesMax	S	A.10 : M	Yes	1..255	
A.10.3	communityNameTable	--	A.10 : M	Yes	---	---
	communityNameTableEntry	--	A.10 : M	Yes	---	---
A.10.3.1	communityNameIndex	S	A.10 : M	Yes	1..255	
A.10.3.2	communityNameUser	P	A.10 : M	Yes	string	
A.10.3.3	communityNameAccessMask	P	A.10 : M	Yes	gauge	

© 2008 AASHTO / ITE / NEMA

A.17 RS232 GROUP

The RS232 Group shall consist of the following objects:

RS232 GROUP						
rfc 1317	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
rs232	**RS232 GROUP**	--	O	Yes / No	----	---
rs232.1	rs232Number	S	rs232 : M	Yes	INT	
rs232.2	rs232PortTable	--	rs232 : M	Yes	---	---
	rs232PortEntry	--	rs232 : M	Yes	---	---
rs232.2.1	rs232PortIndex	S	rs232 : M	Yes	INT	
rs232.2.2	rs232PortType	S	rs232 : M	Yes	1..5	
	other(1)	--	---	Yes / No	---	---
	rs232(2)	--	---	Yes / No	---	---
	rs422(3)	--	---	Yes / No	---	---
	rs423(4)	--	---	Yes / No	---	---
	v35(5)	--	---	Yes / No	---	---
rs232.2.3	rs232PortInSigNumber	S	rs232 : O	Yes / No	INT	
rs232.2.4	rs232PortOutSigNumber	S	rs232 : O	Yes / No	INT	
rs232.2.5	rs232PortInSpeed	P	rs232 : M	Yes	INT	
rs232.2.6	rs232PortOutSpeed	P	rs232 : M	Yes	INT	
rs232.3	rs232AsyncPortTable	--	rs232 : M	Yes	---	---
	rs232AsyncPortEntry	--	rs232 : M	Yes	---	---
rs232.3.1	rs232AsyncPortIndex	S	rs232 : M	Yes	INT	
rs232.3.2	rs232AsyncPortBits	P	rs232 : O	Yes / No	5..8	
	five(5)	--	---	Yes / No	---	---
	six(6)	--	---	Yes / No	---	---
	seven(7)	--	---	Yes / No	---	---
	eight(8)	--	---	Yes / No	---	---
rs232.3.3	rs232AsyncPortStopBits	P	rs232 : O	Yes / No	1..4	
	one(1)	--	---	Yes / No	---	---
	two(2)	--	---	Yes / No	---	---
	one-and-half(3)	--	---	Yes / No	---	---
	dynamic(4)	--	---	Yes / No	---	---
rs232.3.4	rs232AsyncPortParity	P	rs232 : O	Yes / No	1..5	
	none(1)	--	---	Yes / No	---	---
	odd(2)	--	---	Yes / No	---	---
	even(3)	--	---	Yes / No	---	---
	mark(4)	--	---	Yes / No	---	---
	space(5)	--	---	Yes / No	---	---
rs232.3.5	rs232AsyncPortAutobaud	P	rs232 : O	Yes / No	1..2	
	enabled(1)	--	---	Yes / No	---	---
	disabled(2)	--	---	Yes / No	---	---
rs232.3.6	rs232AsyncPortParityErrs	S	rs232 : O	Yes / No	Counter	
rs232.3.7	rs232AsyncPortFramingErrs	S	rs232 : M	Yes	Counter	
rs232.3.8	rs232AsyncPortOverrunErrs	S	rs232 : M	Yes	Counter	

A device may require the rs232PortInSpeed and rs232PortOutSpeed to be the same value. Therefore, a SET of rs232PortInSpeed may automatically SET rs232PortOutSpeed to the same value and vice-versa.

© 2008 AASHTO / ITE / NEMA Copy Per PRL Reproduction Notice

A.18 HDLC GROUP

The HDLC Group shall consist of the following objects:

HDLC GROUP						
rfc 1381	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
lapb	**HDLC GROUP**	--	O	Yes / No	----	---
lapb.1	lapbAdmnTable	--	lapb : M	Yes	---	---
	lapbAdmnEntry	--	lapb : M	Yes	---	---
lapb.1.1	lapbAdmnIndex	S	lapb : M	Yes	IfIndexType	
lapb.1.2	lapbAdmnStationType	P	lapb : O	Yes / No	1..3	
	dte(1)	--	---	Yes / No	---	---
	dce(2)	--	---	Yes / No	---	---
	dxe(3)	--	---	Yes / No	---	---
lapb.1.3	lapbAdmnControlField	P	lapb : O	Yes / No	1..2	
	modulo8(1)	--	---	Yes / No	---	---
	modulo128(2)	--	---	Yes / No	---	---
lapb.1.4	lapbAdmnTransmitN1FrameSize	P	lapb : M	Yes	P Integer	
lapb.1.5	lapbAdmnReceiveN1FrameSize	P	lapb : M	Yes	P Integer	
lapb.1.6	lapbAdmnTransmitKWindowSize	P	lapb : O	Yes / No	1..127	
lapb.1.7	lapbAdmnReceiveKWindowSize	P	lapb : O	Yes / No	1..127	
lapb.1.8	lapbAdmnN2RxmitCount	P	lapb : O	Yes / No	0..65535	
lapb.1.9	lapbAdmnT1AckTimer	P	lapb : M	Yes	P Integer	
lapb.1.10	lapbAdmnT2AckDelayTimer	P	lapb : M	Yes	P Integer	
lapb.1.11	lapbAdmnT3DisconnectTimer	P	lapb : M	Yes	P Integer	
lapb.1.12	lapbAdmnT4IdleTimer	P	lapb : M	Yes	P Integer	
lapb.1.13	lapbAdmnActionInitiate	P	lapb : O	Yes / No	1..5	
	sendSABM(1)	--	---	Yes / No	---	---
	sendDISC(2)	--	---	Yes / No	---	---
	sendDM(3)	--	---	Yes / No	---	---
	none(4)	--	---	Yes / No	---	---
	other(5)	--	---	Yes / No	---	---
lapb.1.14	lapbAdmnActionRecvDM	P	lapb : O	Yes / No	1..3	
	sendSABM(1)	--	---	Yes / No	---	---
	sendDISC(2)	--	---	Yes / No	---	---
	other(3)	--	---	Yes / No	---	---
lapb.2	lapbOperTable	--	lapb : M	Yes	----	---
	lapbOperEntry	--	lapb : M	Yes	----	---
lapb.2.1	lapbOperIndex	S	lapb : M	Yes	IfIndexType	
lapb.2.2	lapbOperStationType	S	lapb : O	Yes / No	1..3	
	dte(1)	--	---	Yes / No	---	---
	dce(2)	--	---	Yes / No	---	---
	dxe(3)	--	---	Yes / No	---	---
lapb.2.3	lapbOperControlField	S	lapb : O	Yes / No	1..2	
	modulo8(1)	--	---	Yes / No	---	---
	modulo128(2)	--	---	Yes / No	---	---
lapb.2.4	lapbOperTransmitN1FrameSize	S	lapb : O	Yes / No	P Integer	
lapb.2.5	lapbOperReceiveN1FrameSize	S	lapb : O	Yes / No	P Integer	
lapb.2.6	lapbOperTransmitKWindowSize	S	lapb : O	Yes / No	1..127	
lapb.2.7	lapbOperReceiveKWindowSize	S	lapb : O	Yes / No	1..127	
lapb.2.8	lapbOperN2RxmitCount	S	lapb : O	Yes / No	0..65535	
lapb.2.9	lapbOperT1AckTimer	S	lapb : O	Yes / No	P Integer	
lapb.2.10	lapbOperT2AckDelayTimer	S	lapb : O	Yes / No	P Integer	
lapb.2.11	lapbOperT3DisconnectTimer	S	lapb : O	Yes / No	P Integer	
lapb.2.12	lapbOperT4IdleTimer	S	lapb : O	Yes / No	P Integer	
lapb.2.13	lapbOperPortId	S	lapb : M	Yes	OID	
lapb.2.14	lapbOperProtocolVersionID	S	lapb : O	Yes / No	OID	

'P Integer = Positive Integer

 © 2008 AASHTO / ITE / NEMA

A.19 INTERFACES GROUP

The Interfaces Group shall consist of the following objects:

INTERFACES GROUP						
rfc 1213	Object Name	Object Type	Object Status	Object Support	Allowed Values	Supported Values
if	**INTERFACES GROUP**	--	O	Yes / No	----	---
if.1	ifNumber	S	if : M	Yes	---	---
if.2	ifTable	--	if : M	Yes	---	---
	ifEntry	--	if : M	Yes	---	---
if.2.1	ifIndex	S	if : M	Yes	INT	
if.2.2	ifDescr	S	if : M	Yes	string	
if.2.3	ifType	S	if : M	Yes	INT	
if.2.4	ifMtu	S	if : M	Yes	INT	
if.2.5	ifSpeed	S	if : M	Yes	gauge	
if.2.6	ifPhysAddress	S	if : M	Yes	PhysAddress	
if.2.7	ifAdminStatus	C	if : O	Yes / No	INT	
if.2.8	ifOperStatus	S	if : M	Yes	INT	
if.2.9	ifLastChange	S	if : O	Yes / No	TimeTicks	
if.2.10	ifInOctets	S	if : O	Yes / No	counter	
if.2.11	ifInUcastPkts	S	if : O	Yes / No	counter	
if.2.12	ifInNUcastPkts	S	if : O	Yes / No	counter	
if.2.13	ifInDiscards	S	if : O	Yes / No	counter	
if.2.14	ifInErrors	S	if : O	Yes / No	counter	
if.2.15	ifInUnknownProtos	S	if : O	Yes / No	counter	
if.2.16	ifOutOctets	S	if : O	Yes / No	counter	
if.2.17	ifOutUcastPkts	S	if : O	Yes / No	counter	
if.2.18	ifOutNUcastPkts	S	if : O	Yes / No	counter	
if.2.19	ifOutDiscards	S	if : O	Yes / No	counter	
if.2.20	ifOutErrors	S	if : O	Yes / No	counter	
if.2.21	ifOutQLen	S	if : O	Yes / No	gauge	
if.2.22	ifSpecific	S	if : O	Yes / No	OID	

A.20 IP GROUP

The IP Group shall consist of the following objects:

IP GROUP						
rfc 1213	Object Name	Object Type	Object Status	Object Support	Allowed Values	Supported Values
ip	**IP GROUP**	--	O	Yes / No	----	---
ip.1	ipForwarding	P	ip : M	Yes	INT	
ip.2	ipDefaultTTL	P	ip : M	Yes	INT	
ip.3	ipInReceives	S	ip : M	Yes	counter	
ip.4	ipInHdrErrors	S	ip : M	Yes	counter	
ip.5	ipInAddrErrors	S	ip : M	Yes	counter	
ip.6	ipForwDatagrams	S	ip : M	Yes	counter	
ip.7	ipInUnknownProtos	S	ip : M	Yes	counter	
ip.8	ipInDiscards	S	ip : M	Yes	counter	
ip.9	ipInDelivers	S	ip : M	Yes	counter	
ip.10	ipOutRequests	S	ip : M	Yes	counter	
ip.11	ipOutDiscards	S	ip : M	Yes	counter	
ip.12	ipOutNoRoutes	S	ip : M	Yes	counter	
ip.13	ipReasmTimeout	S	ip : M	Yes	counter	
ip.14	ipReasmReqds	S	ip : M	Yes	counter	
ip.15	ipReasmOKs	S	ip : M	Yes	counter	
ip.16	ipReasmFails	S	ip : M	Yes	counter	
ip.17	ipFragOKs	S	ip : M	Yes	counter	
ip.18	ipFragFails	S	ip : M	Yes	counter	

© 2008 AASHTO / ITE / NEMA

IP GROUP						
rfc 1213	Object Name	Object Type	Object Status	Object Support	Allowed Values	Supported Values
ip.19	ipFragCreates	S	ip : M	Yes	counter	
ip.20	ipAddrTable	--	ip : M	Yes	---	
ip.20.1	ipAddrEntry	--	ip : M	Yes	---	
ip.20.1.1	ipAdEntAddr	S	ip : M	Yes	IpAddress	
ip.20.1.2	ipAdEntIfIndex	S	ip : M	Yes	INT	
ip.20.1.3	ipAdEntNetMask	S	ip : M	Yes	IpAddress	
ip.20.1.4	ipAdEntBcastAddr	S	ip : M	Yes	INT	
ip.20.1.5	ipAdEntReasmMaxSize	S	ip : M	Yes	INT	
ip.21	ipRouteTable	--	ip : M	Yes	---	
ip.21.1	ipRouteEntry	--	ip : M	Yes	---	
ip.21.1.1	ipRouteDest	P	ip : M	Yes	IpAddress	
ip.21.1.2	ipRouteIfIndex	P	ip : M	Yes	INT	
ip.21.1.3	ipRouteMetric1	P	ip : M	Yes	INT	
ip.21.1.4	ipRouteMetric2	P	ip : M	Yes	INT	
ip.21.1.5	ipRouteMetric3	P	ip : M	Yes	INT	
ip.21.1.6	ipRouteMetric4	P	ip : M	Yes	INT	
ip.21.1.7	ipRouteNextHop	P	ip : M	Yes	IpAddress	
ip.21.1.8	ipRouteType	P	ip : M	Yes	INT	
ip.21.1.9	ipRouteProto	S	ip : M	Yes	INT	
ip.21.1.10	ipRouteAge	P	ip : M	Yes	INT	
ip.21.1.11	ipRouteMask	P	ip : M	Yes	IpAddress	
ip.21.1.12	ipRouteMetric5	P	ip : M	Yes	INT	
ip.21.1.13	ipRouteInfo	S	ip : M	Yes	OID	
ip.22	ipNetToMediaTable	--	ip : M	Yes	---	
ip.22.1	ipNetToMediaEntry	--	ip : M	Yes	---	
ip.22.1.1	ipNetToMediaIfIndex	P	ip : M	Yes	INT	
ip.22.1.2	ipNetToMediaPhysAddress	P	ip : M	Yes	PhysAddress	
ip.22.1.3	ipNetToMediaNetAddress	P	ip : M	Yes	IpAddress	
ip.22.1.4	ipNetToMediaType	P	ip : M	Yes	INT	
ip.23	ipRoutingDiscards	S	ip : M	Yes	counter	

A.21 ICMP GROUP

The ICMP Group shall consist of the following objects:

ICMP GROUP						
rfc 1213	Object Name	Object Type	Object Status	Object Support	Allowed Values	Supported Values
icmp	**ICMP GROUP**	--	O	Yes / No	----	---
icmp.1	icmpInMsgs	S	icmp : M	Yes	counter	
icmp.2	icmpInErrors	S	icmp : M	Yes	counter	
icmp.3	icmpInDestUnreachs	S	icmp : M	Yes	counter	
icmp.4	icmpInTimeExcds	S	icmp : M	Yes	counter	
icmp.5	icmpInParmProbs	S	icmp : M	Yes	counter	
icmp.6	icmpInSrcQuenchs	S	icmp : M	Yes	counter	
icmp.7	icmpInRedirects	S	icmp : M	Yes	counter	
icmp.8	icmpInEchos	S	icmp : M	Yes	counter	
icmp.9	icmpInEchoReps	S	icmp : M	Yes	counter	
icmp.10	icmpInTimestamps	S	icmp : M	Yes	counter	
icmp.11	icmpInTimestampReps	S	icmp : M	Yes	counter	
icmp.12	icmpInAddrMasks	S	icmp : M	Yes	counter	
icmp.13	icmpInAddrMaskReps	S	icmp : M	Yes	counter	
icmp.14	icmpOutMsgs	S	icmp : M	Yes	counter	
icmp.15	icmpOutErrors	S	icmp : M	Yes	counter	

 © 2008 AASHTO / ITE / NEMA

ICMP GROUP						
rfc 1213	Object Name	Object Type	Object Status	Object Support	Allowed Values	Supported Values
icmp.16	icmpOutDestUnreachs	S	icmp : M	Yes	counter	
icmp.17	icmpOutTimeExcds	S	icmp : M	Yes	counter	
icmp.18	icmpOutParmProbs	S	icmp : M	Yes	counter	
icmp.19	icmpOutSrcQuenchs	S	icmp : M	Yes	counter	
icmp.20	icmpOutRedirects	S	icmp : M	Yes	counter	
icmp.21	icmpOutEchos	S	icmp : M	Yes	counter	
icmp.22	icmpOutEchoReps	S	icmp : M	Yes	counter	
icmp.23	icmpOutTimestamps	S	icmp : M	Yes	counter	
icmp.24	icmpOutTimestampReps	S	icmp : M	Yes	counter	
icmp.25	icmpOutAddrMasks	S	icmp : M	Yes	counter	
icmp.26	icmpOutAddrMaskReps	S	icmp : M	Yes	counter	

A.22 TCP GROUP

The TCP Group shall consist of the following objects:

TCP GROUP						
rfc 1213	Object Name	Object Type	Object Status	Object Support	Allowed Values	Supported Values
tcp	**TCP GROUP**	--	O	Yes / No	----	---
tcp.1	tcpRtoAlgorithm	S	tcp : M	Yes	INT	
tcp.2	tcpRtoMin	S	tcp : M	Yes	INT	
tcp.3	tcpRtoMax	S	tcp : M	Yes	INT	
tcp.4	tcpMaxConn	S	tcp : M	Yes	INT	
tcp.5	tcpActiveOpens	S	tcp : M	Yes	counter	
tcp.6	tcpPassiveOpens	S	tcp : M	Yes	counter	
tcp.7	tcpAttemptFails	S	tcp : M	Yes	counter	
tcp.8	tcpEstabResets	S	tcp : M	Yes	counter	
tcp.9	tcpCurrEstab	S	tcp : M	Yes	counter	
tcp.10	tcpInSegs	S	tcp : M	Yes	counter	
tcp.11	tcpOutSegs	S	tcp : M	Yes	counter	
tcp.12	tcpRetransSegs	S	tcp : M	Yes	counter	
tcp.13	tcpConnTable	--	tcp : M	Yes	---	
tcp.13.1	tcpConnEntry	--	tcp : M	Yes	---	
tcp.13.1.1	tcpConnState	C	tcp : M	Yes	INT	
tcp.13.1.2	tcpConnLocalAddress	S	tcp : M	Yes	IpAddress	
tcp.13.1.3	tcpConnLocalPort	S	tcp : M	Yes	INT	
tcp.13.1.4	tcpConnRemAddress	S	tcp : M	Yes	IpAddress	
tcp.13.1.5	tcpConnRemPort	S	tcp : M	Yes	INT	
tcp.14	tcpInErrs	S	tcp : M	Yes	counter	
tcp.15	tcpOutRsts	S	tcp : M	Yes	counter	

© 2008 AASHTO / ITE / NEMA
Copy Per PRL Reproduction Notice

A.23 UDP GROUP

The UDP Group shall consist of the following objects:

UDP GROUP						
rfc 1213	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
udp	**UDP GROUP**	--	O	Yes / No	----	---
udp.1	udpInDatagrams	S	udp : M	Yes	INT	
udp.2	udpNoPorts	S	udp : M	Yes	INT	
udp.3	udpInErrors	S	udp : M	Yes	INT	
udp.4	udpOutDatagrams	S	udp : M	Yes	INT	
udp.5	udpTable	--	udp : M	Yes	---	
udp.5.1	udpEntry	--	udp : M	Yes	---	
udp.5.1.1	udpLocalAddress	S	udp : M	Yes	IpAddress	
udp.5.1.2	udpLocalPort	S	udp : M	Yes	INT	

A.24 ETHERNET GROUP

The Ethernet Group shall consist of the following objects:

ETHERNET GROUP						
rfc 1643	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
dot3	**ETHERNET GROUP**	--	O	Yes / No	----	---
dot3.2	dot3StatsTable	--	dot3 : M	Yes	---	---
dot3.2.1	dot3StatsEntry	--	dot3 : M	Yes	---	---
dot3.2.1.1	dot3StatsIndex	S	dot3 : M	Yes	INT	
dot3.2.1.2	dot3StatsAlignmentErrors	S	dot3 : M	Yes	counter	
dot3.2.1.3	dot3StatsFCSErrors	S	dot3 : M	Yes	counter	
dot3.2.1.4	dot3StatsSingleCollisionFrames	S	dot3 : M	Yes	counter	
dot3.2.1.5	dot3StatsMultipleCollisionFrames	S	dot3 : M	Yes	counter	
dot3.2.1.6	dot3StatsSQETestErrors	S	dot3 : M	Yes	counter	
dot3.2.1.7	dot3StatsDeferredTransmissions	S	dot3 : M	Yes	counter	
dot3.2.1.8	dot3StatsLateCollisions	S	dot3 : M	Yes	counter	
dot3.2.1.9	dot3StatsExcessiveCollisions	S	dot3 : M	Yes	counter	
dot3.2.1.10	dot3StatsInternalMacTransmitErrors	S	dot3 : M	Yes	counter	
dot3.2.1.11	dot3StatsCarrierSenseErrors	S	dot3 : M	Yes	counter	
dot3.2.1.13	dot3StatsFrameTooLongs	S	dot3 : M	Yes	counter	
dot3.2.1.16	dot3StatsInternalMacReceiveErrors	S	dot3 : M	Yes	counter	
dot3.2.1.17	dot3StatsEtherChipSet	S	dot3 : M	Yes	OID	
dot3.5	dot3CollTable	--	dot3 : O	Yes / No		
dot3.5.1	dot3CollEntry	--	dot3 : O	Yes / No		
dot3.5.1.2	dot3CollCount	S	dot3 : O	Yes / No	INT	
dot3.5.1.3	dot3CollFrequencies	S	dot3 : O	Yes / No	counter	
dot3.6	dot3Tests	--	dot3 : O	Yes / No		
dot3.6.1	dot3TestTdr	S	dot3 : O	Yes / No		
dot3.6.2	dot3TestLoopBack	S	dot3 : O	Yes / No		
dot3.7	dot3Errors	--	dot3 : O	Yes / No		
dot3.7.1	dot3ErrorInitError	S	dot3 : O	Yes / No		
dot3.7.2	dot3ErrorLoopbackError	S	dot3 : O	Yes / No		

© 2008 AASHTO / ITE / NEMA

A.25 TIME MANAGEMENT CONFORMANCE GROUP

The Time Management Conformance Group shall consist of the following objects:

DATABASE MANAGEMENT CONFORMANCE GROUP						
NTCIP 1201 Clause	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
2.4	**Time Management Conformance Group**	--	M	Yes	-----	----
2.4.1	globalTime	C	M	Yes	counter	
2.4.2	globalDaylightSaving	P	M	Yes	1–6	
2.4.5	globalLocalTimeDifferential	P	M	Yes	-43200 ..43200	deprecated
2.4.6	controllerStandardTimeZone	P	M	Yes	-43200 ..43200	
2.4.7	controllerLocalTime	S	M	Yes	counter	

© 2008 AASHTO / ITE / NEMA

< This page is intentionally left blank. >

Copy Per PRL Reproduction Notice
© 2008 AASHTO / ITE / NEMA

Annex B
COORDINATOR INFORMATION PROFILE REQUIREMENTS LIST

(Normative)

B.1 INTRODUCTION

This annex provides the Profile Requirements List (PRL) for implementations of the SCP related Coordinator Information in the form of proforma. A Profile Implementation Conformance Specification (PICS) for an implementation is generated by an implementer or supplier by indicating the appropriate level of support provided by an implementation.

To claim conformance with this profile, an implementation shall satisfy the mandatory conformance requirements of this profile.

An implementation's completed PRL is called the PICS. The PICS states which capabilities and options of the protocol have been implemented. The following can use the PICS:

a. The protocol implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight.

b. The supplier and user, as a detailed indication of the capabilities of the implementation.

c. The user, as a basis for initially checking the possibility of interworking with another implementation (note that, while interworking can never be guaranteed, failure to do so can often be predicted from incompatible PICSs).

d. A user, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

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 are found in the NTCIP Object Definition 1200-series standards. The NTCIP Object Definition standards 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. The device shall 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

© 2008 AASHTO / ITE / NEMA Copy Per PRL Reproduction Notice

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.

B.2 NOTATION

The following notations and symbols are used to indicate status and conditional status within NTCIP 1211 v01.

B.2.1 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.
P2	Parameter Object – use of 'dbCreateTransaction' in NTCIP 1201 Clause 2.3.1 to SET this object is mandatory.
S	Status / Information Object – this object is read only therefore a SET is not permitted.

B.2.2 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

 © 2008 AASHTO / ITE / NEMA

B.2.3 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.

B.2.4 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:

B.3 COORDINATOR CONFORMANCE GROUPS

Because Signal Control and Priority in the context of NTCIP 1211 is defined as providing priority while maintaining coordination, there is one change to table that appears in that clause. Reference A.8, Coordination Conformance Group, is Mandatory. In additional to the groups that are listed in table of A.2 of NTCIP 1202, two conformance groups, specifically related to Signal control and priority, also need to be referenced. The following table reflects the change to Clause A.2 of NTCIP 1202 v02.13 SCP Object Definitions and the additional groups needed for SCP.

Ref	Areas	Clause of Profile	Status	Support
A.8	Coordination Conformance Group	NTCIP 1202 – 2.5	M	Yes
B.4	Priority Strategy Conformance Group	NTCIP 1211 – 4.2	M	Yes
B.5	SCP Block Object Conformance Group	NTCIP 1211 – 4.3	M	Yes

Coordinator devices shall adhere to the conformance requirements specified in the as a minimum to claim compliance to NTCIP 1211 v01. Additional objects or groups may be supported without being non-compliant with Coordinator 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 Coordinator objects or NTCIP.

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

© 2008 AASHTO / ITE / NEMA Copy Per PRL Reproduction Notice

B.4 PRIORITY STRATEGY CONFORMANCE GROUP

The Priority Strategy Conformance Group shall consist of the following objects:

PRIORITY STRATEGY CONFORMANCE GROUP						
NTCIP 1211 Clause	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
4.2	**Priority Strategy Conformance Group**	---	M	Yes	---	---
4.2.1	maxStrategyRequestsToConsider	S	M	Yes	0–255	
4.2.2	priorityStrategyRequestedTable	--	M	Yes	----	---
4.2.2.1	priorityStrategyRequestedEntry	--	M	Yes	---	---
4.2.2.1.1	priorityStrategyRequestedNumber	S	M	Yes	1–10	
4.2.2.1.2	PriorityStrategyRequested	S	M	Yes	0–10	
4.2.2.1.3	priorityStrategyRequested TimeOfServiceDesired	S	M	Yes	0–4294967295	
4.2.2.1.4	priorityStrategyRequested TimeOfEstimatedDeparture	S	M	Yes	0–4294967295	
4.2.2.1.5	priorityStrategyRequestStatusInCO	S	M	Yes	1–15	
	idleNotValid	--	---	Yes	---	---
	readyQueued	--	---	Yes	---	---
	readyOverridden	--	---	Yes	---	---
	activeProcessing	--	---	Yes	---	---
	activeCancelpriorityStrategyRequest StatusInCO	--	---	Yes	---	---
	activeOverride	--	---	Yes	---	---
	activeNotOverriden	--	---	Yes	---	---
	closedCanceled	--	---	Yes	---	---
	reserviceError	--	---	Yes	---	---
	closedTimerToLiveError	--	---	Yes	---	---
	closedTimeError	--	---	Yes	---	---
	closedStrategyError	--	---	Yes	---	---
	closedCompleted	--	---	Yes	---	---
	activeAdjustNotNeeded	--	---	Yes	---	---
	closedFlash	--	---	Yes	---	---
4.2.3	coBusy	S	M	Yes	0 | 255	
4.2.4	priorityStrategiesMax	S	M	Yes	1–255	
4.2.5	priorityStrategyTable	--	M	Yes	----	---
4.2.5.1	priorityStrategyEntry	--	M	Yes	---	---
4.2.5.1.1	priorityStrategyNumber	S	M	Yes	1–255	
4.2.5.1.2	priorityStrategyServicePhases	P	M	Yes	string	
4.2.5.1.3	priorityStrategyPhaseOmits	P	M	Yes	string	
4.2.5.1.4	priorityStrategyPedOmits	P	M	Yes	string	
4.2.5.1.5	priorityStrategyDescription	P	M	Yes	string	
4.2.6.1	priorityStrategyExtensionToSplitTable	--	M	Yes	----	---
4.2.6.1	priorityStrategyExtensionToSplitEntry	--	M	Yes	---	---
4.2.6.1.1	priorityStrategyMaximumReductionTime	P	M	Yes	0–255	
4.2.6.1.2	priorityStrategyMaximumExtensionTime	P	M	Yes	0–255	
4.2.7	priorityStrategyDefaultCoordPattern	P	M	Yes	0–253	

Copy Per PRL Reproduction Notice
© 2008 AASHTO / ITE / NEMA

B.5 SCP BLOCK OBJECT CONFORMANCE GROUP

The SCP Block Object Conformance Group shall consist of the following objects:

BLOCK OBJECT CONFORMANCE GROUP						
NTCIP 1211 Clause	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
4.3	**SCP Block Object Conformance Group**	--	O	Yes / No	-----	----
4.3.1	scpBlockGetControl	C	2.12 : M	Yes	string	
4.3.2	scpBlockData	C	2.12 : M	Yes	string	
4.2.3	scpBlockErrorStatus	S	2.12 : M	Yes	0–65535	

NOTE — A device that supports this conformance group shall support the standard data block for each conformance group that is supported.

© 2008 AASHTO / ITE / NEMA

< This page is intentionally left blank. >

Copy Per PRL Reproduction Notice
© 2008 AASHTO / ITE / NEMA

Annex C
SCP DIALOGS

(Normative)

The following diagrams serve to further illustrate the message exchanges and their sequence between the various Processor entities involved in a signal control and prioritization system.

A priority request is initiated by a Priority Request Generator (PRG). The PRG could be a logical entity within a management center. However, for purposes of defining external communications, it is assumed to be part of a separate and distinct entity such as a transit vehicle. External communications are used to pass the priority request to a Priority Request Server (PRS) that sorts and prioritizes all requests coming from multiple sources. The PRS may be a logical unit within a Traffic Signal Controller. However, for purposes of defining external communications, it is assumed to be a separate and distinct physical entity that resides in a Traffic Control Assembly. A separate communications path is used to pass a service request to the Coordinator (CO). The CO then adjusts it coordination timings to provide the service.

C.1 CONTROL DIALOGS

C.1.1 Priority Request Server Control

There are six objects used for signal control and prioritization between the PRG and PRS. They are:

a. prgPriorityRequest
b. prgPriorityUpdate
c. prgPriorityStatusControl
d. prgPriorityStatusBuffer
e. prgPriorityCancel
f. prgPriorityClear

These are sent as SNMP messages with a single variable binding. The specific content of the single variable binding is specified in Sections 3.2.1 through 3.2.6.

The prgPriorityRequest object represents the control information to be passed from the PRG to the PRS for define a priority request. It is sent by the PRG to the PRS in the form of a SNMP SetRequest. The PRS responds in the form of a SNMP GetResponse that either confirms what was sent or indicates some type of error.

The prgPriorityUpdate object represents the control information to be passed from the PRG to the PRS for a priority request update of the time of service desired and estimated departure. It is sent by the PRG to the PRS in the form of a SNMP SetRequest. The PRS responds in the form of a SNMP GetResponse that either confirms what was sent or indicates some type of error.

The prgPriorityStatusControl object represents the control information to be passed from the PRG to the PRS to set up a read of status information. It is sent by the PRG to the PRS in the form of a SNMP SetRequest. The PRS responds in the form of a SNMP GetResponse that either confirms what was sent or indicates some type of error.

The prgPriorityStatusBuffer object represents the status information to be passed from the PRS to the PRG. It is retrieved by PRG sending a SNMP GetRequest to the PRS. The PRS responds in the form of a SNMP GetResponse that either returns the status or indicates some type of error. If the priorityRequestID, priorityRequestVehicleID, priorityRequestVehicleClassType, and

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

priorityRequestServiceStrategyNumber in the response does not match the desired information, the PRG should send a "priority status control" message.

The prgPriorityCancel object represents the control information to be passed from the PRG to the PRS to cancel a request. It is sent by the PRG to the PRS in the form of a SNMP SetRequest. The PRS responds in the form of a SNMP GetResponse that either confirms what was sent or indicates some type of error.

The prgPriorityClear object represents the control information to be passed from the PRG to the PRS to remove a priority request from the PRS. It is sent by the PRG to the PRS in the form of a SNMP SetRequest. The PRS responds in the form of a SNMP GetResponse that either confirms what was sent or indicates some type of error.

C.1.2 Coordinator Control

There is only one object used for signal control and prioritization between the PRS and CO. It is:

a. prsServiceRequest

This is sent as an SNMP message with a single variable binding. The specific content of the single variable binding is specified in Section 4.2.8.

The prsServiceRequest object represents the control information to be passed from the PRS to the CO to retrieve any requests for priority service. It is sent by the CO to the PRS in the form of a SNMP GetRequest. The PRS responds in the form of a SNMP GetResponse with the requests or indicates some type of error.

The prsServiceRequest object also represents the control information to be passed from the CO to the PRS to indicate the status of any request for priority service. It is sent by the CO to the PRS in the form of a SNMP SetRequest. The PRS responds in the form of a SNMP GetResponse that either confirms what was sent or indicates some type of error.

Do Not Copy Without Written Permission © 2008 AASHTO / ITE / NEMA

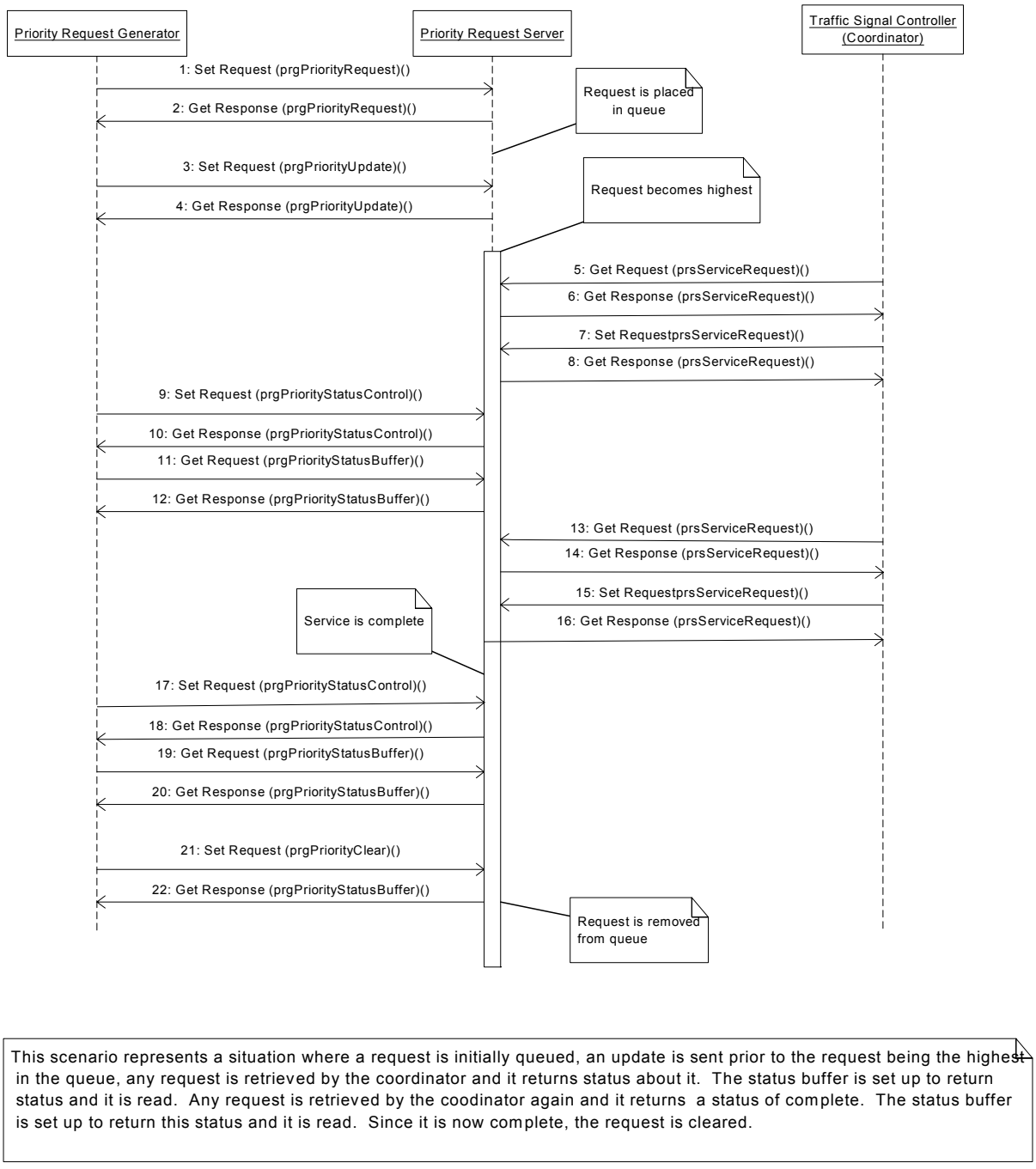


Figure C-1
Priority Request Sequence Diagram

C.2 CONFIGURATION, MONITORING, AND LOGGING DIALOGS

The configuration and monitoring of the PRS or CO is assumed to be controlled by TM. Retrieving any logged information from a PRS or CO is also assumed to be controlled by TM.

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

C.2.1 Priority Request Server Configuration, Monitoring, and Logging

There is one object used for configuring a PRS. It is:

a. prsProgramData

This is sent as an SNMP message with a single variable binding. The specific content of the single variable binding is specified in Section 3.2.7.

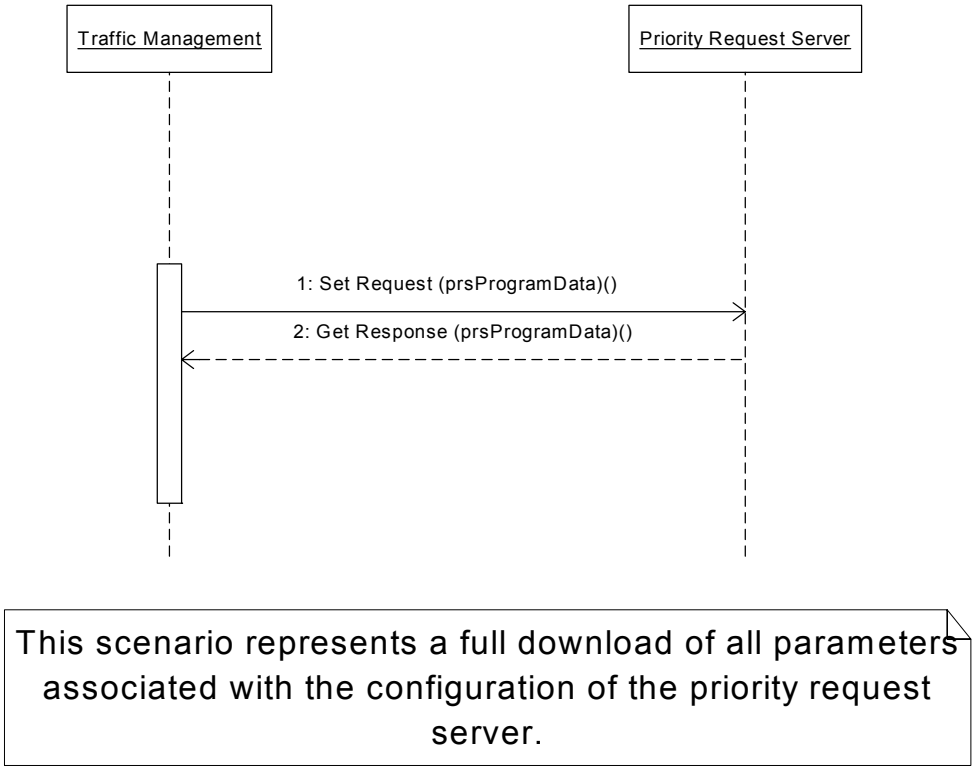


Figure C-2
Define Programming Sequence Diagram

The prsProgramData object represents the control information to be passed from the TM to the PRS to configure the PRS. It is sent by TM to the PRS in the form of a SNMP SetRequest. The PRS responds in the form of a SNMP GetResponse that either confirms what was sent or indicates some type of error.

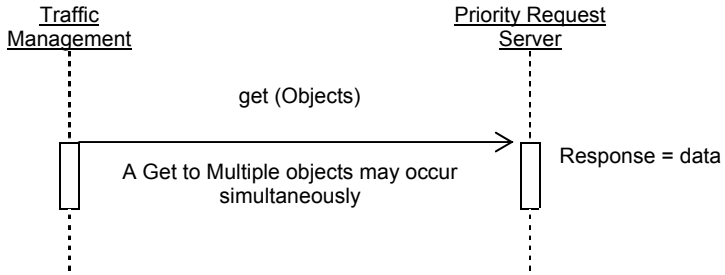
The prsProgramData object also represents the control information to be passed from TM to the PRS to retrieve the configuration of the PRS. It is sent by TM to the PRS in the form of a SNMP GetRequest. The PRS responds in the form of a SNMP GetResponse that either returns the configuration information or indicates some type of error.

There are no specific dialogs used for monitoring and logging. The generic SNMP operations on the various data types are defined in Sections C.2.1.1 through C.2.1.4.

 © 2008 AASHTO / ITE / NEMA

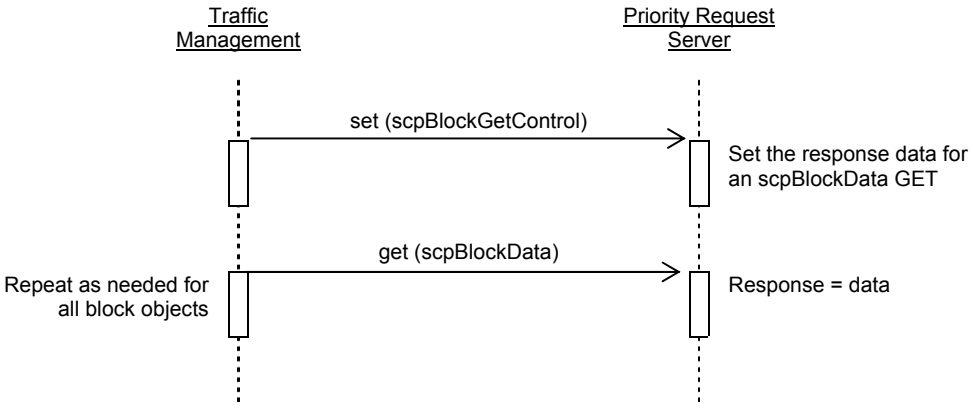
C.2.1.1 Get Type 'C' – 'P' – 'S' Objects

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



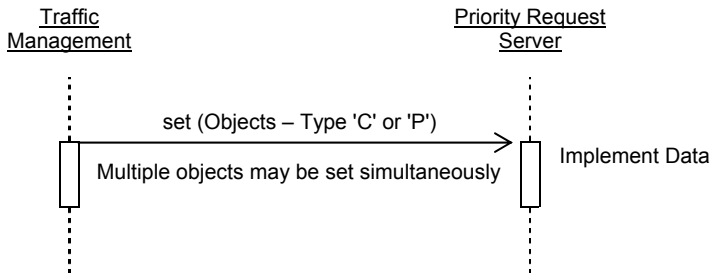
C.2.1.2 Get Block Data

This use case applies when getting block data via object 'scpBlockData'.



C.2.1.3 Set Type 'C' or 'P' Objects

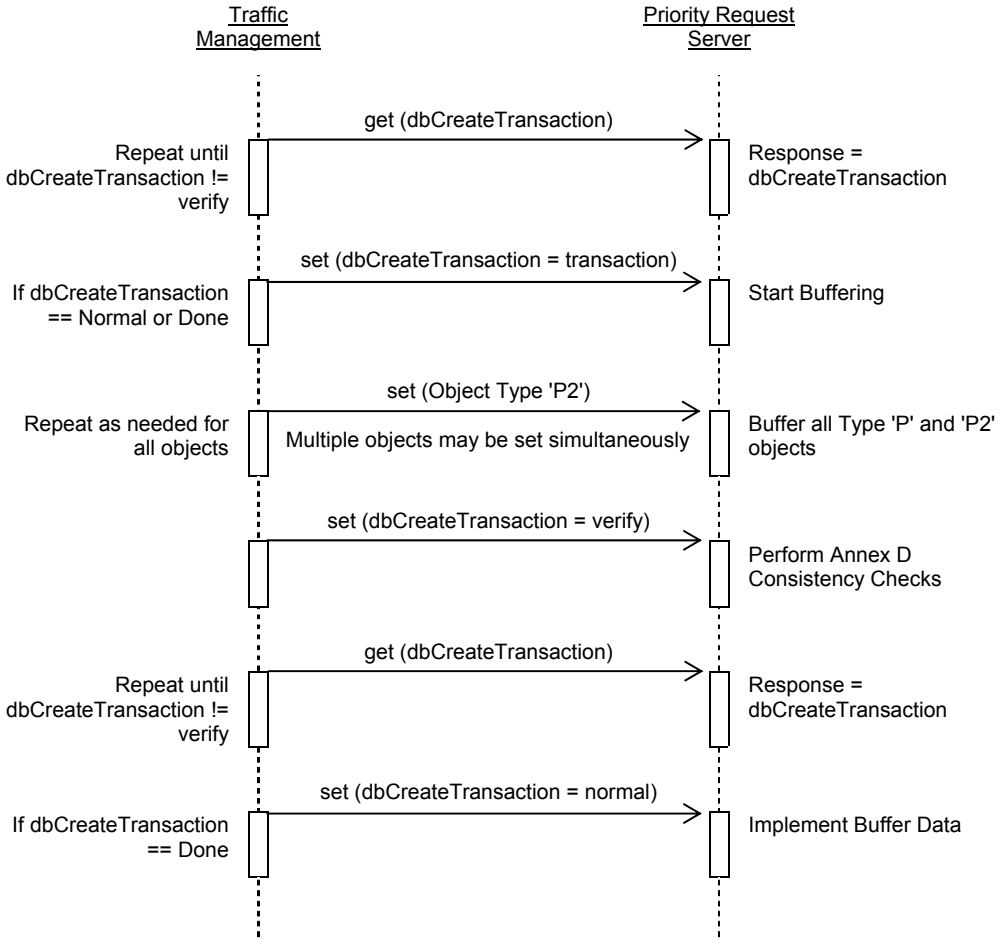
This use case applies when only Type 'C' or 'P' objects are included in the data to be set.



© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

C.2.1.4 Set Type 'P2' Objects

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



 © 2008 AASHTO / ITE / NEMA

C.2.2 Coordinator Configuration and Monitoring

There are two block objects used to configure the CO:

a. CoStrategyPlanBlockData
b. CoStrategySplitsBlockData

The content of these blocks are specified in Sections 4.4.1 and 4.4.2. These block objects are intended to be used with the block object control mechanism as defined in NTCIP 1202.

There are no specific dialogs used for monitoring and logging.

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

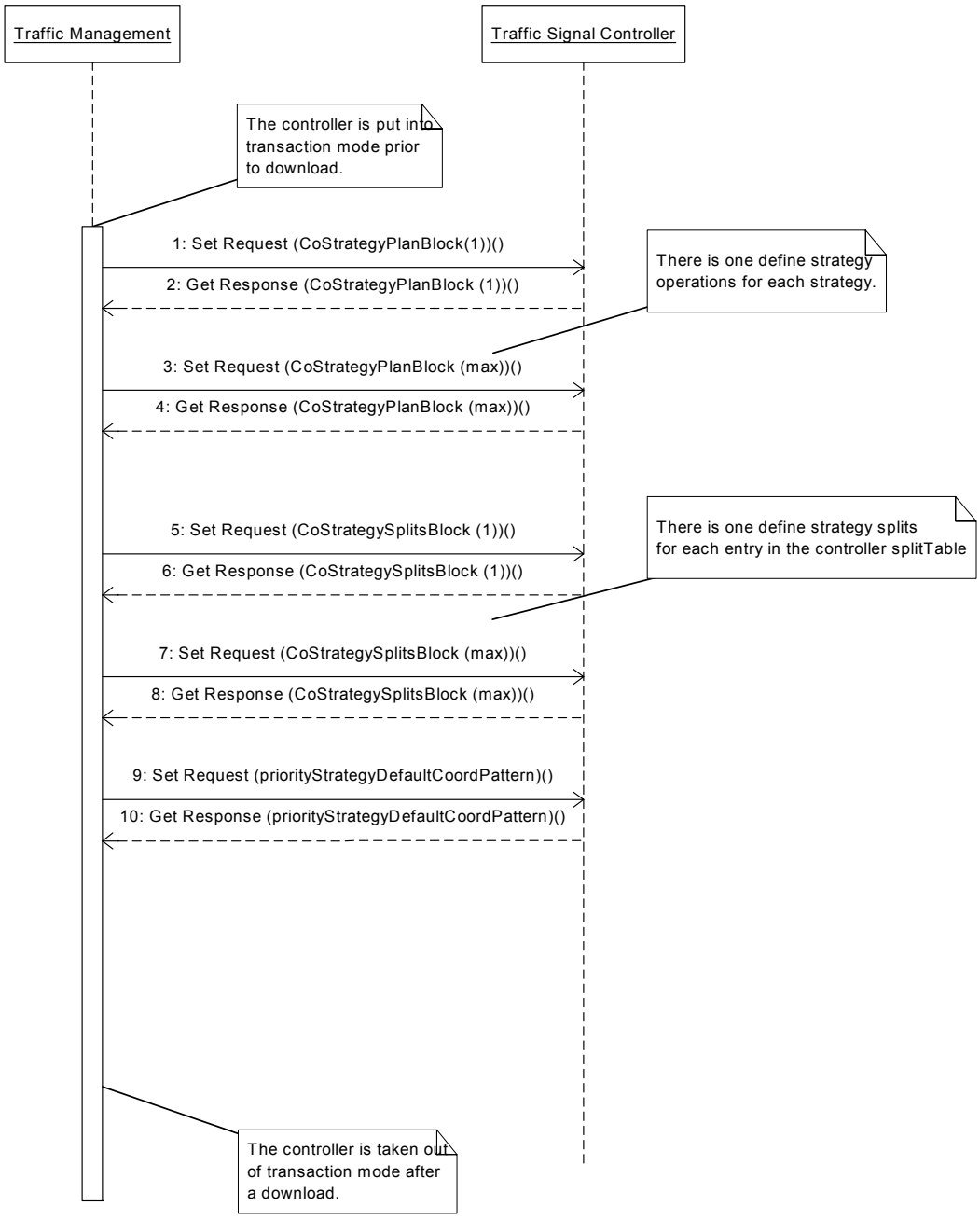


Figure C-3
Define Strategy Sequence Diagram

 © 2008 AASHTO / ITE / NEMA

Annex D
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 Coordinator when the controller is 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.

In addition to the consistency checks defined in NTCIP 1202, the following shall also be performed.

D.1 CONSISTENCY CHECK RULES

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

- The following objects define functionality related to phase assignments. Consistency checks insure that phases specified by these objects may operate concurrently and are defined only once in each string parameter. The value "xx" corresponds to priorityStrategyNumber.

  Priority Strategy Service Phases (priorityStrategyServicePhases)
  Priority Strategy Phase Omits (priorityStrategyPhaseOmits)
  SCP Split Coordinated Phases (splitCoordPhase)

  When more than one service phase is specified and the defined phases CANNOT time concurrently:
  "STRATEGY xx PHASE CG FAULT"

  When more than one service phase is specified and the defined phases are in the same ring:
  "STRATEGY xx PHASE RING FAULT"

  When a defined service phase are in the string parameter more than once:
  "STRATEGY xx PHASE MULTI FAULT"

  When a defined service phase is disabled:
  "STRATEGY PHASE DISABLE FAULT"

NOTE — The order of the checks is not defined. Therefore, for a given set of 'bad' data, the Error Message between different units may be inconsistent.

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission

< This page is intentionally left blank. >

Do Not Copy Without Written Permission © 2008 AASHTO / ITE / NEMA

Annex E
SCP TUTORIAL

(Informative)

In a Signal Control and Prioritization System, a Fleet Vehicle, Fleet Management, or Traffic Management initiates priority service by instructing the Priority Request Generator to send a Priority Request to a Priority Request Server. The Priority Request Server prioritizes and sorts all requests based upon Class Type and Class Level. It then sends its queue of requests to the Coordinator that resides in a Traffic Signal Controller. Based upon the strategy (or strategies based upon the implementation), the Coordinator will attempt to adjust the split timing in the Coordinator to accommodate the service request(s).

Based upon pre-programmed entries, the Coordinator applies phase and ped omits, and increases or decreases non-priority phase splits to extend the priority split time up to some maximum. The following figures illustrate the operation.

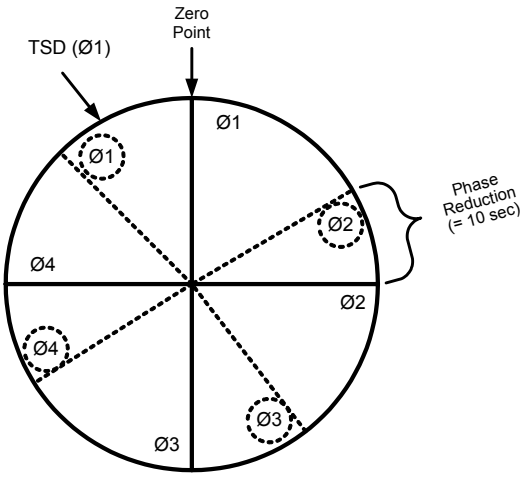


Figure E-1
Early Return to Coordinated Phase

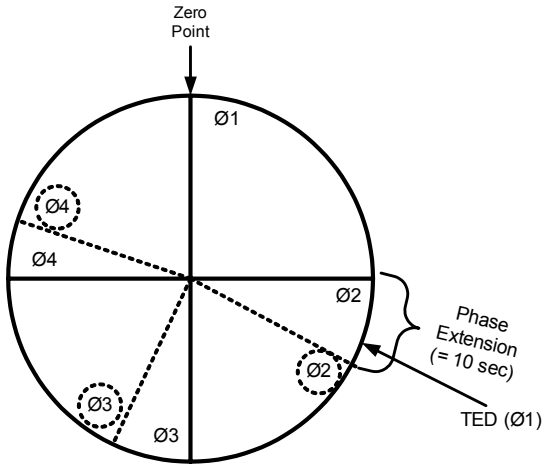


Figure E-2
Late Departure from Coordinated Phase

In the first figure, a priority request message for priority service on Phase 1 is received at some point prior to the "zero point." Projecting the priorityRequestTimeOfServiceDesired (TSD) into the anticipated timing sequence, the phase 1, 2, 3 and 4 split times are reduced by the priorityStrategyMaximumReductionTime to ensure an early return to Phase 1 green at TSD. [The split times for phases 2, 3, and 4 could have also been reduced to zero (0) by application of priorityStrategyPhaseOmits and/or priorityStrategyPedOmits.]

In the second figure, the priority request message's priorityRequestTimeOfEstimatedDeparture (TED) would normally occur after Phase 1 would have terminated. In this case, the Phase 1 split time is extended so that Phase 1 remains green so that the fleet vehicle can make its way through the intersection. [The split times for phases 2, 3, and 4 could have also been reduced to zero (0) by application of priorityStrategyPhaseOmits and/or priorityStrategyPedOmits.]

All coordinator calculations that project TSD and TED into a future timing position shall be based upon the current pattern. The calculations are not required to take into consideration a future change in pattern and any transitioning to that new pattern.

§

© 2008 AASHTO / ITE / NEMA Do Not Copy Without Written Permission