

A Joint Information Management Policy of AASHTO, ITE, and NEMA

NTCIP 8007 version v01

National Transportation Communications for ITS Protocol Testing and Conformity Assessment Documentation within NTCIP Standards Publications

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

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.

TRF Distribution Permission

To the extent that these materials are distributed by AASHTO / ITE / NEMA in the form of a Testing Requirements Form ("TRF"), AASHTO / ITE / NEMA extend the following permission:

- (i) you may make and/or distribute unlimited electronic or hard copies, including derivative works of the TRF, provided that each copy you make and/or distribute contains the citation "Based on NTCIP 0000 [insert the standard number] TRF. Used by permission. Original text © AASHTO / ITE / NEMA.";
- (ii) you may not modify the logical flow of any test procedure, without clearly noting and marking any such modification; and
- (iii) if the TRF excerpt is made from an unapproved draft, add to the citation "TRF excerpted from a draft standard containing preliminary information that is subject to change."

Content and Liability Disclaimer

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

AASHTO, ITE, and NEMA standards and guideline publications, of which the document contained herein is one, are developed through a voluntary consensus standards development process. This process brings together volunteers and/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.

< This page is intentionally left blank. >

ACKNOWLEDGEMENTS

NTCIP 8007 v01 was prepared by the Indefinite Quantities Contract (IQC) Project Team under a contract to the Federal Highway Administration (FHWA) and reflecting input provided by the NTCIP Testing and Conformity Assessment Working Group (TCA WG), a subdivision of the Joint Committee on the NTCIP. The Joint Committee is organized under a Memorandum of Understanding among the American Association of State Highway and Transportation Officials (AASHTO), the Institute of Transportation Engineers (ITE), and the National Electrical Manufacturers Association (NEMA). The Joint Committee on the NTCIP consists of six representatives from each of the standards organizations, and provides guidance for NTCIP development.

At the time that NTCIP 8007 v01 was prepared, the following individuals were active members of the NTCIP TCA WG:

- Steve Dellenback (Co-Chair)
- Joe Stapleton (Co-Chair)
- Alan Clelland
- Bob De Roche
- Ted Hancock
- Martin Knopp
- Jim Mahugh
- Gene Martin
- Jeff Morgan
- Bryan Mulligan
- Mel Partee
- Farhad Pooran
- Bob Rausch
- Lester Yoshida

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
- Battelle
- Caltrans
- Federal Highway Administration
- Florida Department of Transportation (DOT)
- Intelligent Devices, Inc.
- NET
- PB Farradyne, Inc.
- Peek Traffic Systems, Inc.
- Robert De Roche Consulting
- Quixote Corporation
- Siemens ITS
- SimpleSoft, Inc.
- Southwest Research Institute
- Texas DOT
- Texas Transportation Institute
- TKstds Management
- TransCore
- Trevilon Corporation
- URS Corporation
- Utah DOT
- Virginia DOT
- Washington State DOT

FOREWORD

NTCIP 8007 v01 defines the content requirements for the testing documentation for NTCIP standards.

NTCIP 8007 v01 uses only metric units, and NTCIP 8007 v01 includes three informative annexes.

This NTCIP policy is designated, and should be cited as NTCIP 8007 v01. NTCIP 8007 v01 is an adopted and published NTCIP Management Information Policy, reflecting the resolution of comments received during the development and approval process, when drafts were designated using a major and minor version number (v01.01 through v01.21). To distinguish NTCIP 8007 v01 from previous drafts, NTCIP 8007 v01 also includes NTCIP 8007 v01.21 on each page header.

For more information about NTCIP standards, visit the NTCIP Web site at <http://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 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

A User Comment should be submitted in the following form:

Standard Publication number and version:
Page:
Section, Paragraph, or Section:
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; April 2006
ITE—Software Standard; October 2005
NEMA—Standard; October 2005

History

In October 2002, the TCA WG began development of NTCIP 8007 v01. The policy was derived from a Federal Highway Administration report that documented how test procedures might be developed for a specific project. In January 2003, the Federal Highway report was distributed to the TCA WG for review and then became the basis for this policy.

NTCIP 8007 v01.09. August 2003—Accepted as a User Comment Draft by the Joint Committee on the NTCIP. October 2003—NTCIP Standards Bulletin B0084 distributed for user comment.

NTCIP 8007 v01.19. May 2004—Accepted as a User Comment Draft by the Joint Committee on the NTCIP. August 2004—NTCIP Standards Bulletin B0099 distributed for user comment.

NTCIP 8007 v01.20. July 2005—Accepted v01.20 as a Recommended Standard by the Joint Committee on the NTCIP. July 2005—NTCIP Standards Bulletin B0108 referred v01.20a for balloting. Approved by AASHTO in April 2006, approved by ITE in October 2005, and approved by NEMA in October 2005. Jointly Approved in 2006. May 2008—NTCIP 8007 v01.21 formatted and edited for publication.

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 NTCIP 8007 v01 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 NTCIP 8007 v01 should also consult NTCIP 8004 v01 for specific guidelines on compatibility.

INTRODUCTION

NTCIP 8007 v01 defines requirements that are applicable to all NTCIP environments, and it also contains optional and conditional sections that are applicable to specific environments for which they are intended.

The following keywords apply to this document: AASHTO, ITE, NEMA, NTCIP, testing.

In 1992, the NEMA 3-TS Transportation Management Systems and Associated Control Devices Section began the effort to develop the NTCIP. The Transportation Section's purpose was to respond 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.

In December 2001, the NTCIP Joint Committee agreed to include test documentation in the NTCIP standards and formed the TCA WG to further investigate conformance testing. In October 2002, the TCA WG held its first meeting and identified NTCIP 8007 v01 as a high priority work item.

CONTENTS

	Page
Section 1 INTRODUCTION	1
1.1 Purpose	1
1.2 Scope	1
1.3 Applicability.....	1
1.4 References	1
1.5 Contact Information	2
1.5.1 RFC Documents.....	2
1.6 Definitions.....	2
1.7 Abbreviations.....	3
Section 2 REQUIREMENTS FOR DEVELOPING TESTING REQUIREMENTS DOCUMENTATION [Informative]	4
2.1 Field Device Test Environment	4
2.2 Requirements to Test Cases Traceability	4
2.3 Traceability	4
2.3.1 Well-Written Requirements	5
2.3.2 Number of Test Cases to Develop.....	5
2.3.3 Tests for Referenced Requirements.....	5
2.3.4 Example 1	6
2.3.5 Example 2	6
2.3.6 Performance and Stress Testing	6
2.3.7 Presentation	6
2.4 Test Procedure	6
2.4.1 General	6
2.4.2 Required Elements.....	6
2.4.3 Presentation	8
Section 3 KEYWORDS	9
Annex A Sample Requirements [Informative]	15
A.1 Explore Data.....	15
A.2 Component Information.....	15
A.3 NTCIP Community Names	15
A.3.1 Administrator Community Name	15
A.3.2 Other Community Names	15
Annex B Sample Testing Requirements [Informative]	17
B.1 Field Device Test Environment	17
B.2 Requirements to Test Case Traceability	17
B.3 Test Procedures	18
B.3.1 Walk the MIB.....	18
B.3.2 Retrieve Module Table.....	19
B.3.3 Test Visibility of Security Data with Administrator Community Name.....	20
B.3.4 Change Time with Administrator Community Name.....	20
B.3.5 Change Administrator Community Name	21
B.3.6 Change User Community Name	22

< This page is intentionally left blank. >

Section 1 INTRODUCTION

1.1 PURPOSE

NTCIP standards are intended to promote the development of interoperable equipment within the Intelligent Transportation Systems (ITS) industry. Any component that does not fully follow the rules defined by the NTCIP standards will inhibit or preclude interoperability from being realized. As a result, industry needs a well-defined and consistent way to test equipment that claims conformance to the NTCIP standards.

NTCIP 8007 v01 is not intended to be used directly by manufacturers or public agencies that may procure NTCIP equipment; rather, it is intended to promote a consistent look and feel for testing documentation throughout various NTCIP standards. Thus, any user wishing to test an implementation of an NTCIP standard could use this subsequent documentation.

1.2 SCOPE

NTCIP 8007 v01 defines requirements to be used by NTCIP Working Groups in producing test documentation as a part of the NTCIP standards process.

1.3 APPLICABILITY

While NTCIP 8007 v01 has a similar scope to and is based on the principles defined in IEEE 829, NTCIP 8007 v01 has customized these principles for the NTCIP environment. Specifically, NTCIP 8007 v01 has refined the scope of IEEE 829 by:

- a) Defining a set of keywords, specific to the NTCIP environment, in order to ensure precise interpretations of what each term means.
- b) Allowing the definition of test variables in order to:
 - 1) Accommodate project-specific limitations (e.g., to define how many messages are supported by a sign or how many phases are supported by a signal controller).
 - 2) Allow for some level of variability in performing the tests so that a device cannot be hard-coded to a simple test.

1.4 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 document, and recommended revisions of the NTCIP Joint Committee, visit www.ntcip.org.

1.4.1 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) are referenced in these documents, which might provide a complete understanding of the entire protocol and the relations between all parts of the protocol. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standard listed.

IEEE Std 829-1998	<i>Standard for Software Test Documentation</i>
AASHTO / ITE / NEMA NTCIP 2301 v01	<i>Simple Transportation Management Framework Application Profile</i> , published in December 2001.

1.5 CONTACT INFORMATION

1.5.1 RFC Documents

Request for Comment (RFC) documents may be obtained from several repositories on the World Wide Web, or by “anonymous” File Transfer Protocol (FTP) with several hosts. Browse or FTP to:

www.rfc-editor.org
www.rfc-editor.org/repositories.html
 for FTP sites, read <ftp://ftp.isi.edu/in-notes/rfc-retrieval.txt>

1.6 DEFINITIONS

For the purposes of NTCIP 8007 v01, the following definitions for terms apply. Electrical and electronic terms not defined in this section are used in accordance with their definitions in IEEE Std 100-1996. For terms not defined in this section or in IEEE Std 100-1996, English words are used in accordance with their definitions in the latest edition of *Webster’s New Collegiate Dictionary*.

Conformance	<p>a condition that exists when an item meets all of the mandatory Requirements as defined by a formal standard.</p> <p>NOTE—Conformance may be measured on the standard as a whole, which means that it meets all mandatory (and applicable conditional) Requirements of the standard or on a feature level (i.e., it conforms to feature X as defined in section X.X.X), which means that it meets all mandatory (and applicable conditional) Requirements of the feature.</p>
Device Under Test (DUT)	NTCIP device that is the object of testing.
Negative Test	a test that verifies the DUT reacts properly to error conditions.
Risk	a subjective estimate of the probability of an error occurring and the amount of damage that may occur as a result of the error.

Test Case	a scenario that is designed to test a Requirement by specifying inputs and predicted results for a Device Under Test. NOTE—See IEEE 829 for a more detailed discussion of test cases.
Test Procedure	a step by step list of actions that establishes a repeatable method for conducting a test case.
Testing Requirements	the documentation that defines the test environment, the relationships between Requirements and Test Cases, and the Test Procedures.

1.7 ABBREVIATIONS

ASN.1	Abstract Syntax Notation One
IEEE	Institute of Electrical and Electronics Engineers Coordinator
NTCIP	National Transportation Communications for Intelligent Transportation Systems (ITS) Protocol
OID	Object Identifier
SNMP	Simple Network Management Protocol
TCA	Testing and Conformity Assessment

Section 2

REQUIREMENTS FOR DEVELOPING TESTING REQUIREMENTS DOCUMENTATION

[Informative]

Testing Requirements for a field device standard shall have the following major sections:

- a) Field Device Test Environment
- b) Requirements to Test Case Traceability
- c) Test Procedures

The major sections shall be ordered in the specified sequence. Details on the content for each section are defined in this section. Example Testing Requirements documents are provided in Annex B.

NOTE—This structure assumes that Testing Requirements are typically in one section (or annex) of a larger document. At a minimum, a general section (containing a scope, references and terms) would be needed as well.

2.1 FIELD DEVICE TEST ENVIRONMENT

The first section of the Testing Requirements shall define the test environment to be used for executing the Test Cases.

Some test procedures may require specialized test environments (e.g., if one wishes to test performance under high or low temperature extremes). The details for such environments shall be defined in one of the following ways:

- a) As an additional subsection defined in the Field Device Test Environment section (in which case, each Test Case shall indicate which test environment applies).
- b) Within the definition of each Test Case.
- c) As preconditions contained within the Test Procedure.

2.2 REQUIREMENTS TO TEST CASES TRACEABILITY

The second section of the Testing Requirements shall define the traceability among the Requirements of a standard and the associated Test Cases.

2.3 TRACEABILITY

Each Test Case shall derive from at least one Requirement and each Requirement shall trace to at least one Test Case.

NOTE—This allows a many-to-many relationship between Requirements and Test Cases.

NOTE—The same Test Case will likely be referenced by multiple Requirements, but this should be minimized to the extent possible in order to promote simplicity and to avoid ambiguity as to which Requirement has actually been tested and has passed or failed. Test Cases tend to become more complex when they are designed to fulfill multiple Requirements. Keep it simple; if a Requirement needs revision to simplify testing, the Working Group (WG) should consider making the revisions.

2.3.1 Well-Written Requirements

Each Requirement should:

- a) Use only well defined terms:
 - 1) Key terms should be defined in a glossary or other referenced source.
 - 2) Terms should be used in a consistent manner throughout (e.g., do not alternate between terms such as 'system operator' and 'operator').
- b) Not contain any conflicts with any other Requirements.
- c) Be a complete sentence:
 - 1) Each Requirement should have a subject.
 - 2) Each Requirement should indicate the required action.
 - 3) Each Requirement should identify the result of the action.
 - 4) Each Requirement should identify the object receiving the results of the action.
- d) Use a simple sentence structure to avoid ambiguity (e.g., complex and/or compound sentence structures increase the likelihood of multiple interpretations). Any complex or compound statements should be divided into simpler statements.
- e) Be written in the active voice.
- f) Define what is required and not how it is achieved.
- g) **Be testable.**

NOTE—A sample set of requirements are provided in Annex A.

2.3.2 Number of Test Cases to Develop

The number and extent of Test Cases developed for a given Requirement should reflect the risk assessment for the Requirement. The risk assessment should consider:

- a) The frequency of use of the feature to which the Requirement relates,
- b) The likelihood of finding an error given the other tests performed,
- c) The amount of damage that may occur as a result of the error,
- d) The cost of fixing or replacing the device after an error, and
- e) Many other factors dependent upon the application or device.

NOTE—The exact number of tests to be developed for a given Requirement is to be determined by the WG. The WG must balance the desire for testing to every possible scenario and the practical limitations of performing such tests. For example, a simple 8-character changeable message sign that supports a single 40 character font (e.g., 26 letters, 10 digits, and a handful of symbols) could display 6.55 trillion possible valid messages, and there could be many more invalid messages. Clearly, it is impractical to test every combination; on the other hand, testing only one may be insufficient. One way to minimize the problems of defining multiple tests is to identify values that may be variable in the test procedure.

NOTE—When high risk is involved, negative testing should also be considered (i.e., how does the device respond to an invalid message/request/dialog). Some examples of negative testing include boundary violation testing (e.g., attempting to set a parameter to a value above or below its valid range) and attempting to set parameters when their values should be protected by the current operational state of the device (e.g., a table may be designed to require a user to set the subject row to a 'modify' state prior to modifying any data). Negative testing may become unlimited in nature so it is important for the WG to weigh the risks against the number of tests developed.

2.3.3 Tests for Referenced Requirements

A device standard may reference Requirements, dialogs, interfaces, or objects from other standards (e.g., NTCIP 1201 Global Objects). The Test Case(s) for a given Requirement shall be contained in the Testing Requirements documentation for the standard containing the Requirement to be tested.

2.3.4 Example 1

The Requirement to "Set the Time" is a generic Requirement that may apply to a wide number of device types. As a result, it is most properly defined in a generic standard such as NTCIP 1201 (Global Objects). NTCIP 1201 would also define or unambiguously reference how the Requirement is to be fulfilled (i.e., the Design). Because NTCIP 1201 would define (either directly or by reference) the Requirement and the Design, the Test Case for this Requirement would be contained in NTCIP 1201.

2.3.5 Example 2

The Requirement to "Display the Time" is specific to dynamic message signs and would therefore be defined in NTCIP 1203 (DMS Objects). NTCIP 1203 would also define or unambiguously reference how the Requirement is to be fulfilled. While the process would likely reference objects in NTCIP 1201 (Global Objects), the testable Requirement (giving the context in which the data is used) is defined by NTCIP 1203. Thus, the Test Case for this Requirement would be contained in NTCIP 1203.

2.3.6 Performance and Stress Testing

When performance or stress testing is required, the test case shall include verification points where performance metrics (e.g., the response time to a request) are defined and sampled.

2.3.7 Presentation

Traceability shall be shown in a table consisting of the following columns:

- a) Requirement Identifier—The section number of the Requirement per the device standard.
- b) Requirement Title—The section title of the Requirement per the device standard.
- c) Test Case Identifier—An identifier, unique within the scope of the standard, for the Test Case. When referenced from outside of the subject standard, one should prefix the Test Case identifier with the identifier for the standard (e.g., NTCIP 1201 TC 001).
- d) Test Case Title—The test case title shall be a short name for the test case that is descriptive enough to identify the test case by name.

2.4 TEST PROCEDURE

The third section of the Testing Requirements shall define the Test Procedure for each Test Case.

2.4.1 General

The Test Procedure shall define the step-by-step sequence of actions that shall be performed in order to execute the associated Test Case, but shall not define the test tools employed to generate the actions on the device under test.

2.4.2 Required Elements

The definition for each Test Procedure shall include the following information:

- a) Test Case Identifier—Per Section 2.3.7.c.
- b) Test Case Title—Per Section 2.3.7.d.
- c) Test Case Description—The test case description shall describe the objective of the test and identify the key inputs and expected outputs.
- d) Variables—The variables section shall identify all variables that a tester must specify before performing the subject test case. All variable names should be unique within the document and only reused in different tests if referencing the same logical value (e.g., if one test case references another).
- e) Pass/Fail Criteria—The pass/fail criteria shall indicate the criteria to be used to determine if the device has passed or failed the test.

NOTE—Typically, the pass/fail criteria should state something like: "*The device under test (DUT) shall pass every verification step included within the Test Case in order to pass the Test Case.*"

- f) Test Steps—The procedure shall consist of a number of steps. Each step shall include the following information:
- 1) Test Step Number—The test step number shall be an identifier, unique within the scope of the test procedure, that defines the normal sequential order of execution of test procedure steps (i.e., the order of steps are sequentially numbered, unless changed by a branch statement such as "... then go to step N..."). The sequential order may include sub-identifiers as deemed appropriate (e.g., Step 1, Step 2, Step 2.1, Step 2.2, Step 3). The Test Step Number may be supplemented with a name in order to simplify references in branching statements. When present, Test Step Names shall be presented in parenthesis after the Test Step Number (e.g., "3 (LoopStart)").
 - 2) Test Step Action—The Test Step Action shall define the action that must be performed by the tester to carry out the Test Step.
 - i. The Action shall be tool independent.
 - ii. The Action shall not define the test operator interface mechanisms (e.g., use of a keyboard, mouse, front panel, etc.)
 - iii. The Action shall be a single action for the tester to:
 1. Do something;
 2. Wait (do nothing for a period);
 3. Observe something and verify that the observation is consistent with the required observation, in order to pass the Test. There shall be a maximum of one result per Test Step
 - iv. When referring to a value, the Action shall incorporate either
 1. The precise value to be used when performing a Test, or
 2. An indication that the value may be modified to:
 - a. Prevent a device from being manufactured to pass test with **ONLY** the pre-determined variable values.
 - b. Change values (e.g. Test Messages) that more accurately reflect the specific parameters of the Device Under Test (e.g. the size of the sign).
 - v. The Action shall be written as one or more English sentences. The definition of words not shown in boldface shall be per their definition (or referenced definition) in the Terms section of the subject standard. Words shown in boldface shall be considered to be keywords. The definition of keywords shall be per their definition in the Testing Keywords section, if any, of the subject standard. The definition of keywords that are not defined in the Testing Keywords section of the subject standard shall be per Section 3 of NTCIP 8007 (this standard).
 - 3) Test Step Notes—The Test Step may be supplemented with notes that may be helpful to the tester.
 - 4) Test Step Results—Each Test Step that has a verifiable outcome shall incorporate a Pass/Fail check box to record the results of the Test Step.
- g) Test Case Notes—The Test Case may be supplemented with notes that may be helpful to the tester.

2.4.3 Presentation

Each Test Case shall be presented in a format that allows a tester to record results directly on the form, although the use of the form by the tester is not required. In addition, the Testing Requirements documentation shall grant the proper permissions to allow the content of the Test Procedures to be reproduced both electronically and in paper form. An example format follows:

<i>Test Case:</i> <number>	<i>Title:</i>	<Test Case Title>		
	Description:	<Test Case Description>		
	Variables:	<Variables>		
	Pass/Fail Criteria:	<Pass/Fail Criteria>		
<i>Test Step Number</i> <number>	<i>Test Steps (Procedure)</i> <test steps (procedure)> NOTES— <notes>			<i>Results</i>
<number>	<procedure>			Pass/Fail
Test Case Results				
Tested By:		Date Tested		Pass/Fail
<i>Test Case Notes:</i>	<notes>			

Annex B provides an example using this format.

Section 3 KEYWORDS

NTCIP 8007 v01 Section 3 defines standardized keywords that may be used within Test Procedures. Keywords shall be shown in all capital letters with identical punctuation as shown in this Section.

A standard containing Test Procedures may define additional keywords for use within that standard.

3.1 COMMUNITY NAME IN: The value contained in the 'community' field of the last SNMP Message received from the DUT. See RFC 1157 for additional details related to the community name.

3.2 COMMUNITY NAME OUT: The value that the test application shall use for the 'community' field of the next SNMP Message sent to the DUT. See RFC 1157 for additional details related to the community name. Unless otherwise specified, this value shall be set to the administrator community name as stored in the DUT.

3.3 CONFIGURE: The CONFIGURE keyword shall be used as a predicate to the text of a test step in order to indicate that the text identifies a configurable variable that should be defined by the user prior to performing the test.

3.4 DELAY: The test application and user shall not perform any actions for a defined period of time, which may be measured in time units or by monitoring some event that does not involve any exchange of information over the communications media (e.g., DELAY until the temperature exceeds a threshold). In the later case, the step should also define exception conditions to allow for possibility that the event never happens.

3.5 DYNAMIC OBJECT NUMBER IN: The value contained in the 'Object ID' field of the last STMP Data Packet received from the DUT. See NTCIP 1103 for additional details related to the dynamic object number.

3.6 DYNAMIC OBJECT NUMBER OUT: The value that the test application shall use for the 'Object ID' field of the next STMP Data Packet sent to the DUT. See NTCIP 1103 for additional details related to the dynamic object number.

3.7 ERROR INDEX: The value contained in the 'error-index' field of the last SNMP RESPONSE received from the DUT. See RFC 1157 for additional details related to the error index.

3.8 EXIT: This keyword indicates that the user and test application should terminate the test case without performing any more steps. The keyword by itself does not have any implications as to whether a given test passes or fails.

3.9 GET: The test application shall transmit to the DUT one SNMP Message containing a GetRequest-PDU, per the rules of NTCIP 2301. Each statement using this keyword shall unambiguously reference the value for the 'name' field(s) to be included in the request. The GetRequest-PDU shall include all of the names in its 'variable-bindings' field. See RFC 1157 for additional details related to the GetRequest-PDU.

Unless otherwise indicated, the user or test application shall VERIFY the following, in order:

- a) The DUT responds with exactly one SNMP Message that contains a GetResponse-PDU, per the rules of NTCIP 2301; this is the RESPONSE. The DUT may also transmit one or more SNMP Messages that contain a Trap-PDU
- b) The value contained in the 'version' field of the RESPONSE equals 0 (version-1)
- c) COMMUNITY NAME IN equals COMMUNITY NAME OUT
- d) REQUEST ID IN equals REQUEST ID OUT
- e) RESPONSE ERROR equals 0 (noError)
- f) ERROR INDEX equals 0
- g) The 'variable-bindings' field contains the same number of VarBind structures as where contained in the GetRequest-PDU
- h) The value of each name field in the RESPONSE equals the value of the name field in the GetRequest-PDU that is in the same ordered position.

3.10 GET-NEXT: The test application shall transmit to the DUT one SNMP Message containing a GetNextRequest-PDU, per the rules of NTCIP 2301. Each statement using this keyword shall unambiguously reference the value for the 'name' field(s) to be included in the request. The GetNextRequest-PDU shall include all of the names in its 'variable-bindings' field. See RFC 1157 for additional details related to the GetNextRequest-PDU.

Unless otherwise indicated, the user or test application shall VERIFY the following, in order:

- a) The DUT responds with exactly one SNMP Message that contains a GetResponse-PDU, per the rules of NTCIP 2301; this is the RESPONSE. The DUT may also transmit one or more SNMP Messages that contain a Trap-PDU
- b) The value contained in the 'version' field of the RESPONSE equals 0 (version-1)
- c) COMMUNITY NAME IN equals COMMUNITY NAME OUT
- d) REQUEST ID IN equals REQUEST ID OUT
- e) RESPONSE ERROR equals 0 (noError)
- f) ERROR INDEX equals 0
- g) The 'variable-bindings' field contains the same number of VarBind structures as where contained in the GetNextRequest-PDU
- h) The value of each name field in the RESPONSE is greater than the value of the name field in the GetNextRequest-PDU that is in the same ordered position.

3.11 NEXT: A reference to the next sequential test step.

3.12 PERFORM: The user or test application shall perform another test case as a part of this test case. Unless otherwise indicated in the "PERFORM" statement, the user (and test application) shall use the variable values defined when the other test case is performed in a stand-alone fashion.

Example 1: In order to test the illumination features of a sign, it may be necessary to display a message on the sign; however displaying a message on the sign may be a separate requirement that is addressed by a separate test case. For the illumination test, it does not matter what text is displayed, any message will do. Thus, the call to the other test case may look something like:

PERFORM the "Display a Message" test case.

Example 2: In order to test the flashing capabilities of a sign, it may be necessary to display a message with a certain set of specific characteristics. Thus, the flashing test case might have a step to CONFIGURE the flashing_message variable and a separate step elsewhere in the procedure to:

PERFORM the "Display a Message" test case where message = flashing_message.

3.13 PRE-CONDITION: The PRE-CONDITION keyword shall be used as a predicate to the text of a test step in order to indicate that the text provides a textual description of any pre-conditions for the test case.

Pre-conditions are conditions that must be met prior to running a test case. Only one pre-condition shall exist in a test case and it shall always be the first step listed, if present.

3.14 POST-CONDITION: The POST-CONDITION keyword shall be used as a predicate to the text of a test step in order to indicate that the text provides a textual description of any post-conditions for the test case. Post-conditions are conditions that exist after the successful completion of a test case. Only one post-condition shall exist in a test case and it shall always be the last step listed, if present.

3.15 RECORD: The user (or test application) shall record the information indicated by the test step as a part of the test results. This information may be referenced by a later step of the test case (or by a later step of a calling step case).

3.16 REQUEST ID OUT: The value that the test application shall use for the 'request-id' field of the next SNMP Message sent to the DUT. See RFC 1157 for additional details related to the request id. Unless otherwise specified, this value shall start at an arbitrary value and shall increment by one for each SNMP Message sent by the test application.

3.17 REQUEST ID IN: The value contained in the 'request-id' field of the last SNMP Message received from the DUT. See RFC 1157 for additional details related to the request id.

3.18 RESPONSE: The last SNMP Message containing a GetResponse-PDU received from the DUT.

NOTE—SNMP uses the same message structure to respond to a GetRequest-PDU, a GetNextRequest-PDU, and a SetRequest-PDU. SNMP calls this message structure a 'GetResponse-PDU', even though it may be a response to a SetRequest-PDU.

3.19 RESPONSE ERROR: The value contained in the 'error-status' field of the last SNMP Message received from the DUT. See RFC 1157 for additional details related to the error status.

3.20 RESPONSE OID: The value contained in the indicated 'name' field of the last SNMP Message received from the DUT. Each statement using this keyword shall unambiguously reference which name field is to be considered, if the response is expected to contain multiple name fields.

3.21 RESPONSE VALUE: The value contained in the indicated 'value' field of the last SNMP Message received from the DUT. Each statement using this keyword shall unambiguously reference which value field is to be considered, if the response is expected to contain multiple value fields.

3.22 RESTART-POINT: The RESTART-POINT keyword shall be used as a predicate to the text of a test step in order to indicate that the step is a point in the procedure that the test can be restarted if the test had to stop for any reason (e.g., due to a failure in the DUT, a failure by the test application, a break taken by the user, etc.). The test step shall identify the actions and conditions necessary to restart the procedure at the given location. When normally performing the test, the RESTART-POINT step should be ignored.

3.23 SET: The test application shall transmit to the DUT one SNMP Message containing a SetRequest-PDU, per the rules of NTCIP 2301. Each statement using this keyword shall unambiguously reference the value for the 'name' field(s) to be included in the request. The statement shall also indicate the value of the 'value' field associated with each 'name' field. Unless otherwise indicated, the value will be encoded according to the SYNTAX of the associated object. The SetRequest-PDU shall include all of the names and values, with their indicated associations in its 'variable-bindings' field. See RFC 1157 for additional details related to the SetRequest-PDU.

Unless otherwise indicated, the user or test application shall VERIFY that:

- a) The DUT responds with exactly one SNMP Message that contains a GetResponse-PDU, per the rules of NTCIP 2301; this is the RESPONSE. The DUT may also respond with one or more SNMP Messages that contain a Trap-PDU
- b) The value contained in the 'version' field of the RESPONSE equals 0 (version-1)
- c) COMMUNITY NAME IN equals COMMUNITY NAME OUT
- d) REQUEST ID IN equals REQUEST ID OUT
- e) RESPONSE ERROR equals 0 (noError)
- f) ERROR INDEX equals 0
- g) The 'variable-bindings' field contains the same number of VarBind structures as where contained in the SetRequest-PDU
- h) The value of each name field in the RESPONSE equals the value of the name field in the SetRequest-PDU that is in the same ordered position.
- i) The value of each value field in the RESPONSE equals the value of the value field in the SetRequest-PDU that is in the same ordered position.

3.24 SET-UP: The SET-UP keyword shall be used as a predicate to the text of a test step in order to indicate that the test step is a preparatory step in order to set up an environment in which the actual test can take place. If the user and/or test application is unsuccessful in performing the test step, the user (and/or test application) shall EXIT the test case and the test case will neither pass nor fail. The user should then investigate the problem in performing the step and restart the test.

3.25 STMP-GET: The test application shall transmit to the DUT one STMP Data Packet containing a STMP-GetRequest-PDU, per the rules of NTCIP 2301. Each statement using this keyword shall unambiguously reference the dynamic object number to be included in the request. The STMP-GetRequest-PDU shall include the dynamic object number in the request. See NTCIP 1103 for additional details related to the STMP-GetRequest-PDU.

Unless otherwise indicated, the user or test application shall VERIFY the following, in order:

- a) The DUT responds with exactly one STMP Data Packet that contains a STMP-GetResponse-PDU, per the rules of NTCIP 2301; this is the RESPONSE. The DUT may also transmit one or more SNMP Messages that contain a Trap-PDU
- b) DYNAMIC OBJECT NUMBER IN equals DYNAMIC OBJECT NUMBER OUT
- c) The 'data' field can be properly parsed given the current dynamic object definition

3.26 STMP-SET: The test application shall transmit to the DUT one STMP Data Packet containing a STMP-SetRequest-PDU, per the rules of NTCIP 2301. Each statement using this keyword shall unambiguously reference the dynamic object number to be included in the request. The statement shall also indicate the value of each Referenced Object. Unless otherwise indicated, the value will be encoded according to the SYNTAX of the associated Referenced Object. The STMP-SetRequest-PDU shall include the dynamic object number and the values of the Referenced Objects. See NTCIP 1103 for additional details related to the STMP-SetRequest-PDU.

Unless otherwise indicated, the user or test application shall VERIFY that:

- a) The DUT responds with exactly one STMP Data Packet that contains a STMP-SetResponse-PDU, per the rules of NTCIP 2301; this is the RESPONSE. The DUT may also respond with one or more SNMP Messages that contain a Trap-PDU
- b) DYNAMIC OBJECT NUMBER IN equals DYNAMIC OBJECT NUMBER OUT
- c) The PDU Information field is empty.

3.27 VERIFY: The user or test application shall evaluate the expression that follows this keyword. Each statement using this keyword shall contain an unambiguous expression that will always evaluate to either true or false without subjective or qualitative judgments by the tester.

If the result is true:

- a) The verification step shall pass, and
- b) The test shall continue to the next step, unless otherwise indicated in the test case.

Otherwise, if the result is false:

- a) The verification step shall fail,
- b) The test case shall fail, and
- c) The test case shall EXIT, unless otherwise indicated in the test case.

NOTE—While criteria are often stated in exact terms (e.g., "The response shall be '3'"; or, "The sign shall display 'TEST'"; etc.), it may also be the case that criteria may be stated as ranges or thresholds (e.g., "The response shall be between '2' and '16' inclusive"; or, "The response shall be '3' or greater"; etc.). Each approach is valid and should be considered in the construction of a test case.

< This page is intentionally left blank. >

Annex A

SAMPLE REQUIREMENTS

[INFORMATIVE]

The sample Testing Requirement documentation provided in Annex B is based on the following sample Requirements. Actual Testing Requirements would reference Requirements that would be contained in the standard for which the procedures were prepared.

A.1 Explore Data

The device shall allow the management station to dynamically discover what data and data instances are supported by the device.

A.2 Component Information

The device shall allow a management station to determine identification information for each module contained in the device, including:

- a) An indication of the type of device
- b) The manufacturer of the module
- c) The model number or firmware reference of the module
- d) The version of the module
- e) An indication of whether it is a software or hardware module

A.3 NTCIP Community Names

The device shall vary the access to information stored in the device based on the community name provided with the request.

A.3.1 Administrator Community Name

The device shall recognize one name, and only one name, as the administrator community name. The following requirements apply to the administrator community name:

- a) The device shall allow the administrator to retrieve any data supported by the device, including community name information.
- b) The device shall allow the administrator to modify any modifiable data supported by the device, including community name information.
 - 1) The device shall allow the administrator to modify the community names recognized by the device.
 - 2) The device shall allow the administrator to modify the read-write privileges associated with each user community name, but under no circumstance shall a user community name have access to modify community name information and other sensitive data as defined by other standards.

A.3.2 Other Community Names

The device shall recognize any other defined community name as a user community name. The following requirements apply to user community names:

- a) The device shall support at least the number of community names as defined in the standard. If the standard does not specify the number to support, the device shall support at least one user community name.

- b) The device shall allow a user to retrieve any data supported by the device, except for community name information and other sensitive data as defined by other standards.
- c) The device shall allow a user to modify any modifiable data for which the user community name has been authorized to modify.

Annex B SAMPLE TESTING REQUIREMENTS [INFORMATIVE]

B.1 Field Device Test Environment

All Test Cases covered by this Testing Requirements documentation require the DUT to be connected to a test application as depicted in Figure B-1. A data analyzer may also be used to capture the data exchanged between the two components. The test environment should be designed to minimize any complicating factors that may result in anomalies unrelated to the specific test case. Failure to isolate such variables in the test environment may result in false results to the test. For example, the device may be conformant with the standard, but communication delays could result in timeouts and be misinterpreted as failures.

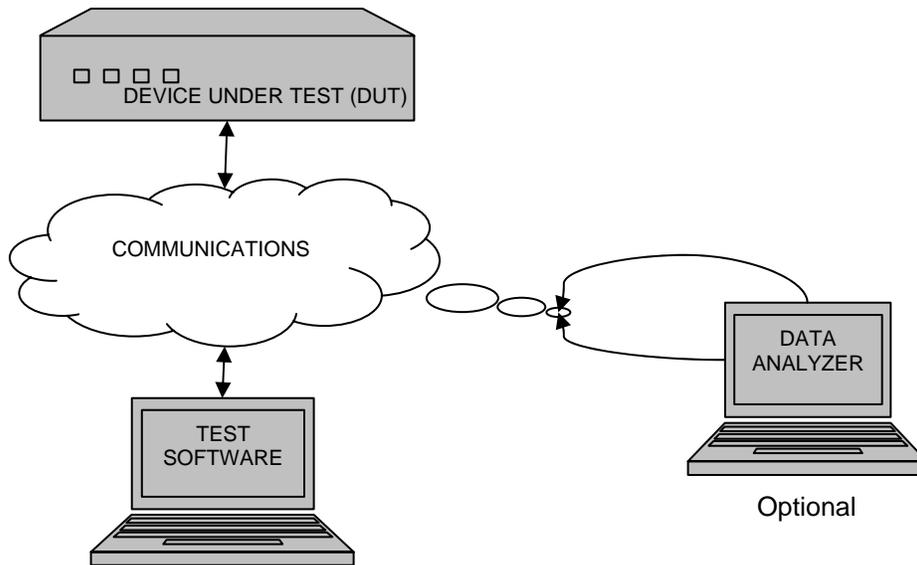


Figure B-1 Test Environment

B.2 Requirements to Test Case Traceability

Annex B.2 defines the formal relationships between the Requirements and the Test Cases.

Requirements to Test Case Matrix			
Requirement ID	Requirement Title	Test Case ID	Test Case Title
A.1	Explore Data	TC001	Walk the MIB
A.2	Component Information	TC002	Retrieve Module Table
A.3.1a)	Administrator Community Names— Retrieve Any Data	TC003	Test Visibility of Security Data with Administrator Community Name
A.3.1b)	Administrator Community Names— Modify Any Data	TC004	Change Time with Administrator Community Name
A.3.1b)1)	Administrator Community Names— Modify Community Names	TC005	Change Security Information

A.3.1b)2)	Administrator Community Names— Modify Community Name Access		
A.3.2.a	Other Community Names—Support a Number of Names	TC006	Test Visibility of Security Data with Administrator Community Name
A.3.2.b	Other Community Names—Retrieve any Data	TC007	Change User Community Names
•	•	•	•
•	•	•	•
•	•	•	•
X.X.X.X	Example Requirement X	TCXXX	Example Test Case X

B.3 Test Procedures

B.3.1 Walk the MIB

<i>Test Case:</i> TC001	<i>Title:</i>	Walk the MIB		
	<i>Description:</i>	This Test Case ensures that the data returned by the device during an exploratory walk is returned in a sequential order.		
	<i>Variables:</i>	start OID, stop OID		
	<i>Pass/Fail Criteria:</i>	The DUT shall pass every verification step included within the Test Case in order to pass the Test Case.		
<i>Test Step Number</i>	<i>Test Procedure</i>			<i>Results</i>
1	CONFIGURE: As a part of the project-specific Test Plan, define the OIDs used to start and stop the walk. NOTE—In order to walk the full database, the user could set the start OID to be zero (0) and the end point to be any value greater than 3 (i.e., due to the dot notations used, the value of 3 can never be exceeded). See RFC 1157 for more information.			
2	Perform a GET-NEXT using the start OID. If the RESPONSE ERROR equals 2 (noSuchName), pass this step and EXIT; otherwise go to NEXT.			Pass/Fail
3	VERIFY that the RESPONSE OID is greater than the start OID.			Pass/Fail
4 (StartLoop)	If the RESPONSE OID is greater than or equal to the stop OID, go to the end of the procedure.			
5	Perform a GET-NEXT using the RESPONSE OID. If the new RESPONSE ERROR equals 2 (noSuchName), pass this step and EXIT; otherwise go to NEXT.			Pass/Fail
6	VERIFY that the RESPONSE OID is greater than the previous Response OID.			Pass/Fail
7	Go to step StartLoop.			
Test Case Results				
<i>Tested By:</i>		<i>Date Tested</i>		Pass/Fail
<i>Test Case Notes:</i>				

B.3.2 Retrieve Module Table

<i>Test Case:</i> TC002	<i>Title:</i> Retrieve Module Table		
	<i>Description:</i>	This Test Case retrieves the module table, and allows the tester to verify that the DUT reports the proper type of device, manufacturer, model and version.	
	<i>Variables:</i>	OID for device type,	
	<i>Pass/Fail Criteria:</i>	The DUT shall pass every verification step included within the Test Case in order to pass the Test Case.	
<i>Test Step Number</i>	<i>Test Procedure</i>	<i>Results</i>	
1	GET globalMaxModules.0	Pass/Fail	
2	For each value of N, from 1 to the RESPONSE VALUE of step 1, perform steps 2.1 through 2.7.		
2.1	GET the following objects: 1. moduleNumber.N 2. moduleDeviceNode.N 3. moduleMake.N 4. moduleModel.N 5. moduleVersion.N 6. moduleType.N	Pass/Fail	
2.2	VERIFY that the RESPONSE VALUE for moduleNumber.N equals N.	Pass/Fail	
2.3	VERIFY that the RESPONSE VALUE for moduleDeviceNode.N equals the OID that has been designated for the type of device being tested. NOTE—For example, a signal controller should respond with an OID of 1.3.6.1.4.1.1206.4.2.1., per the rules defined by NTCIP 8004.	Pass/Fail	
2.4	VERIFY that the RESPONSE VALUE for moduleMake.N indicates the manufacturer of the module. NOTE—This might be the manufacturer of the hardware if the moduleType.N is hardware, or the developer of the software, if the moduleType.N is software.	Pass/Fail	
2.5	VERIFY that the RESPONSE VALUE for moduleModel.N indicates the correct model number of the component.	Pass/Fail	
2.6	VERIFY that the RESPONSE VALUE for moduleVersion.N indicates the correct version number for the component.	Pass/Fail	
2.7	VERIFY that the RESPONSE VALUE for moduleType.N indicates the correct type. NOTE—Values correspond to the following: other (1) hardware (2) software (3)	Pass/Fail	
Test Case Results			
<i>Tested By:</i>		<i>Date Tested</i>	<i>Pass/Fail</i>
<i>Test Case Notes:</i>			

B.3.3 Test Visibility of Security Data with Administrator Community Name

<i>Test Case:</i> TC003	<i>Title:</i> Test Visibility of Security Data with Administrator Community Name			
	<i>Description:</i>	This Test Case verifies that the security information is visible to the administrator.		
	<i>Variables:</i>	<none>		
	<i>Pass/Fail Criteria:</i>	The DUT shall pass every verification step included within the Test Case in order to pass the Test Case.		
<i>Test Step Number</i>	<i>Test Procedure</i>			<i>Results</i>
1	GET communityNameAdmin.0			Pass/Fail
2	VERIFY that the RESPONSE VALUE is equal to the COMMUNITY NAME OUT.			Pass/Fail
3	GET communityNamesMax.0			Pass/Fail
4	VERIFY that the RESPONSE VALUE is between 1 and 255, inclusive.			Pass/Fail
5	For each value of N, from 1 to the RESPONSE VALUE returned in Step 5, perform steps 6 through 8			
6	GET the following objects 1. communityNameIndex.N 2. communityNameUser.N 3. communityNameAccessMask.N			Pass/Fail
7	VERIFY that the RESPONSE VALUE for communityNameIndex.N is equal to N			Pass/Fail
8	VERIFY that the RESPONSE VALUE for communityNameUser.N is an OCTET STRING of more than or equal to 6 characters and less than or equal to 16 characters.			Pass/Fail
Test Case Results				
<i>Tested By:</i>		<i>Date Tested</i>		Pass/Fail
<i>Test Case Notes:</i>				

B.3.4 Change Time with Administrator Community Name

<i>Test Case:</i> TC004	<i>Title:</i> Change Time with Administrator Community Name			
	<i>Description:</i>	The Test Case verifies that the administrator can change the time (i.e. one sample object) in the DUT.		
	<i>Variables:</i>	<none>		
	<i>Pass/Fail Criteria:</i>	The DUT shall pass every verification step included within the Test Case in order to pass the Test Case.		
<i>Test Step Number</i>	<i>Test Procedure</i>			<i>Results</i>
1	GET globalTime.0			Pass/Fail
2	RECORD the RESPONSE VALUE as the "initial time"			
3	SET globalTime.0 to the initial time plus 3600. NOTE—Since the units of globalTime is in seconds, this has the effect of setting the clock one hour ahead.			Pass/Fail
4	GET globalTime.0			Pass/Fail

5	VERIFY that the RESPONSE VALUE is greater than or equal to the initial time plus 3600 and is less than the initial time plus 3660. NOTE—The upper limit is set with the assumption that Steps 3 and 4 require less than a minute to perform.	Pass/Fail
6	SET globalTime.0 to the current time.	Pass/Fail
Test Case Results		
Tested By:		Date Tested
		Pass/Fail
<i>Test Case Notes:</i>	It is not intended to check the accuracy of the clock, and it assumes that the DUT supports the <i>globalTime.0</i> object. At the end of this procedure, the clock may not be accurate, especially if the test fails.	

B.3.5 Change Administrator Community Name

<i>Test Case:</i> TC005	<i>Title:</i> Change Administrator Community Name	
	<i>Description:</i> This Test Case verifies that the administrator can change the administrator community name stored in the DUT and that the change properly affects the operation of the device.	
	<i>Variables:</i> Temporary community name	
	<i>Pass/Fail Criteria:</i> The DUT shall pass every verification step included within the Test Case in order to pass the Test Case.	
<i>Test Step Number</i>	<i>Test Procedure</i>	<i>Results</i>
1	CONFIGURE: As a part of the project-specific Test Plan, define the temporary community name to use for testing the administrator community name. This must be a different name than the current community name, which will be referred to as the original community name. NOTE—Valid administrator community names are between 8 and 16 characters long, inclusive and may contain any ASCII character, including non-printable characters.	
2	GET communityNameAdmin.0	Pass/Fail
3	VERIFY that the RESPONSE VALUE is equal to the COMMUNITY NAME OUT	Pass/Fail
4	SET communityNameAdmin.0 to the temporary community name.	Pass/Fail
5	GET communityNameAdmin.0 and Verify that either: 1. No response is received, or 2. The RESPONSE ERROR is 2 (noSuchName) and the RESPONSE INDEX is 1.	Pass/Fail
6	Change the COMMUNITY NAME OUT to the temporary community name.	
7	GET communityNameAdmin.0	Pass/Fail
8	VERIFY that the RESPONSE VALUE equals COMMUNITY NAME OUT	Pass/Fail
9	SET communityNameAdmin.0 to the original community name.	Pass/Fail
10	GET communityNameAdmin.0 and Verify that either:	Pass/Fail

	1. No response is received, or 2. The RESPONSE ERROR is 2 (noSuchName) and the RESPONSE INDEX is 1.	
11	Change the COMMUNITY NAME OUT to the original community name.	
12	GET communityNameAdmin.0	Pass/Fail
13	VERIFY that the RESPONSE VALUE equals COMMUNITY NAME OUT	Pass/Fail
Test Case Results		
Tested By:		Date Tested
		Pass/Fail
<i>Test Case Notes:</i>	If the test case fails, the administrator community name in the DUT may be the temporary community name.	

B.3.6 Change User Community Name

<i>Test Case:</i> TC006	<i>Title:</i> Change User Community Name <i>Description:</i> This Test Case verifies that the administrator can change the user community names stored in the DUT and that the change properly affects the operation of the device. <i>Variables:</i> Required number of community names <i>Pass/Fail:</i> The DUT shall pass every verification step included within the <i>Criteria:</i> Test Case in order to pass the Test Case.	
<i>Test Step Number</i>	<i>Test Procedure</i>	<i>Results</i>
1	PRE-CONDITION: Ensure that the DUT is configured such that the user community names are something other than "NTCIP USER #", where # is the row number in the community name table.	
2	GET globalMaxModules.0 NOTE—This object is referenced throughout this procedure. A Test Plan may replace all references to this object in this Test Case with a reference to any other object supported by the DUT that will have a constant value for the duration of the Test Case. The globalMaxModules is used in the standard since it is supported by just about every device.	Pass/Fail
3	RECORD the RESPONSE VALUE as the "number of modules"	
4	GET communityNamesMax.0	Pass/Fail
5	RECORD the RESPONSE VALUE as the "number of community names"	
6	VERIFY that the RESPONSE VALUE is greater than or equal to the number of community user names required by the Project Specification.	Pass/Fail
7	For each value of N, from 1 to the number of community names recorded in Step 5, perform BeginLoop1 through EndLoop1 NOTE—Steps BeginLoop1 through EndLoop1 test the original community names, to make sure that they work and then sets them to new values and ensures that the new values work	
8 (BeginLoop1)	GET the following objects: 1. communityNameUser.N 2. communityNameAccessMask.N	Pass/Fail

9	RECORD the RESPONSE VALUE for each object indicating which instance of N is being used.	
10	Change the COMMUNITY NAME OUT to the RESPONSE VALUE for communityNameUser.N	
11	GET globalMaxModules.0	Pass/Fail
12	VERIFY that the RESPONSE VALUE equals the number of modules as recorded in Step 3.	Pass/Fail
13	Change the COMMUNITY NAME OUT to the administrator community name.	
14	SET the following objects to the values as shown: 1. communityNameUser.N to the string "NTCIP USER #", where # is the row number N 2. communityNameAccessMask.N to the value of zero (0)	Pass/Fail
15	GET the following objects: 1. communityNameUser.N 2. communityNameAccessMask.N	Pass/Fail
16	VERIFY that the RESPONSE VALUE for communityNameUser.N is equal to "NTCIP USER #", where # is the row number N.	Pass/Fail
17	VERIFY that the RESPONSE VALUE for communityNameAccessMask.N is equal to zero (0).	Pass/Fail
18	Change the COMMUNITY NAME OUT to "NTCIP USER #", where # is the row number N.	
19	GET globalMaxModules.0	Pass/Fail
20	VERIFY that the RESPONSE VALUE equals the number of modules as recorded in Step 3.	Pass/Fail
21 (EndLoop1)	Change the COMMUNITY NAME OUT to the administrator community name. NOTE—End of first For Loop. See Step 7.	
22	For each value of N, from 1 to the number of community names recorded in Step 5, perform steps BeginLoop2 through EndLoop2 NOTE—Steps 23 to 24 test the original community names, to make sure that they no longer work	
23 (BeginLoop2)	Change the COMMUNITY NAME OUT to the communityNameUser.N value previously recorded in Step 9 for the given value of N.	
24 (EndLoop2)	GET globalMaxModules.0 and VERIFY that either: 1. No response is received, or 2. The RESPONSE ERROR is 2 (noSuchName) and the RESPONSE INDEX is 1. NOTE—End of second For Loop. See Step 22.	Pass/Fail
25	For each value of N, from 1 to the number of community names recorded in Step 5, perform steps BeginLoop3 through EndLoop3 NOTE—Steps 26 to 33 test the original community names once restored to the DUT, to make sure that they work	
26 (BeginLoop3)	Change the COMMUNITY NAME OUT to the administrator community name.	
27	SET the following objects to the values as shown: 1. communityNameUser.N to the value previously recorded in Step 8 for the same object and given value of N. 2. communityNameAccessMask.N to the value previously recorded in Step 8 for the same object and given value of N.	Pass/Fail

28	GET the following objects: 1. communityNameUser.N 2. communityNameAccessMask.N	Pass/Fail
29	VERIFY that the RESPONSE VALUE for communityNameUser.N is equal to the value previously recorded in Step 8 for the same object and given value of N.	Pass/Fail
30	VERIFY that the RESPONSE VALUE for communityNameAccessMask.N is equal to the value previously recorded in Step 8 for the same object and given value of N.	Pass/Fail
31	Change the COMMUNITY NAME OUT to the communityNameUser.N value previously recorded in Step 8 for the given value of N.	
32	GET globalMaxModules.0	Pass/Fail
33 (EndLoop3)	VERIFY that the RESPONSE VALUE equals the number of modules as recorded in Step 3.	Pass/Fail
34	NOTE—End of third For Loop. See Step 25. For each value of N, from 1 to the number of community names recorded in Step 5, perform steps BeginLoop4 through EndLoop4 NOTE—Steps 35 to 36 test the temporary community names created during this Test Case to make sure that they no longer work	
35 (BeginLoop4)	Change the COMMUNITY NAME OUT to "NTCIP USER #", where # is the row number N.	
36 (EndLoop4)	GET globalMaxModules.0 and VERIFY that either: 1. No response is received, or 2. The RESPONSE ERROR is 2 (noSuchName) and the RESPONSE INDEX is 1.	Pass/Fail
Test Case Results		
Tested By:		Date Tested
		Pass/Fail
<i>Test Case Notes:</i>	If the Test Case fails, one or more of the user community names in the DUT may be "NTCIP USER#", where # is the row number in the user community name table	

§