Technical Report Documentation Page

1. Report No.
FHWA/TX-06/0-5003-1

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
DEVELOPMENT OF TXDOT PROCEDURES AND SPECIFICATIONS FOR TESTING DEVICE COMPLIANCE TO NTCIP STANDARDS

5. Report Date
September 2005

6. Performing Organization Code

7. Author(s)
Robert De Roche

8. Performing Organization Report No.
Report 0-5003-1

9. Performing Organization Name and Address
Texas Transportation Institute
The Texas A&M University System
College Station, Texas 77843-3135

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
Project 0-5003

12. Sponsoring Agency Name and Address
Texas Department of Transportation
Research and Technology Implementation Office
P. O. Box 5080
Austin, Texas 78763-5080

13. Type of Report and Period Covered
Technical Report:
September 2004-August 2005

14. Sponsoring Agency Code

15. Supplementary Notes
Project performed in cooperation with the Texas Department of Transportation and the Federal Highway Administration.
Research Project Title: Testing for Compliance to NTCIP Standards
URL: http://tti.tamu.edu/documents/0-5003-1.pdf

16. Abstract

The primary objectives of this two-year project are to define a framework for testing conformance to National Transportation Communications for ITS Protocol (NTCIP) standards, identify the approaches used to describe the extent to which testing is needed, and recommend the appropriate documentation for such testing activities. To meet these objectives, the researcher looked at what other groups and organizations have done in support of testing. The researcher then describes the basic types of testing tools and provides descriptions and comparisons of applicable products. A survey of Texas Department of Transportation (TxDOT) division and district personnel was conducted to help understand the current TxDOT testing process and to identify any specific needs. Elements of a testing framework are then discussed. This discussion covers the basic steps involved in conformance testing, how NTCIP requirements are specified, aspects of management information base (MIB) files, current testing processes, reporting results, and mapping requirements to tests. Recommendations on defining the framework are given. The report concludes with an enumerated list of recommendation to establish a testing framework.

17. Key Words
NTCIP, ITS

18. Distribution Statement
No restrictions. This document is available to the public through NTIS:
National Technical Information Service
Springfield, Virginia 22161
http://www.ntis.gov

19. Security Classif.(of this report)
Unclassified

20. Security Classif.(of this page)
Unclassified

21. No. of Pages
96

22. Price

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

DEVELOPMENT OF TXDOT PROCEDURES AND SPECIFICATIONS FOR TESTING DEVICE COMPLIANCE TO NTCIP STANDARDS

by

Robert De Roche
Senior Research Specialist
Texas Transportation Institute

Report 0-5003-1
Project 0-5003
Project Title: Testing for Compliance to NTCIP Standards

Performed in cooperation with the
Texas Department of Transportation
and the
Federal Highway Administration

September 2005

TEXAS TRANSPORTATION INSTITUTE
The Texas A&M University System
College Station, Texas 77843-3135

DISCLAIMER

The contents of this report reflect the views of the author, who is responsible for the facts and the accuracy of the data, the opinions, and the conclusions presented herein. The contents do not necessarily reflect the official view or policies of the Federal Highway Administration (FHWA) or the Texas Department of Transportation (TxDOT). This report does not constitute a standard, specification, or regulation. This report is not intended for construction, bidding, or permit purposes. The researcher in charge was Robert De Roche.

The United States Government and the State of Texas do not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the object of this report.

ACKNOWLEDGMENTS

The research of this project was conducted under a cooperative program between the Texas Transportation Institute (TTI), the Texas Department of Transportation, and the Federal Highway Administration. The author wishes to thank the Project Monitoring Committee. Ms. Carol Rawson is the program coordinator, Mr. Fabian Kalapach is the project director, and Messrs. Charlie Farnham, David Danz, and Steve Barnettt served as the project advisors. Mr. Wade Odell also served as a member of the Project Monitoring Committee.

Special thanks go out to the people who participated in the interviews as part of this project: Messrs. Larry Wyble and Robert McKee of the Amarillo District, Mr. Brian Burk of the Austin District, Mr. Dexter Turner of the Corpus Christi District, Messrs. Dock Gee and Allen Hohle of the Houston District, and Messrs. Fabian Kalapach and David Danz at Traffic Operations Division.

Within TTI, the author wishes to thank Dr. Kevin Balke for reviewing and commenting on the report and Mr. Jeremy Johnson for his invaluable help in developing scripts. A debt of gratitude also goes to Mr. G. Curtis Herrick (deceased) who helped get this project off the ground.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

LIST OF ABBREVIATIONS AND SYMBOLS

AASHTO	American Association of State Highway and Transportation Officials
ASC	Actuated Signal Controller
ATMS	Advanced Transportation Management System
CCTV	Closed Circuit Television
CHAP	Challenge Handshake Authentication Protocol
CHART/CHART II	Coordinated Highways Action Response Team
CID	Controller Interface Device
CLS	Closed Loop System
CORSIM	Corridor Simulator
DMS	Dynamic Message Sign
DOS	Disk Operating System
DOT	Department of Transportation
DUT	Device Under Test
ESS	Environmental Sensor Station
FDOT	Florida Department of Transportation
FHWA	Federal Highway Administration
FMS	Field Management Station
GPS	Global Positioning System
GUI	Graphical User Interface
ICS	Implementation Conformance Specification
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ITE	Institute of Transportation Engineers
ITL	Interoperability Test Lab
ITS	Intelligent Transportation Systems
LAN	Local Area Network
MIB	Management Information Base
MUTCD	Manual on Uniform Traffic Control Devices
NEMA	National Electrical Manufacturers Association

NIATT	National Institute for Advanced Transportation Technology
NTCIP	National Transportation Communications for ITS Protocol
NTCIP 1201-GLO	NTCIP 1201 – Global Object Definitions
NTCIP 1202-ASC	NTCIP 1202 – Object Definitions for Actuated Signal Controller Units
NTCIP 1210-FMS	NTCIP 1210 – Field Management Stations - Part 1: Object Definitions for Signal System Masters
NTCIP 8007-TEST	NTCIP 8007 – Testing and Conformity Assessment Documentation within NTCIP
OID	Object Identifier
PC	Personal Computer
PDF	Portable Document Format
PMPP	Point-to-Multi-Point Protocol
PPP	Point-to-Point Protocol
PRL	Protocol (or Profile) Requirements List
PTZ	Pan, Tilt, and Zoom
QPL	Qualified Products List
SFMP	Simple Fixed Management Protocol
SNMP	Simple Network Management Protocol
STMP	Simple Transportation Management Protocol
TCA	Testing and Conformity Assessment
TCIP	Transit Communications Interface Profiles
Tcl	Tool Command Language
TCP	Transmission Control Protocol
TERL	Traffic Engineering and Research Lab
TNT	TxDOT NTCIP Tester
TRF	Traffic Operations Division
TS	Traffic Section
TTI	Texas Transportation Institute
TxDOT	Texas Department of Transportation
UDP	User Datagram Protocol
VIVDS	Video Imaging Vehicle Detection Systems
WAN	Wide Area Network

WG	Working Group
XML	Extensible Markup Language

GLOSSARY

Compliance	A condition that exists when an item meets all of the requirements of a procurement specification.
Conformance	A condition that exists when an item meets all of the mandatory requirements as defined by a formal standard.
Management Application	A generic term for any computer-based software used to configure, control, or monitor the operation of a field device or devices.

CHAPTER 1:
INTRODUCTION

PROJECT OBJECTIVES

The objectives of this two-year project are to define a framework for testing conformance to National Transportation Communications for Information Technology Systems Protocol (NTCIP) standards, identify the approaches used to describe the extent to which testing is needed, and recommend appropriate documentation for such testing activities. This research project will accomplish the following for TxDOT:

- Assist TxDOT in developing a comprehensive approach to testing Intelligent Transportation Systems (ITS)-related hardware and software to ensure conformance with national standards and compliance with TxDOT specifications.
- Identify TxDOT testing needs and available resources to meet those needs.
- Develop a framework, along with methodologies and procedures as needed, for conducting both laboratory and field-testing of devices.
- Assist TxDOT in evaluating options for testing of ITS hardware and software as part of procurement and construction projects.
- Assist TxDOT in developing procedures and reports for documenting the results of the testing program.
- Develop outlines for training courses that convey how to use and interpret the results of the testing program.

The goal of testing is to ensure that the design, implementation, and functionality of a product meet user needs and requirements. NTCIP standards define a set of protocols associated with communications technologies used in transportation-related products. These protocols ensure systems that integrate NTCIP-conformant products can communicate using a common language and describe information in a consistent manner. The goal of NTCIP testing is to ensure that a product follows the protocol rules that define the common language and that the information exchanged by the components of a system is meaningful and understood. The intent of this project is to look at tasks involved and methods used to check conformance to NTCIP.

The intention is to look also at a means of integrating NTCIP testing into TxDOT's current testing program.

SCOPE OF PROJECT

The scope of this research is testing of conformance to NTCIP standards and compliance to TxDOT specifications that reference NTCIP standards. NTCIP standards define common methods and protocols that enable ITS devices to communicate. The standards also define the language and words used when communicating. In some cases, no other standards exist that define the meaning of the words or how the words relate to functionality; therefore, some NTCIP standards also define the functionality of an ITS device. This research looks at the types of testing involved in showing conformance to NTCIP standards and compliance to TxDOT specifications, the resources available to accomplish the tasks, the specific needs for testing by TxDOT, the current testing process within TxDOT, and the testing tools available to help in the testing process.

ORGANIZATION OF THIS REPORT

In addition to this introduction chapter, this report contains seven chapters providing details on the project's findings. Chapter 2 provides background material on the goals of testing and the types of testing involved in a testing program. Chapter 3 presents a synopsis of industry resources that may be applicable to the testing process. Chapter 4 provides a summary and comparison of what tools are available for testing. Chapter 5 summarizes the responses to a questionnaire on testing. Chapter 6 looks at how TxDOT specifications specify NTCIP, aspects of TxDOT requirements that are not addressed by the NTCIP standards, and the testing process within TxDOT. Finally, Chapter 7 provides a summary of the recommendations on defining a framework for conformance testing.

CHAPTER 2:
BACKGROUND

OVERVIEW

As national standards become available for NTCIP and other ITS standard-based devices and services, coupled with the agency's ongoing need to expand its ITS infrastructure, TxDOT will need an approach to ensure that comprehensive testing is conducted in an efficient manner that does not sacrifice quality.

As background, this research identifies the goals that can be realized using ITS standards. We explain the terminology that shows how implementations relate to project specifications and standards. In addition, we discuss the concept of testing, along with the various types of testing commonly used by agencies for testing compliance to NTCIP standards. Finally, we explore some of the issues related to communications testing versus functional testing.

GOALS OF ITS STANDARDS AND TESTING

Intelligent Transportation Systems use computer, electronics, and communications technologies to manage and operate surface transportation systems as safely and efficiently as possible. These systems integrate many different advanced technologies, and therefore standards are crucial. With ITS standards, transportation agencies can add components to their systems as needed without creating incompatibilities or parallel communications infrastructures. The standards facilitate information sharing and accommodate equipment replacements, system upgrades, and system expansion. The traveling public will also benefit from ITS standards in that products that use them will function consistently throughout the country and be more cost-effective. ITS standards facilitate national and regional interoperability and lead to competitive markets for transportation services and products (*1*).

NTCIP is a subset of the ITS standards that provides both rules for communicating (protocols) and the vocabulary (object definitions) necessary to allow similar devices from different manufacturers to operate with each other as a system. The rules and vocabulary also allow dissimilar devices from different manufacturers to operate within a system. The goal of the standards development effort is to promote compatibility, interoperability, and interchangeability in support of the deployment of regional and national ITS architectures and systems.

The goal of testing is to ensure that equipment meets an agency's requirements and expectations (which now incorporate NTCIP). This is different from what was previously required. Traditionally, an agency did not specify a communications protocol other than the proprietary protocol defined by a vendor or one already in use by the agency. A means did not exist to check an agency's functional requirements except manually or with the vendor's software. By including NTCIP in the specifications, an agency must still determine if a device meets the agency's functional requirements and the identified NTCIP standards, but the agency can use NTCIP to do both. If an agency can communicate with a device using NTCIP protocols, that process can show that the device is conformant to the NTCIP protocol standards. If an agency uses the objects in NTCIP to control and monitor the functions within a device, that process can show that the device is conformant to NTCIP object definition standards. That same process can also show that a device is compliant with an agency's functional specifications.

TYPES OF TESTING

There are five general classes or categories of testing: physical, environmental, communications, functional, and performance. Physical testing focuses on the mechanical characteristics of a device. Environmental testing focuses on the external conditions and surroundings in which an implementation will operate. Communications testing focuses on the ability to exchange information between components of a system. Functional testing deals with how an implementation uses or reacts to changes in control information that it receives. Performance testing focuses on the ability of an implementation to accept and respond to a set of commands in a timely manner.

Physical

In physical testing, the tester is trying to determine whether a device meets the material, dimensional, and other mechanical characteristics of the specified requirements. ITS devices reside in an industrial environment. They are subject to rough handling and size restrictions and must interconnect with other components. Testing of the requirements related to the physical characteristics of a device is usually performed by visual inspection and follows a checklist of items to inspect. To gauge how well a device survives rough handling, a "drop test" determines if a device will survive sudden impacts.

Environmental

With environmental testing, the tester determines whether devices can function in the environment in which they must operate. Most ITS devices must endure a demanding outdoor environment. Exposure to temperature and humidity extremes can be very stressful to electronic components. Typically placed close to a roadway, they also must endure shock and vibration that comes with that location. Electrical disturbances and variations in the source of power put special demands on implementations. Environmental testing typically consists of operating a device in a temperature chamber as temperature and voltage vary across extremes. A vibration test determines if a device can withstand the constant shaking associated with roadside or roadway locations.

Communications

The purpose of communications testing is to determine whether information can be sent to and received from a device or system. The primary focus of NTCIP is the definition of common protocols for exchanging information and the vocabulary for expressing information. Protocols are the rules for establishing a connection between components and passing information through the connection. Data dictionaries define the vocabulary or the words to describe information and their meaning. A good analogy to explain the difference is sending a letter. The protocol for exchanging a letter is to place it in an envelope, add the address of the person to receive it, put on a stamp, and then put it in a mailbox. Following the rules of this protocol allows us to exchange information with just about anyone. It does not ensure that the person who receives it understands the contents of the letter. For that to happen, the sender and recipient need to use a common dictionary. If the sender writes the letter in English and the recipient only understands Spanish, the contents of the letter are meaningless.

The first part of communications testing checks whether an implementation follows the protocol rules. Testing procedures check the equivalent of whether the recipient’s address is correct and whether the postage is adequate. The second part of communications testing checks whether the correct data dictionary is used. This testing is equivalent to checking whether the words are spelled correctly.

Initially, NTCIP standards were considered strictly communications standards and functionality was defined elsewhere. The first object definition standard was for traffic signal controllers, and the general understanding was that functionality was defined in either the National Electrical Manufacturers Association (NEMA) Traffic Section (TS) 1 – Traffic Control Systems or TS 2 Standard – Traffic Controller Assemblies with NTCIP Requirements (*2*, *3*). As the standard evolved, however, the description of the objects took on more of a functional definition. For example, both NEMA TS 2 and NTCIP 1202 – Object Definitions for Actuated Signal Controller Units (NTCIP 1202-ASC) use the following words to describe *Maximum Green* (*4*):

> *This time setting shall determine the maximum length of time this phase may be held Green in the presence of a serviceable conflicting call. In the absence of a serviceable conflicting call the Maximum Green timer shall be held reset...*

The intention of using the same definition for a common term in both standards was to make sure that an ambiguity did not exist between them. That definition, however, is a functional requirement of how it operates. The unintended result was that many consider NTCIP standards as de facto functional standards, as well.

Some NTCIP standards could not capitalize on the existence of a functional standard. At the time work started on the dynamic message sign (DMS) standard, the functionality of message signs was not covered by an existing standard. The NEMA TS 4-2005 – Hardware Standards for Dynamic Message Signs with NTCIP Requirements was released on April 11, 2005 (*5*). The Field Management Station (FMS) Standard falls under the same category. The researcher learned from the NEMA Section Manager that the work effort on a NEMA TS 5 functional standard for FMS should begin in the fall of 2005 (Vicki Schofield, May 19, 2005). The NTCIP FMS Working Group (WG) is in the process of finalizing their standard. Since a number of devices do not have a functional standard, the appropriate NTCIP is also the functional standard.

If one argues that the NTCIP standards only cover communications, then conformance to NTCIP means that an implementation has to support the appropriate protocols to exchange information and define information according to the appropriate data dictionary. If, on the other hand, one argues that the NTCIP standards also cover functionality, then conformance to NTCIP also means that an implementation has to perform all of its functions in a specific way. For any given configuration and set of inputs, an output must function in a prescribed manner.

Functional

Functional testing is the most difficult and time-consuming part of any test suite. In functional testing, the tester examines whether a device uses information properly or indicates the proper status information. Even though communications protocols can be used to check the equivalent of proper spelling, the actual meaning or semantics of information is application specific. For example, consider the word “wind.” In one context, it means the movement of air. In another context, it means to wrap or encircle. The meaning and behavior associated with a word may be very different depending on the context. In NTCIP, behavior translates to an operation or a function.

For example, in NTCIP, a device determines local time by adding a time zone offset to global time (Greenwich Mean Time) and then adjusting for daylight savings time. A communications test procedure could test whether global time, time zone, and daylight savings time can be set and whether the current value can be read back. This procedure does not check any behavior or functionality associated with the three parameters. A functional test procedure would set the three variables to specific values and then read the value of local time to ensure that it is correct. The functional test would also check to see whether the hour advances or falls back on the daylight savings time dates. Another example in the context of a traffic signal controller is the use of NTCIP protocols to set the parameters associated with a coordination plan scheduler. Communications testing could confirm that the appropriate date, time, and plan number can be set. A functional test would confirm that at an established time, the coordination plan number goes into effect.

Performance

In performance testing, the tester determines whether a device can accept and respond to a given number of messages in a certain established time frame. Some ITS field devices do not have stringent performance requirements. For example, the speed at which a DMS changes a message or a closed circuit television (CCTV) pans 90 degrees is usually measured in seconds. Other ITS devices, such as traffic signal controllers that are part of a closed-loop system or directly controlled from a central location, may impose performance requirements.

In the past, some proprietary systems could respond to 12-24 messages per second using 1200 bits per second communications links. Because the open architecture approach of NTCIP incurs more overhead and processing messages takes longer, NTCIP is not as efficient as some proprietary systems. As such, it would be useful to test whether a device meets message timing and processing requirements. While a particular system's requirements may be more stringent, Version 2 of NTCIP 2301 – Simple Transportation Management Framework Application Profile states that the response time to a Simple Network Management Protocol (SNMP) or Simple Transportation Management Protocol (STMP) command shall be ≤100 milliseconds ± 1 millisecond per byte (*6*). A performance test procedure can measure this value.

CHAPTER 3:
RESOURCES TO SUPPORT TESTING

OVERVIEW

In this chapter, the researcher discusses what major organizations and agencies are doing in support of NTCIP testing. This chapter begins with a discussion and background information on the Enterprise and I-95 Corridor Coalition and their test procedures. A discussion of two new testing-related documents from the NTCIP standards group and some of their current plans follows. The researcher follows this with a description of the NTCIP testing programs at the Idaho Transportation Department and at the Florida Department of Transportation. Next, we provide a discussion of the American Association of State Highway and Transportation Officials (AASHTO) project on traffic signal controller interoperability and interchangeability. Finally, the chapter concludes with a summary of these resources.

GROUP EFFORTS

Through the researcher's involvement with the NEMA NTCIP Technical Committee, it is known that concerns about NTCIP testing were being raised as early as 1994. This lead to FHWA's sponsorship of the development of the NTCIP Exerciser in 1996 (*7*). Since that time, there have been numerous efforts to formalize test procedures for use with the NTCIP Exerciser and newer products. The following is a discussion of those efforts.

Enterprise and I-95 Corridor Coalition

The Enterprise Consortium is by far the most active group developing testing procedures. The Enterprise Program is a pooled-fund study with member agencies from North America and Europe (*8*). Its main purpose is to use the pooled resources of its members, private sector partners, and the United States federal government to develop, evaluate, and deploy ITS. The members of the consortium are as follows:

- Arizona Department of Transportation (DOT),
- Colorado DOT,
- Dutch Ministry of Transport,
- Federal Highway Administration,

- Iowa DOT,
- Kansas DOT,
- Ministry of Transportation of Ontario,
- Minnesota DOT,
- New Mexico State Highway Transportation Department,
- Transport Canada,
- Virginia DOT, and
- Washington State DOT.

The consortium has sponsored development of the following NTCIP test procedures:

- Class B Test Procedure,
- SNMP Test Procedure,
- Global Objects Test Procedure,
- DMS Test Procedure, and
- ESS (Environmental Sensor Station) Test Procedure.

The I-95 Corridor Coalition is an alliance of transportation and other agencies extending from Maine to Florida along I-95 (*9*). The coalition provides a forum for issues of common interest. One particular issue is the use of NTCIP-compliant DMSs. The coalition worked with the Enterprise group and the Federal Highway Administration to fund the development of an active testing tool that runs the Enterprise Consortium test procedures. The test tool supports the following communications interface standards:

- NTCIP 2301 – Simple Transportation Management Framework Application Profile (SNMP only),
- NTCIP 2201 – Transportation Transport Profiles (Null only) (*10*),
- NTCIP 2202 – Internet (UDP/IP and TCP/IP) Transport Profile (*11*),
- NTCIP 2101 – Point-to-Multi-Point Protocol Using RS-232 Subnetwork Profile (*12*),
- NTCIP 2103 – Point-to-Point Protocol over RS-232 Subnetwork Profile (*13*), and
- NTCIP 2104 – Ethernet Subnetwork Profile (*14*).

These communications protocols cover operation over standard network and non-networked (direct connect) environments. The physical interfaces cover serial RS-232, dial-up modem, and

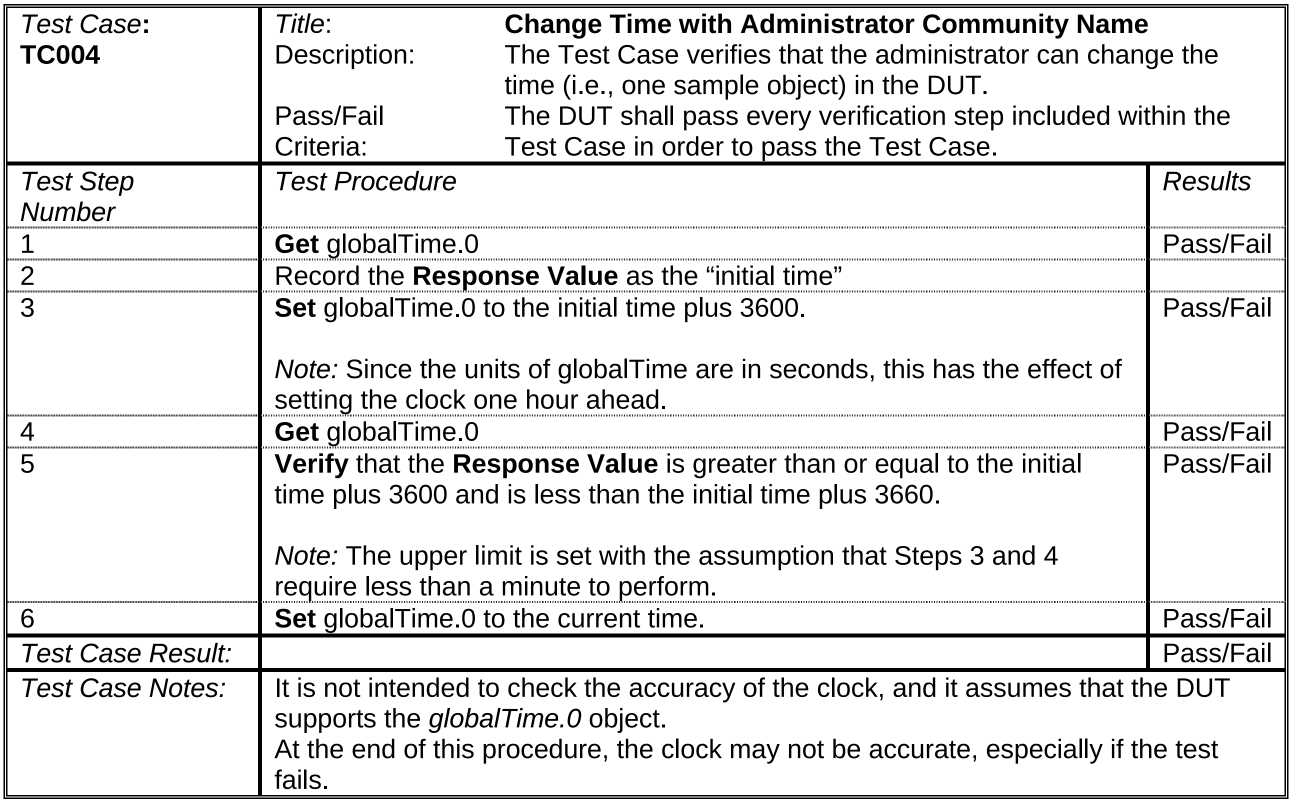
Ethernet connections. Given the broad base of support from the various agencies, the Enterprise test procedures have met with wide acceptance.

NTCIP Program

NTCIP standards do not currently contain material that describes how to test for conformance. However, the NTCIP Program Manager informed the researcher that two NTCIP working groups have just recently submitted project proposals for 2006 funding for inclusion of test procedures in the NTCIP standards (Vicki Schofield, July 20, 2005). To date, a number of outside groups and organizations have developed various test procedures; however, none of them has yet to receive formal recognition by the Joint Committee on NTCIP or the NTCIP WGs. A number of test procedures are expected make their way into NTCIP standards by the end of 2007. These test procedures will define the standard methods and criteria to which conformance to NTCIP standards are to be measured.

A number of procedures have received de facto industry approval. A dozen or more state DOTs use the Enterprise test procedures. Florida DOT has a surveillance camera pan, tilt, and zoom (PTZ) set of procedures that a number of vendors use to show compliance. As part of a New York City project, a traffic equipment manufacturer has prepared an extensive test procedure for traffic signal controllers. These procedures and others will likely serve as the basis for what is included in future versions of the NTCIP standards. The biggest impediment to including the test procedures, so far, has been the issue of format and language. The release of NTCIP 8007 – Testing and Conformity Assessment Documentation within NTCIP Standards Publications (NTCIP 8007-TEST) resolves the issue *(15)*. This standard is currently in the formal approval stage.

NTCIP 8007-TEST defines the format and language that should be used for describing testing procedures within NTCIP standards *(15)*. The standard prescribes a natural language approach to describe procedures and the steps to follow. There are some technology-specific keywords, but the terminology is meant to be implementation independent. Any testing tool developed for the technology should be able to implement the procedures. Figure 1 is one of the test procedure samples extracted from NTCIP 8007-TEST.



Test Case: **TC004**	*Title*: **Change Time with Administrator Community Name** Description: The Test Case verifies that the administrator can change the time (i.e., one sample object) in the DUT. Pass/Fail Criteria: The DUT shall pass every verification step included within the Test Case in order to pass the Test Case.	
Test Step Number	*Test Procedure*	*Results*
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 are 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 Result:		Pass/Fail
Test Case Notes:	It is not intended to check the accuracy of the clock, and it assumes that the DUT supports the *globalTime.0* object. At the end of this procedure, the clock may not be accurate, especially if the test fails.	

Figure 1. NTCIP 8007 Example Test Procedure.

With NTCIP 8007-TEST, the NTCIP working groups will have a uniform and consistent format for defining test procedures within their standards (*15*). The definition of test procedures within the standards will go a long way in ensuring that a device that claims conformance to the standards, in fact, does so.

Through the researcher's work with the NTCIP Testing and Conformity Assessment (TCA) Working Group, it is known that the WG is currently developing an information report NTCIP 9012 – Testing and Conformity Assessment User Guidance to the Issues. The intent of this report is to provide guidance to agencies in developing an NTCIP testing program for their procurements. While the primary focus of the report is NTCIP, it includes discussions of the following issues:

- prototype testing and inspection,
- design approval testing,
- factory acceptance testing,

- incoming device testing,
- site acceptance, and
- observation testing.

The report treats the testing of conformance to the standard, testing of compliance to specification, and performance testing as separate and distinct processes. This treatment contrasts somewhat with the researcher's view that the three processes are not that distinct in that conformance and performance testing are an integral part of compliance testing.

The TCA WG has already published a white paper entitled "ITS Standards Testing – A State of the Practice Report" (*16*). The paper centers on a survey of 10 state agencies. The paper reviews the testing approaches used by these agencies, provides insight as to why the agencies have selected these approaches, and captures their experiences in using these approaches. The authors caution that the review should be taken in light of its limitations. The interviews reflect experiences related to testing dynamic message signs and may not apply to other devices. One should also consider the sample size of the survey.

Idaho Transportation Department

The National Institute for Advanced Transportation Technology (NIATT) at the University of Idaho, in conjunction with the Idaho Transportation Department, has a project that will involve a test for the implementation of NTCIP standards in a small-town traffic control system (*17*). The project focuses on development and application of NTCIP standards for design, implementation, and testing of traffic signal timing plans using real-time hardware-in-the-loop simulation. One of the objectives is to investigate a new application area for controller interface device (CID) technology through developing and testing a prototype to use the CID and the Corridor Simulator (CORSIM) simulation-modeling tool to test traffic controller conformance to NTCIP communications standards.

The significance of the project is that it is the first to state an intention to test the following NTCIP standards:

- NTCIP 1202 – Object Definitions for Actuated Traffic Signal Controller Units (*4*) and
- NTCIP 1210 – Field Management Stations – Part 1: Object Definitions for Signal System Masters (NTCIP 1210-FMS) (*18*).

Florida Department of Transportation

The Florida Department of Transportation (FDOT) Traffic Engineering and Research Lab (TERL) has a project involving NTCIP testing. The objective of the testing effort at TERL is to facilitate efficient and effective testing of the implementation of NTCIP standards by traffic control devices. These efforts range from providing support to FDOT in development of specifications to implementation of testing programs. Included in these efforts are development and enhancement of testing tools as well as development of test procedures. The main efforts of the last phase focused on development and implementation of a DMS NTCIP qualification program (*19*).

As part of their DMS Qualification Program, TERL developed the following documentation:

- program summary,
- overview of NTCIP and relevant protocols,
- testing process documentation,
- test procedure, and
- testing macros for NTCIP Exerciser.

After conducting research into testing tools, TERL is now working with Intelligent Devices, Inc., with the aim of replacing the NTCIP Exerciser with DeviceTester as their primary testing tool (*20*). The researcher has also learned that Intelligent Devices has developed a set of CCTV test procedures as part of another Florida project (Curtis Herrick, August 27, 2004).

AASHTO

In 2003, the Interoperability Test Lab (ITL) at TTI undertook an AASHTO project to assess the effectiveness of NTCIP standards to promote interchangeability and interoperability of traffic signal controllers (*21*). ITL used some Enterprise and newly developed test procedures to check the basic functions and features of controllers as they relate to NTCIP 1201 – Global Object Definitions (NTCIP 1201-GLO), NTCIP 1202 – Object Definitions for Actuated Traffic Signal Controller Units, and NTCIP 2001 – Class B Profile (*22*, *4*, *23*).

The following is a list of some of the traffic signal controller functional test procedures that we developed as part of the project:

- transaction mode (configure phase start-up interval),
- intersection map (check of phase, overlap and pedestrian outputs, and miscellaneous phase status),
- system map (check of phase and overlap outputs, detector inputs, and coordination status),
- upload and download (compact encoding of phase data),
- time base scheduler (setup and execution of coordination plan changes),
- retrieve log data (trigger events and retrieve them), and
- response time (time check of how long to process messages).

SUMMARY OF RESOURCES

The transportation industry has spent considerable effort on developing test procedures for showing NTCIP conformance. The researcher believes that many of these procedures can serve as the basis for a TxDOT conformance testing program or act as baselines for developing their own procedures. Procedures for DMS, ESS, CCTV, and some elements for actuated signal controllers (ASCs) exist, and reuse would save some effort. The same situation applies to a number of the underlying protocols and global objects. Several groups provide SNMP test procedures, and some test tools come with extensive built-in SNMP procedures. Generic global object procedures and some customized for ASC are in the public domain. With the introduction of NTCIP 8007-TEST, the industry has a standardized template and format for documenting test procedures *(15)*. Agencies and groups now have a means to share what they have done and have it easily understood by others.

CHAPTER 4:
TESTING TOOLS

TEST TOOLS

There are a number of test tools that one could employ to evaluate conformance to NTCIP standards and compliance with TxDOT specifications. It is important to first clarify what the term "test tools" means as used in the context of this report. A test tool is the software, hardware, or both used to either provide command messages in the manner that a management application does or interpret the commands and generate a response in the manner that an ITS device does. The term also refers to software and/or hardware or both used to monitor or analyze the data exchanges between two or more such ITS devices or subsystems. The chapter begins with a description of various types of testing tools and terminology used to describe them. A review of a number of active, emulator, and instrumentation testing tools follows. The discussion of each type of testing tool includes a summary and comparison chart.

Active Testing Tools

The term active testing refers to the test tools that simulate a management application in sending commands and checking responses. The output of an active testing tool is expected to cause the device (or another ITS subsystem) to react with an appropriate response or change of state, both of which should be observable. In a typical active testing scenario, the test tool takes on the role of a central facility to exercise one or more field devices (or external subsystems).

For example, TxDOT would employ an active testing tool (acting as an Advanced Transportation Management System [ATMS]) to test the ability of a DMS to respond to command and control messages. An active testing tool can also test the ability of an external Traffic Management Subsystem to respond properly to queries and information updates from a Transit Management Subsystem. Figure 2 illustrates how an active testing tool is used.

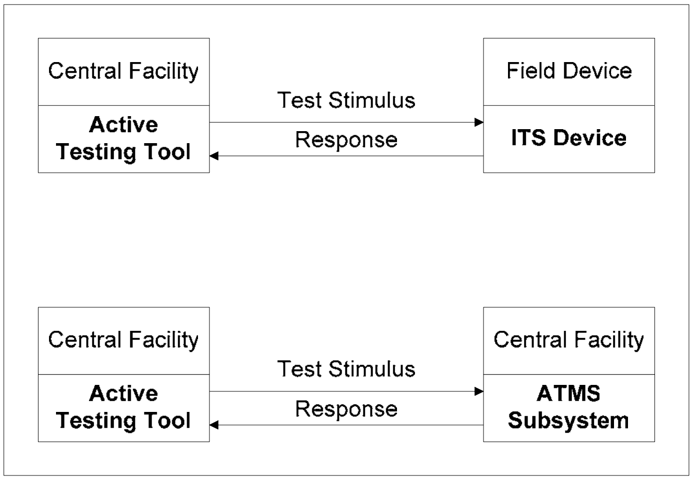


Figure 2. Active Testing Tool Scenarios.

There are two major subclasses of active testing tools: programmable and Go/No Go. Programmable refers the ability of a tool to run a user-developed test procedure. A programmable tool includes a scripting or macro language that allows a user to develop or modify a test procedure to suit his or her needs. Go/No Go refers to a tool that is developed for a specific application and only tests those functions or features that are built-in. Only the original vendor can change its operation.

Emulator Testing Tools

The term emulation testing refers to those testing tools that act like a field device or simulate the functions of a central facility or subsystem in how they respond to commands and control messages. This emulation consists of accepting commands or requests, responding with appropriate status reports, or performing information transfers when implementation that is under test initiates these actions or requests. A typical emulator testing tool takes on the role of a field device or subsystem and responds to the command messages. An emulator testing tool does not initiate commands as does an active testing tool and usually produces no observable result other than to return a proper status or acknowledgment response to the command.

TxDOT could employ an emulator testing tool as a substitute for one or more field devices or external subsystems to test the ability of the management application system to properly control and manage its field devices and subsystems. Additionally, an emulator testing tool may be used in addition to one or more "real" devices or subsystems to simulate or approximate full system loading effects. Some field device emulator testing tools have the ability to simulate multiple devices. Figure 3 illustrates the scenarios in which an emulator testing tool is used.

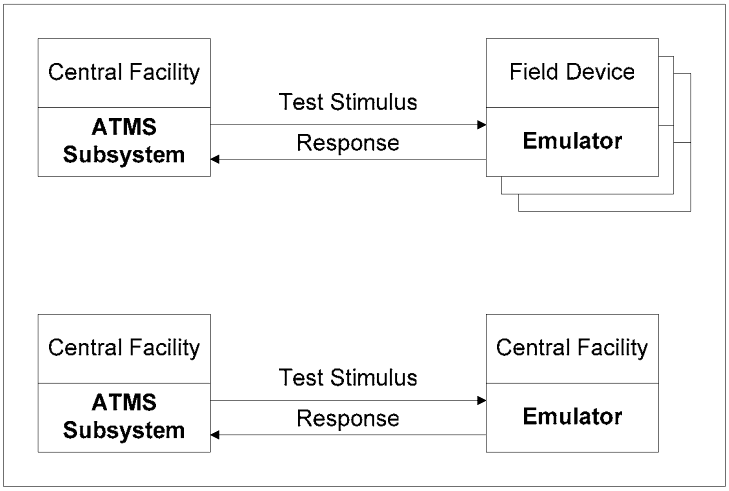


Figure 3. Emulator Testing Tool Scenarios.

Emulator testing tools can also be classified as programmable and Go/No Go. In the case of programmable emulator testing tools, the management information base (MIB) file that is loaded into the tool defines the application (ITS device) to emulate. There is usually a menu screen to allow entry of values to return. However, some programmable emulator testing tools allow a user to add behavior to object values. For example, each command sent to a field device could increment the value of an object. An object's value could be linked to the value of another object. Setting an object to a specific value could also change the value of another object. The most significant feature of programmable emulator testing tools is that they can return errors or produce unexpected results. In the case of a Go/No Go emulator testing tool, only those objects that the vendor built into the tools are accessible. They also have a menu screen to allow entry of values to return, but that is usually the extent of their capabilities.

Instrumentation

The term instrumentation refers to those test tools used to monitor, examine, and perhaps evaluate the data or information transfer across an interface between any combination of a management application, field devices, and/or subsystems. An instrumentation testing tool is commonly referred to as a data monitor or protocol analyzer. The typical configuration is to insert the instrumentation testing tool in parallel with the interface so as not to affect, interfere, or in any way participate in the data/information transfer dialogues. A typical instrumentation testing tool has the capability to display and analyze the data and information packets exchanged across the interface. Some can further perform protocol and/or content analysis and report on the validity of these exchanges in the context of the specific testing domain. The protocol and content analyses consist of breaking the data into fields, providing a text description of the protocol fields, and showing a text description of the data values. A key point in differentiating an instrumentation testing tool from the others is that an instrumentation testing tool does not represent a functional entity in the testing environment and does not initiate or respond to testing actions. An instrumentation testing tool could be used between a management application and field devices or subsystems at any time during the testing process with or without other testing tools. Figure 4 illustrates the general configuration when using an instrumentation testing tool. Instrumentation testing tools provide an independent, third-party view of the information exchanges and arbitrate faults.

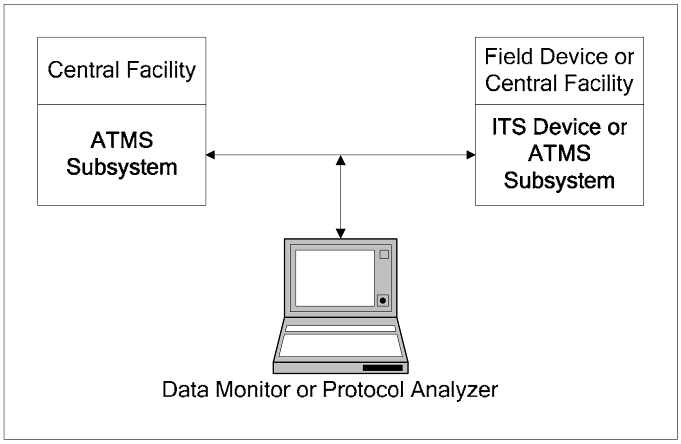

Central Facility
ATMS Subsystem
Field Device or Central Facility
ITS Device or ATMS Subsystem
Data Monitor or Protocol Analyzer

Figure 4. Instrumentation Testing Tool Scenarios.

Open and Closed Testers

There are two approaches to take when it comes to implementing a testing tool. One could use a general-purpose testing platform and customize it to specific requirements. Alternately, one could develop a stand-alone computer program with specific capabilities and functions and then distribute it to whoever desires it. This report uses the terms open and closed to refer to these two approaches.

An open tester is a general-purpose platform that can be customized to address specific functions using scripts or macros. An open testing tool's platform is generally tailored for a specific environment (i.e., SNMP) and may include some predefined procedures for the environment but relies on user-defined scripts or macros for the application-specific test functions. A closed tester is a computer program designed for specific testing tasks and only those tasks. It can act like different field devices or subsystems, but the functions that it tests for are fixed and cannot be changed except by the original developer of the program.

ACTIVE TESTING TOOL PRODUCTS

In a draft white paper, "Identification of Test Tools for Intelligent Transportation Systems Devices and Subsystems," prepared by Battelle for the U.S. Department of Transportation, Battelle identified five active testing tools. However, one major tool was misidentified, SimpleTester™[1]. It is used by a number of well-known companies as an active testing tool. The company that created the tool, in fact, has a customized version (SimpleTester for NTCIP) that adds support for Point-to-Multi-Point Protocol (PMPP) and is currently in the process of adding STMP. Thus, the list of active testing tools is:

- Chart II NTCIP Exerciser,
- DeviceTester for NTCIP,
- NTCIP Exerciser,
- NTester™[2],
- SimpleTester for NTCIP, and
- TxDOT NTCIP Tester.

The NTCIP Exerciser, DeviceTester for NTCIP, SimpleTester for NTCIP, and TxDOT NTCIP Tester are programmable. Chart II and NTester fall into the Go/No Go classification. A discussion of each of the tools follows.

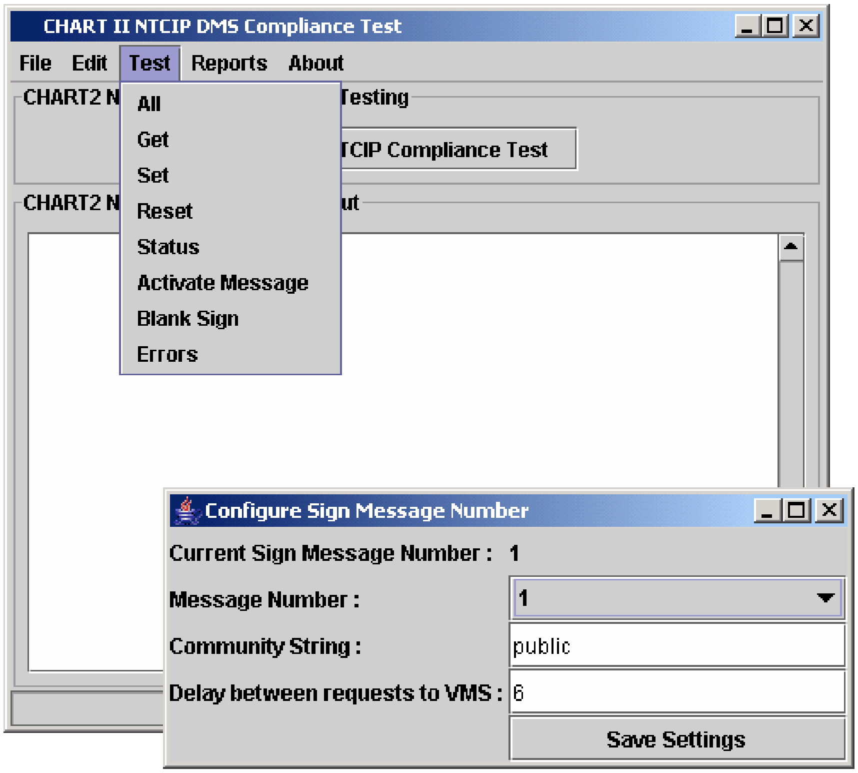
[1] SimpleTester™ is a trademark of SimpleSoft, Incorporated, Mountain View, California.
[2] NTester™ is a trademark of Trevilon Corporation, Herndon, Virginia.

Chart II NTCIP Exerciser

Maryland DOT developed the Coordinated Highways Action Response Team (CHART) II NTCIP Exerciser (*24*). The DOT uses it as a prequalification testing tool for DMSs. CHART II NTCIP Exerciser is a simple Go/No Go testing tool to help ensure NTCIP-DMS compatibility within the CHART ATMS environment. The core functionalities are the get and set of specific configuration objects, setting and activating a message on the DMS, blanking the DMS, getting specific status objects, and resetting the sign. An NTCIP-compliant sign must pass the test functions in order to be compliant with the CHART II environment.

This program tests approximately a dozen basic functions (such as "get date" and resetting a sign) related to the way that Maryland plans to use an NTCIP-conformant DMS in the CHART II system. The design is such that a vendor can take part of the CHART software and install it on a personal computer (PC). That PC will then be able to communicate with NTCIP-compatible signs the same way that CHART does. In this way, a vendor can see how the device would work with the CHART software. The tester generates a portable document format (PDF) report detailing which functions ran as expected and which exhibited some type of problem. Chart II NTCIP Exerciser only supports the PMPP serial interface as specified in NTCIP 1201-GLO. As a Go/No Go, there is no support for scripting or customization.

Figure 5 shows two screen images of the main Chart II NTCIP Exerciser user interface.



Current Sign Message Number : 1

Message Number : 1

Community String : public

Delay between requests to VMS : 6

Figure 5. CHART II Exerciser Main Interfaces.

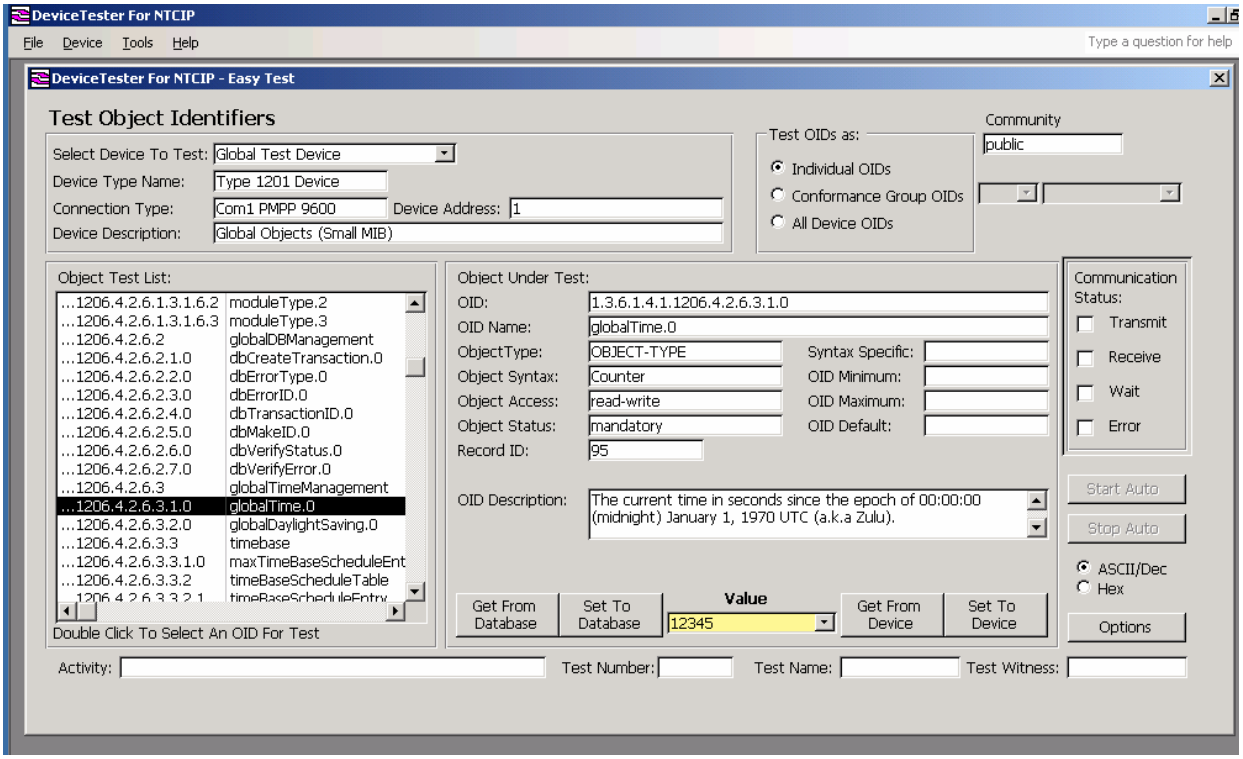
DeviceTester for NTCIP

DeviceTester for NTCIP is a database product with a communications interface that can serve as an active testing tool or mimic the functions of a field device. When operating in the "central operation" mode, the testing tool allows a user to generate commands that are sent as if they were coming from a management application such as ATMS. When operating in the "mimic" mode, the software responds to commands sent to it and can return values as if it were a field device. The tool can automatically check all NTCIP parameters, controls, and status information within a device, perform automatic error simulation, and create result logs and reports. DeviceTester for NTCIP is based upon a Microsoft Access®[3] database structure and includes interfaces for simple and advanced testing of NTCIP objects and conformance groups. It provides a scripting environment for creating and executing scripts for repetitive and automated testing. It logs all test results. DeviceTester for NTCIP can export data in a variety of formats for additional custom reporting. The testing tool can import MIBs for testing of any device type.

[3] Access is a registered trademark of Microsoft Corporation.

DeviceTester for NTCIP is the only active testing tool that currently has support for STMP and the Simple Fixed Management Protocol (SFMP). An add-on product, ActiveX®[4] Control, allows a programmer to build applications that use the NTCIP protocol functions built into DeviceTester for NTCIP. The resulting programs can be distributed royalty free.

DeviceTester for NTCIP was developed specifically for the NTCIP environment. The testing tool supports the communications interfaces defined within NTCIP, as well as any interface that is configurable on a PC. This includes PMPP, Point-to-Point Protocol (PPP) (dial-up), and Ethernet. Figure 6 shows a screen image of one of the user interfaces in DeviceTester for NTCIP. One feature of the Easy Test interface is that tests can be run on individual object identifiers, objects in a conformance group, or all device objects.



Test Object Identifiers

Select Device To Test: Global Test Device
Device Type Name: Type 1201 Device
Connection Type: Com1 PMPP 9600 Device Address: 1
Device Description: Global Objects (Small MIB)

Test OIDs as:
● Individual OIDs
○ Conformance Group OIDs
○ All Device OIDs

Community: public

Object Test List:

...1206.4.2.6.1.3.1.6.2	moduleType.2
...1206.4.2.6.1.3.1.6.3	moduleType.3
...1206.4.2.6.2	globalDBManagement
...1206.4.2.6.2.1.0	dbCreateTransaction.0
...1206.4.2.6.2.2.0	dbErrorType.0
...1206.4.2.6.2.3.0	dbErrorID.0
...1206.4.2.6.2.4.0	dbTransactionID.0
...1206.4.2.6.2.5.0	dbMakeID.0
...1206.4.2.6.2.6.0	dbVerifyStatus.0
...1206.4.2.6.2.7.0	dbVerifyError.0
...1206.4.2.6.3	globalTimeManagement
...1206.4.2.6.3.1.0	globalTime.0
...1206.4.2.6.3.2.0	globalDaylightSaving.0
...1206.4.2.6.3.3	timebase
...1206.4.2.6.3.3.1.0	maxTimeBaseScheduleEnt
...1206.4.2.6.3.3.2	timeBaseScheduleTable
...1206.4.2.6.3.3.2.1	timeBaseScheduleEntry

Double Click To Select An OID For Test

Object Under Test:
OID: 1.3.6.1.4.1.1206.4.2.6.3.1.0
OID Name: globalTime.0
ObjectType: OBJECT-TYPE Syntax Specific:
Object Syntax: Counter OID Minimum:
Object Access: read-write OID Maximum:
Object Status: mandatory OID Default:
Record ID: 95
OID Description: The current time in seconds since the epoch of 00:00:00 (midnight) January 1, 1970 UTC (a.k.a Zulu).

Value: 12345

Communication Status:
□ Transmit
□ Receive
□ Wait
□ Error

● ASCII/Dec
○ Hex

Activity: Test Number: Test Name: Test Witness:

Figure 6. DeviceTester's Easy Test Interface.

DeviceTester for NTCIP uses an internal script generator to automate repetitive test procedures. The script generator provides a limited set of commands to generate SNMP

[4] ActiveX is a registered trademark of Microsoft Corporation.

commands and check responses. Although the add-on ActiveX Control is not a script generator, it can serve the same purpose. ActiveX is an application programming interface that allows a program to use the internal functions of DeviceTester for NTCIP to generate commands and check responses. The design and layout of any ActiveX graphical user interface (GUI), however, is up to the program developer. Programs that use the ActiveX Control are typically written in one of the Microsoft visual programming languages such as Visual Basic®[5] or Visual C++®[6]. ActiveX Control comes with a distribution license to distribute the control with user applications.

DeviceTester for NTCIP comes preconfigured with three scripts for testing dynamic message signs. Included are scripts for downloading a message and activating it, displaying 20 different messages for 5 seconds each, and downloading and displaying 35 different changeable messages.

As a commercial product, the documentation that comes with DeviceTester for NTCIP is sparse in a number of areas. However, most of the user interfaces are very intuitive. The software comes with a block diagram of master tables, devices, and device types and shows their relationships. The diagram is invaluable when a user creates the tables, devices, and types. When entering a script, one should be careful about making mistakes. Response time of technical support may be an issue in some cases.

NTCIP Exerciser

The NTCIP Exerciser is a general-purpose testing and development tool for NTCIP systems developers, testers, and evaluators. The tool is non-device specific in that it uses a MIB to define the objects on which to operate. The NTCIP Exerciser has a built-in scripting capability that supports a robust scripting language to automate tasks and string operations together. It is very flexible and includes a majority of the features that one would expect. The only limitation is in regard to adding comments or remarks to a script. The run time display includes a full translation of transmitted and received data. This display is comparable to what would be found on an instrumentation testing tool. The NTCIP Exerciser supports two distinct communications interfaces: serial and dial-up. The serial interface is RS-232 and is fully programmable. The dial-up interface supports Challenge Handshake Authentication Protocol (CHAP) for secure

[5] Visual Basic is a trademark of Microsoft Corporation.

[6] Visual C++ is a registered trademark of Microsoft Corporation.

communications. The NTCIP Exerciser can act like an emulation testing tool. When configured for field operation (responds to commands), a MIB defines the type of device that it emulates and values that are returned in a response.

The NTCIP Exerciser was created with funding from FHWA and is currently in the public domain. The testing tool comes with documentation and all source code used to create it. Unfortunately, there is no funding for support or further development. The NTCIP Exerciser is generally considered to be “the” reference for developing implementations. It was the first tool specifically written for NTCIP, and as such, many people still use it to test implementations. Numerous manufacturers have developed procedures to exercise various functions and demonstrate conformance to the NTCIP standards. One traffic signal controller manufacturer has a full set of scripts to configure controllers for different modes of operation and test a significant portion of their controller’s implementation of NTCIP.

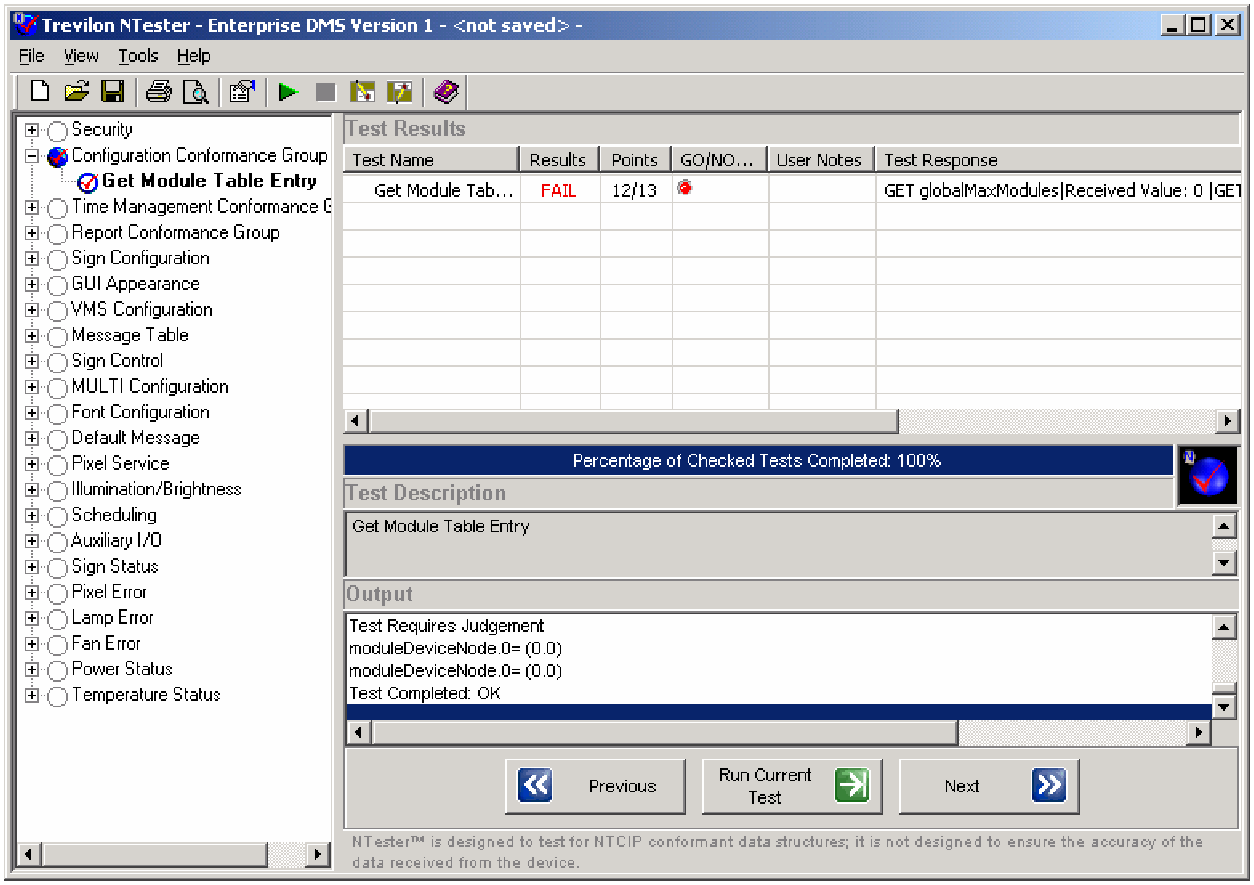
A number of state DOTs and other government agencies are still using NTCIP Exerciser as part of their qualification and acceptance procedures. Even without support and with several known problems and software bugs, many people still use the NTCIP Exerciser to debug and diagnose errors. When the Enterprise Consortium sponsored the development of test procedures that focus on the specifics of the application, network, and subnetwork level profiles (communications layers), they were initially implemented as scripts on the NTCIP Exerciser. When the Oregon Department of Transportation sponsored the development of procedures related to global objects, DMSs, and ESSs, these too were first implemented on the NTCIP Exerciser. The NTCIP Exerciser also served as the prototype for NTester.

NTester

NTester is a Go/No Go active testing tool for checking NTCIP conformance of DMSs, ESSs, and CCTV camera control. NTCIP uses the Enterprise Consortium test procedures as the basis for checking conformance. NTester is available for free download through a cooperative effort of the Enterprise Consortium, the I-95 Corridor Coalition, and FHWA (*25*). Given its basis of support, NTester is the most widely used active testing tool.

NTester has a simple GUI that allows an individual unfamiliar with NTCIP to test supported ITS devices. Selection of the physical interface and communications protocols is quite simple. The testing interface uses three distinctive panes to present information. The test selection pane (left portion of Figure 7) allows a user to select the tests to run easily and quickly. The results pane (top right portion of Figure 7) presents the results of the tests. Finally, the test description pane (bottom right of Figure 7) displays detailed information about the test results.

NTester supports the communications interfaces defined within NTCIP. This includes any interface configurable on a PC, including PMPP and PPP (dial-up). NTester does not support STMP.



Test Name	Results	Points	GO/NO...	User Notes	Test Response
Get Module Tab...	FAIL	12/13	●		GET globalMaxModules|Received Value: 0 |GET

Percentage of Checked Tests Completed: 100%

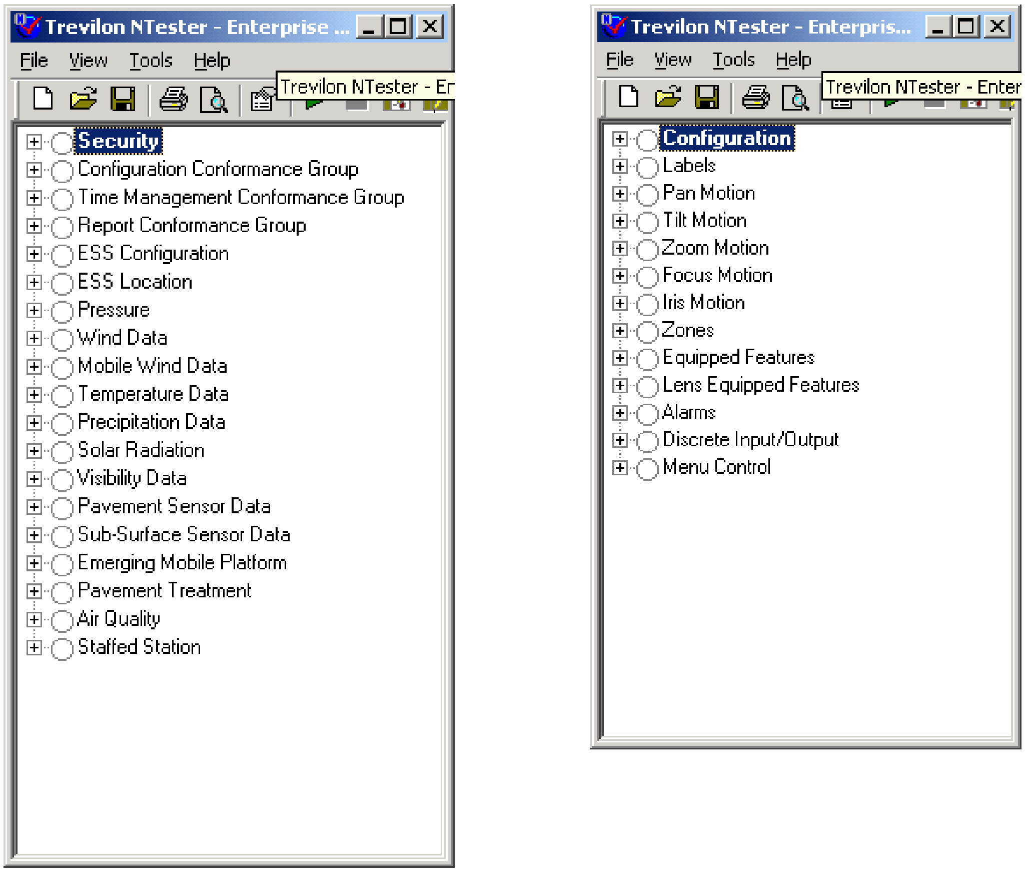
Get Module Table Entry

```
Test Requires Judgement
moduleDeviceNode.0= (0.0)
moduleDeviceNode.0= (0.0)
Test Completed: OK
```

NTester™ is designed to test for NTCIP conformant data structures; it is not designed to ensure the accuracy of the data received from the device.

Figure 7. NTester DMS Test Interface.

The test suite for the supported field devices is quite extensive. As one can see from Figure 7, there are 22 different test areas associated with a DMS. Figure 8 illustrates the 19 test areas for ESS and the 13 test areas for CCTV. The test areas contain one or more subfunctions.



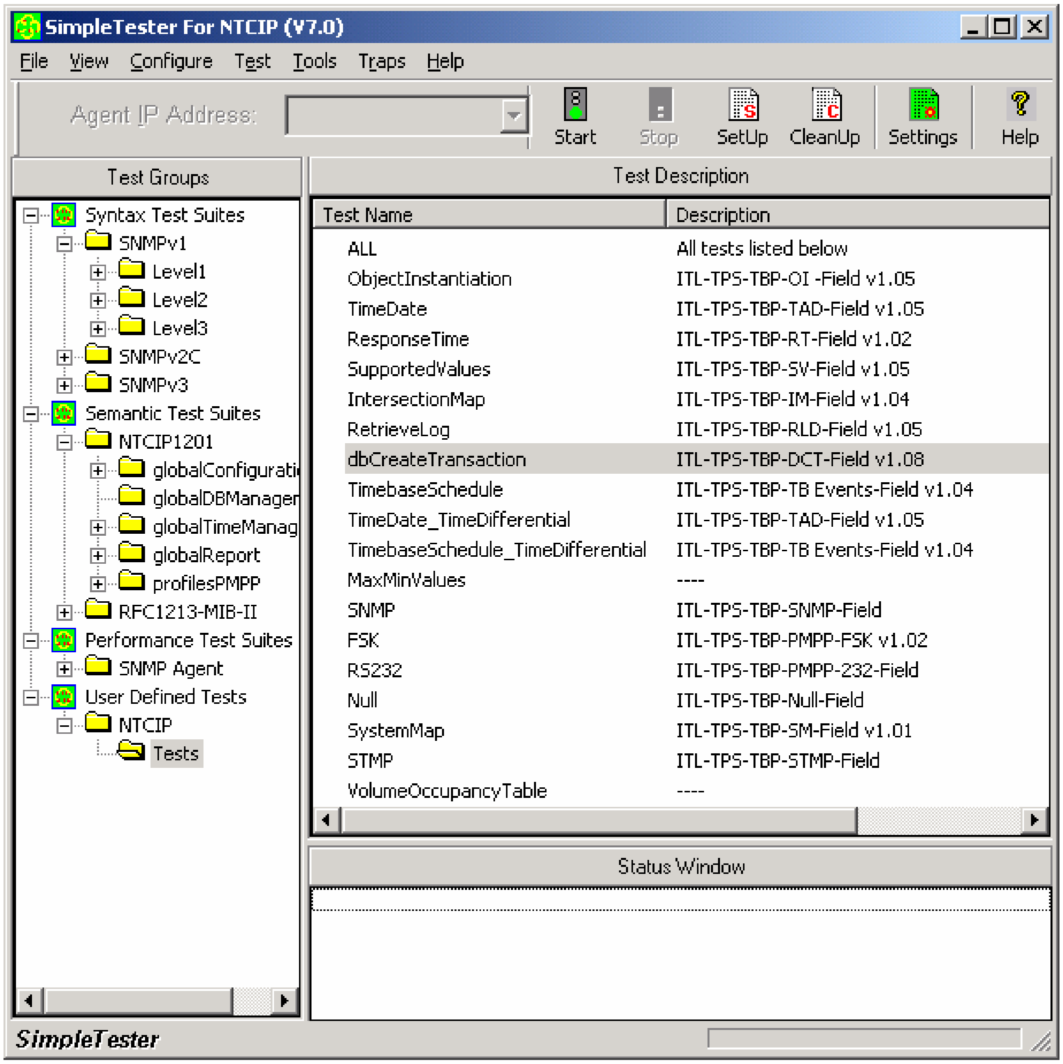
- Security
- Configuration Conformance Group
- Time Management Conformance Group
- Report Conformance Group
- ESS Configuration
- ESS Location
- Pressure
- Wind Data
- Mobile Wind Data
- Temperature Data
- Precipitation Data
- Solar Radiation
- Visibility Data
- Pavement Sensor Data
- Sub-Surface Sensor Data
- Emerging Mobile Platform
- Pavement Treatment
- Air Quality
- Staffed Station

- Configuration
- Labels
- Pan Motion
- Tilt Motion
- Zoom Motion
- Focus Motion
- Iris Motion
- Zones
- Equipped Features
- Lens Equipped Features
- Alarms
- Discrete Input/Output
- Menu Control

Figure 8. NTester ESS and CCTV Test Suites.

SimpleTester

SimpleTester for NTCIP is a general-purpose testing tool for NTCIP systems developers, testers, and evaluators. Like the Exerciser, the tool is non-device specific in that it uses a MIB to define the objects on which to operate. After internally compiling the appropriate MIB files, it uses this knowledge in predefined tests to send requests and analyze the responses to check MIB and SNMP protocol conformance. SimpleTester also includes a MIB browser, script generation, and a Tool Command Language (Tcl) script interpreter. Figure 9 shows an example of the user interfaces in SimpleTester for NTCIP.



Test Name	Description
ALL	All tests listed below
ObjectInstantiation	ITL-TPS-TBP-OI -Field v1.05
TimeDate	ITL-TPS-TBP-TAD-Field v1.05
ResponseTime	ITL-TPS-TBP-RT-Field v1.02
SupportedValues	ITL-TPS-TBP-SV-Field v1.05
IntersectionMap	ITL-TPS-TBP-IM-Field v1.04
RetrieveLog	ITL-TPS-TBP-RLD-Field v1.05
dbCreateTransaction	ITL-TPS-TBP-DCT-Field v1.08
TimebaseSchedule	ITL-TPS-TBP-TB Events-Field v1.04
TimeDate_TimeDifferential	ITL-TPS-TBP-TAD-Field v1.05
TimebaseSchedule_TimeDifferential	ITL-TPS-TBP-TB Events-Field v1.04
MaxMinValues	----
SNMP	ITL-TPS-TBP-SNMP-Field
FSK	ITL-TPS-TBP-PMPP-FSK v1.02
RS232	ITL-TPS-TBP-PMPP-232-Field
Null	ITL-TPS-TBP-Null-Field
SystemMap	ITL-TPS-TBP-SM-Field v1.01
STMP	ITL-TPS-TBP-STMP-Field
VolumeOccupancyTable	----

Figure 9. SimpleTester for NTCIP User Interface.

SimpleTester was originally developed for checking networking devices that support SNMP. The manufacturer claims that it is the most popular Windows®[7]-based SNMP agent tester that automatically tests computer network devices. Hundreds of networking companies worldwide use SimpleTester, including telecommunications giants like Cisco, Lucent, Nortel, 3Com, and Motorola. It comes with built-in procedures for exhaustive testing of SNMP- and MIB-defined objects. A version specifically customized for NTCIP, SimpleTester for NTCIP, was introduced in 2002. The only major difference between SimpleTester and SimpleTester for NTCIP is that the latter has support for NTCIP PMPP with RS-232 interface as specified in

[7] Windows® is a registered trademark of Microsoft Corporation.

NTCIP-GLO. One can achieve true multi-drop operation by use of external RS-232 to RS-485 converters. SimpleTester also supports any other interface configurable on a PC.

SimpleTester for NTCIP has two methods for generating automated test procedures: a script generator and a Tcl interpreter. The script generator allows a user to define a group of variables and then perform SNMP operations on them. This is similar to testing a conformance group but differs in that you can separate out read-write and read-only objects. This is more in tune with actual operations. The Tcl interpreter uses a simple, programmable syntax (programming language) that allows full access and control over all variables and operations. Over half a million developers worldwide use Tcl and incorporate it in many commercial products. The outstanding features of Tcl are that it is simple to understand and use. Due to its popularity, it enjoys significant Internet resources and support. Although SimpleTester for NTCIP currently lacks direct support for STMP, SimpleSoft added a send-serial command to Tcl, allowing a user to construct and send an STMP message. The send-serial command also permits a user to introduce errors in PMPP messages.

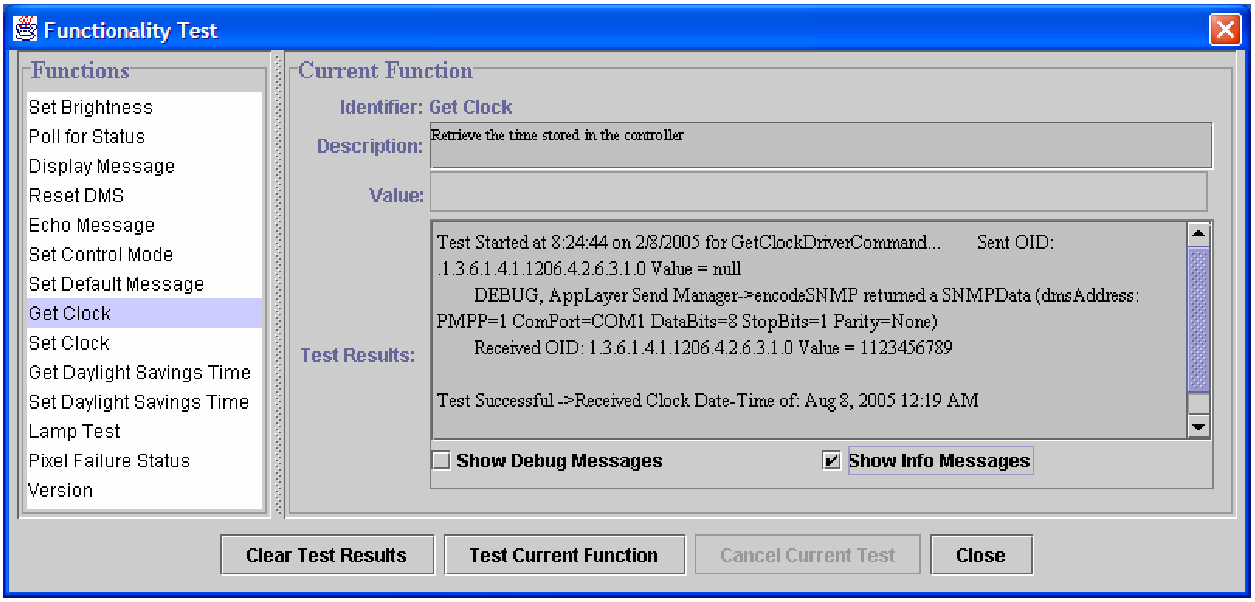
One of the test procedures developed by the Enterprise Consortium calls for the use of SimpleTester to check NTCIP implementations that use PPP as specified in NTCIP 1203 – Object Definitions for Dynamic Message Signs (DMS) (*26*). Another set of procedures developed under an AASHTO-NTCIP Laboratory Testing for Actuated Signal Controllers Project focuses on traffic signal controllers and is available in pseudo-NTCIP 8007-TEST format and as Tcl scripts (*27*, *15*). As shown in Figure 9, the traffic signal controller procedures focus on five underlying protocol tests, three general procedures for checking object instantiation (support) and supported values (range), and eleven functional procedures.

As a commercial product, the documentation that comes with SimpleTester for NTCIP is extensive. The Tcl commands specific to SimpleTester for NTCIP are fully documented, and there are numerous example scripts. Although the Tcl commands are fully documented on the Internet, the SimpleSoft documentation does not include a summary or quick reference. Most of the user interfaces are very intuitive. Response time of technical support should not be an issue.

TxDOT NTCIP Tester

TxDOT NTCIP Tester (TNT) is an active testing tool developed specifically for testing DMS compliance with the TxDOT Special Specification for Dynamic Message Signs, November

1998 (*28*). Southwest Research Institute designed TNT for TxDOT. TNT only supports RS-232, Ethernet, and modem interfaces. Although the device does not support testing of PPP, CHAP, and STMP directly, the documentation describes test procedures for checking them if a vendor's master software is available. It includes a MIB browser and object identifier (OID) tests to perform individual SNMP get and set operations on supported objects. A test suite automates a number of the get, set, and functional test operations. It has a JavaScript scripting capability so that a user can add test procedures. Figure 10 is an image of the functionality test user interface screen in TNT.



Identifier: Get Clock

Description: Retrieve the time stored in the controller

Value:

Test Results:

```
Test Started at 8:24:44 on 2/8/2005 for GetClockDriverCommand...      Sent OID:
.1.3.6.1.4.1.1206.4.2.6.3.1.0 Value = null
      DEBUG, AppLayer Send Manager->encodeSNMP returned a SNMPData (dmsAddress:
PMPP=1 ComPort=COM1 DataBits=8 StopBits=1 Parity=None)
      Received OID: 1.3.6.1.4.1.1206.4.2.6.3.1.0 Value = 1123456789

Test Successful ->Received Clock Date-Time of: Aug 8, 2005 12:19 AM
```

☐ Show Debug Messages ☑ Show Info Messages

Figure 10. TNT Functionality Test Interface.

Active Testing Tools Summary

Each of the active testing tools that support NTCIP has its own set of advantages and disadvantages, and no one tool stands out above the rest. Because it is free and has de facto endorsement by the Enterprise Consortium, NTester seems to lead the pack with its support of DMS, ESS, and CCTV test procedures. For conformance and compliance testing of ITS devices that do not have defined procedures or where there is a need to enhance existing ones, SimpleTester for NTCIP provides a good platform for fast prototyping and full development. SimpleTester has a beta release that adds some support of STMP, but DeviceTester for NTCIP is the only active testing tool that currently supports it. Otherwise, the features and functions of

DeviceTester for NTCIP are very similar to SimpleTester for NTCIP. Intelligent Devices, the manufacturer of DeviceTester for NTCIP, also offers ActiveX Control or library of communications functions for developers who would like have complete control over the user interface and develop scripts in Visual Basic or one of the other .NET programming languages. Because TNT was specifically developed with TxDOT's DMS requirements in mind, it is "the" tool for checking compliance to TxDOT's DMS specification. In cases where one is just looking for basic functionality except for STMP, the NTCIP Exerciser still proves to be useful. Even with its known software bugs and crashes, a number of people still use it extensively for isolating problems. Table 1 provides a quick comparison chart of the active testing tools. The above assessment may not hold true a year from now. The developer of NTester has stated his desire to develop a "professional" version by the end of 2005. The developer is also considering something similar to Intelligent Devices' ActiveX Control. The professional version of NTester will reportedly include full user scripting capability and add some support for STMP. The price of the "professional" version should be competitive with SimpleTester for NTCIP and DeviceTester for NTCIP and will allow testing of any NTCIP ITS device. SimpleTester for NTCIP is also currently undergoing revision to support STMP and to add a database similar to DeviceTester for NTCIP. DeviceTester's programmable interface (ActiveX Control) comes with an unlimited distribution license. This may be more attractive if TxDOT desires to tailor test procedures to their particular specifications and then freely distribute the software to anyone within the divisions or districts.

Table 1. Active Testing Comparison Chart.

		Active Testing Tool				
		NTester	SimpleTester for NTCIP	DeviceTester for NTCIP	TxDOT NTCIP Tester	NTCIP Exerciser
Feature	Built-In Procedures	DMS, ESS, CCTV	SNMP, Global	DMS, ESS, CCTV	DMS	No
	Scripting	No	Excellent	Fair	Fair	Good
	STMP	No	Limited	Yes	No	No
	Physical Interfaces	All	All	All	All	PMPP PPP
	Byte Stream Debugging	No	No	No	No	Yes
	Cost	Free	$$$$[8]	$$$$	Free	Free

EMULATOR TESTING TOOL PRODUCTS

The draft of "Identification of Test Tools for Intelligent Transportation Systems Devices and Subsystems" prepared by Battelle identifies five emulator testing tools:

- DeviceTester for NTCIP,
- NTCIP Exerciser,
- SimpleAgentPro®[9],
- Daktronics Emulator, and
- TCIP Tester.

The emulation capabilities of DeviceTester for NTCIP, NTCIP Exerciser, and SimpleAgentPro are discussed in the following subheadings.

Daktronics Emulator is a vendor-specific tool and is not discussed any further. The Federal Transit Administration originally had plans to develop a Transit Communications Interface Profiles (TCIP) Tester to validate and verify an Extensible Markup Language (XML) interface used in center-to-center transit applications. Funding for the TCIP Tester, however, is on hold.

[8] $s = Number of figures in price.

[9] SimpleAgentPro® is a registered trademark of SimpleSoft, Incorporated, Mountain View, California.

DeviceTester for NTCIP

Although DeviceTester for NTCIP does not come with any documentation that indicates that it has the capability, it does have the ability to act as an emulator testing tool. The selection of “Dynamic Message Sign Emulator” enables it to respond to commands related to objects in the MIB. The MIB does not have to be the DMS MIB. It will emulate any object contained in any compiled MIB. A value stored in the emulator’s database can be retrieved. A value can be set in the emulator’s database by a management application. Both SNMP and STMP operations are supported.

DeviceTester for NTCIP was developed specifically for the NTCIP environment. It supports the communications networking and non-networking protocols and the interfaces defined within NTCIP. This includes TCP/IP, UDP/IP, Ethernet, PMPP, and PPP (dial-up). It can support any interface configurable on a PC.

DeviceTester for NTCIP does not have any script support associated with the emulation mode. There are no test procedures associated with DeviceTester for NTCIP when operating in emulation mode. There is one quirk in that the software requires a value to store in the database before it can be retrieved. There are no default values, and in some cases, it will return the wrong syntax if the value is not initialized.

NTCIP Exerciser

NTCIP Exerciser has the ability to act as a central device and as a field device. “Field mode” can emulate how a field device such as a traffic signal controller, dynamic message sign, or camera PTZ controller responds to a command from a management application. However, beyond simply setting or returning a value there is no behavior associated with an object or any relationship to others.

NTCIP Exerciser was created with funding from FHWA and is currently in the public domain. The Exerciser comes with documentation and all source code used to create it. Unfortunately, support and any further development are not funded. The same debugging information available in the active testing mode is also available in the emulation mode.

The Exerciser supports two distinct communications interfaces: serial and dial-up. The serial interface is RS-232 and is fully programmable. An external RS-232 to RS-485 converter can be used to support multi-drop operation. The dial-up interface supports CHAP for secure communications. Although undocumented, it may also support an Ethernet interface.

The NTCIP Exerciser scripting capability does not extend to the field (emulation) mode. There are no procedures associated with NTCIP Exerciser when operating in that mode, either.

SimpleAgentPro

SimpleAgentPro is a field device emulator testing tool with an easy to use GUI that can simulate an entire system made up of thousands of field devices. Each simulated device can support its own MIBs, data, and address. SimpleAgentPro has a unique ability to create object default values from the MIB or generate them dynamically. The use of Tcl-based scripting allows for advanced modeling of device behavior, creation of error scenarios, and expression of inter-relationships between the variables defined in a MIB.

SimpleAgentPro was developed for the general computer networking environment. SimpleSoft's client base includes some of the most respected company names in the telecommunications industry. Although SimpleAgentPro does not currently support PMPP, SimpleSoft is considering adding an NTCIP-specific interface. SimpleAgentPro supports any interface configurable on a PC.

SimpleAgentPro has the ability to add behavior and relationships to the objects in a MIB. It can configure the values returned in responses to commands. Each object's value can be characterized as fixed, sequential, random, random incremental, clock based, and last sent value. The Tcl scripts can also add complex behaviors and define inter-relationships between MIB variables. SimpleAgentPro does not come with any NTCIP-specific test procedures.

Emulator Testing Tools Summary

The most fully featured and capable emulator testing tool is SimpleAgentPro. With its ability to emulate hundreds of devices simultaneously and customize the behavior of each device, it is by far the most powerful tool. These features, however, come with a four-figure price. While DeviceTester for NTCIP lacks the sophisticated features of SimpleAgentPro, it has an advantage over NTCIP Exerciser in that it maintains test values in a database and supports STMP. Unlike SimpleAgentPro, DeviceTester for NTCIP is not a separate product; it can serve as both an active testing tool and an emulator testing tool. NTCIP Exerciser includes a MIB compiler, so it supports any device. NTCIP Exerciser's software bugs also detract from its use. Table 2 provides a comparison chart of the key features of the emulator testing tools.

Table 2. Emulator Testing Tool Comparison Chart.

		Emulator Testing Tool		
		SimpleAgent Pro	DeviceTester for NTCIP	NTCIP Exerciser
Feature				
	Scripting	Yes	No	No
	STMP	No	Yes	No
	Physical Interfaces	All	All	PMPP PPP
	Cost	$$$$	$$$$	Free

INSTRUMENTATION TESTING TOOL PRODUCTS

Instrumentation testing tools allow a tester to view the details of any protocol exchange. They can isolate error conditions down to either the source or responder and can document exactly what was sent. The Battelle report on test tools for ITS itemized numerous products that can serve as instrumentation testing tools. With one exception, the products were developed for the general computer networking environment. They range from rather simple software tools that allow one to view the data exchanged over an RS-232 connection to hardware products that can decode virtually every protocol used in computer networks that is sent over any type of physical interface. Some of the products have add-ons or a capability to add new user-defined protocols. To limit the number of products discussed, the following is a discussion of the only two that include specific support for the protocols defined within NTCIP: FTS®[10] for NTCIP and TrafficView™[11].

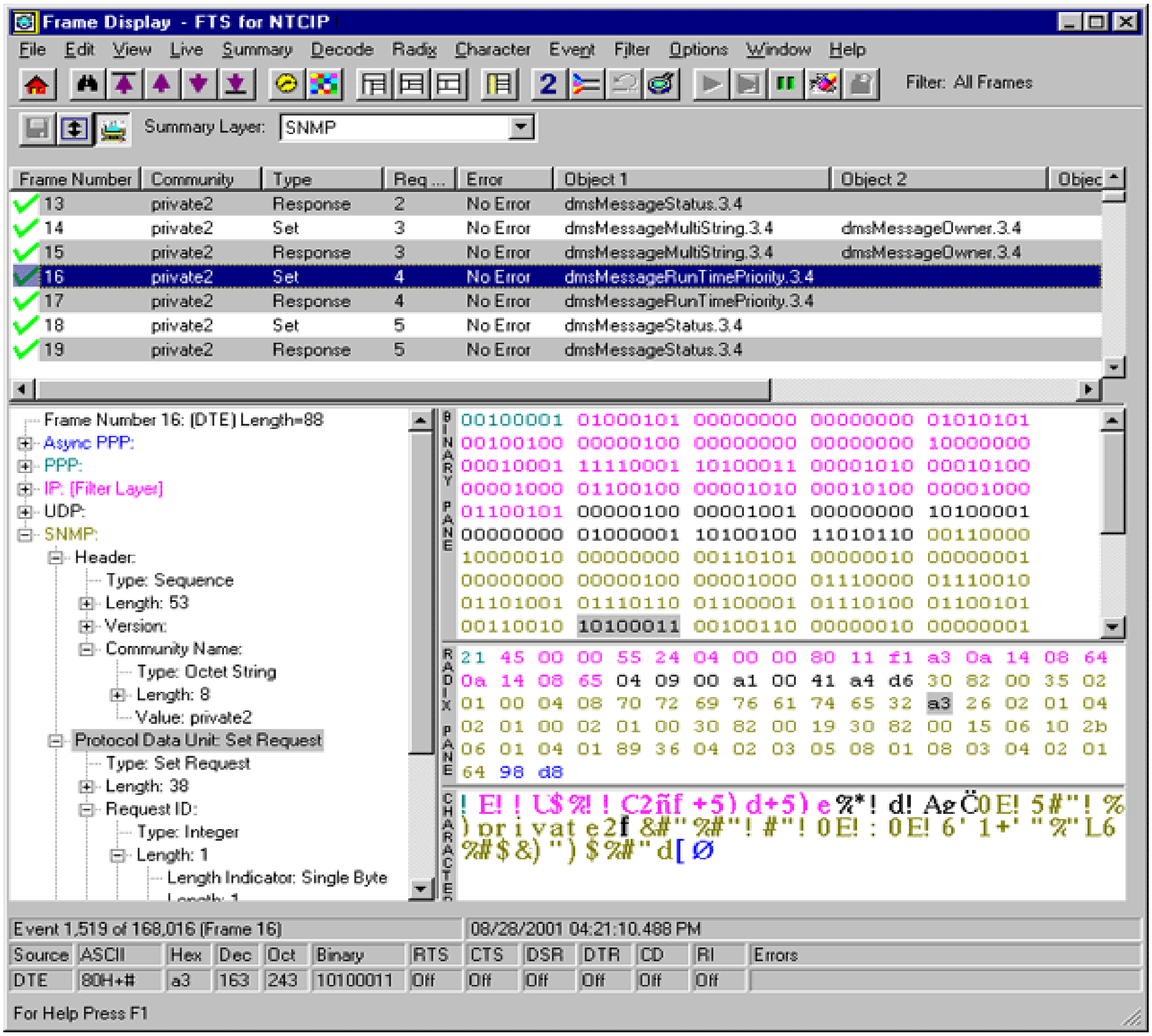
FTS for NTCIP

FTS for NTCIP is a general-purpose monitor and analyzer that is customized to add support for NTCIP protocols. It can decode all levels in a communications stack from the

[10] FTS® is a registered trademark of Frontline Test Equipment, Incorporated, Charlottesville, Virginia.
[11] TrafficView™ is a trademark of Klos Technologies, Incorporated, Cortland, New York.

information level down to the subnetwork level (see the NTCIP Guide, Section 3.5 [29]). It provides information-level support of any ITS device with its ability to compile and decode any MIB. One can view decoded data at the bit, byte, or frame level using either a serial or Ethernet connection. It can analyze data in real time using the different alternate views or save data for later analysis. The only limit on the size of the capture data buffer is disk space, so it can capture and store large amounts of data for future analysis. In addition to decoding captured packets, FTS for NTCIP will also capture and display the status of RS-232 control signals in real time.

FTS for NTCIP runs on any Windows-compatible machine. The serial interface option requires installation of a hardware card and special cable for analysis, which are both included with the software. The parallel interface option requires a connection to an external parallel to serial converter. The parallel interface is ideally suited for laptop computers. The product supports data rates of up to 115.2 Kbps. Figure 11 and Figure 12 are screen images that show some of the different aspects and detailed analysis provided by the software.



Filter: All Frames

Summary Layer: SNMP

Frame Number	Community	Type	Req ...	Error	Object 1	Object 2	Objec
✓ 13	private2	Response	2	No Error	dmsMessageStatus.3.4		
✓ 14	private2	Set	3	No Error	dmsMessageMultiString.3.4	dmsMessageOwner.3.4	
✓ 15	private2	Response	3	No Error	dmsMessageMultiString.3.4	dmsMessageOwner.3.4	
✓ 16	private2	Set	4	No Error	dmsMessageRunTimePriority.3.4		
✓ 17	private2	Response	4	No Error	dmsMessageRunTimePriority.3.4		
✓ 18	private2	Set	5	No Error	dmsMessageStatus.3.4		
✓ 19	private2	Response	5	No Error	dmsMessageStatus.3.4		

```
Frame Number 16: (DTE) Length=88
⊞ Async PPP:
⊞ PPP:
⊞ IP: (Filter Layer)
⊞ UDP:
⊟ SNMP:
    ⊟ Header:
        Type: Sequence
        ⊞ Length: 53
        ⊞ Version:
        ⊟ Community Name:
            Type: Octet String
            ⊞ Length: 8
            Value: private2
    ⊟ Protocol Data Unit: Set Request
        Type: Set Request
        ⊞ Length: 38
        ⊟ Request ID:
            Type: Integer
            ⊟ Length: 1
                Length Indicator: Single Byte
                Length: 1
```

BINARY PANE

```
00100001 01000101 00000000 00000000 01010101
00100100 00000100 00000000 00000000 10000000
00010001 11110001 10100011 00001010 00010100
00001000 01100100 00001010 00010100 00001000
01100101 00000100 00001001 00000000 10100001
00000000 01000001 10100100 11010110 00110000
10000010 00000000 00110101 00000010 00000001
00000000 00000100 00001000 01110000 01110010
01101001 01110110 01100001 01110100 01100101
00110010 10100011 00100110 00000010 00000001
```

RADIX PANE

```
21 45 00 00 55 24 04 00 00 80 11 f1 a3 0a 14 08 64
0a 14 08 65 04 09 00 a1 00 41 a4 d6 30 82 00 35 02
01 00 04 08 70 72 69 76 61 74 65 32 a3 26 02 01 04
02 01 00 02 01 00 30 82 00 19 30 82 00 15 06 10 2b
06 01 04 01 89 36 04 02 03 05 08 01 08 03 04 02 01
64 98 d8
```

CHARACTER

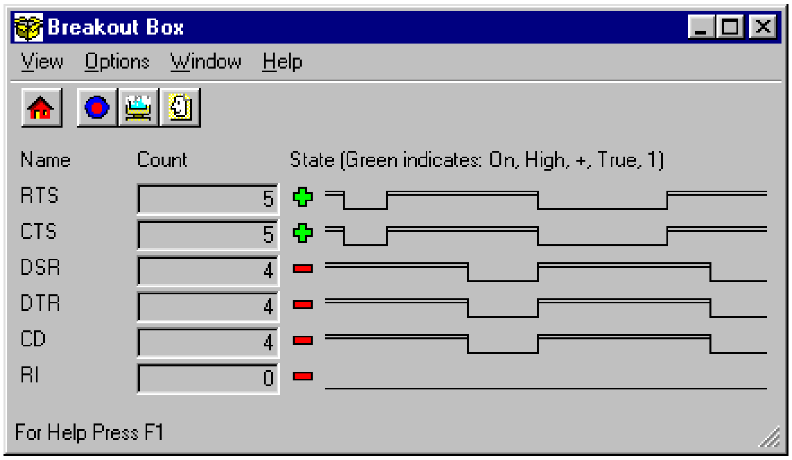
```
! E! ! U$%! ! C2ñf +5) d+5) e%*! d! AgČ0E! 5#"! %
) pr i vat e2f &#"%#"! #"! 0E! : 0E! 6' 1+' "%"L6
%#$&) ") $%#"d[ Ø
```

Event 1,519 of 168,016 (Frame 16) 08/28/2001 04:21:10.488 PM

Source	ASCII	Hex	Dec	Oct	Binary	RTS	CTS	DSR	DTR	CD	RI	Errors
DTE	80H+#	a3	163	243	10100011	Off	Off	Off	Off	Off	Off	

For Help Press F1

Figure 11. FTS for NTCIP Frame Display.



Name	Count	State (Green indicates: On, High, +, True, 1)
RTS	5	+
CTS	5	+
DSR	4	−
DTR	4	−
CD	4	−
RI	0	−

For Help Press F1

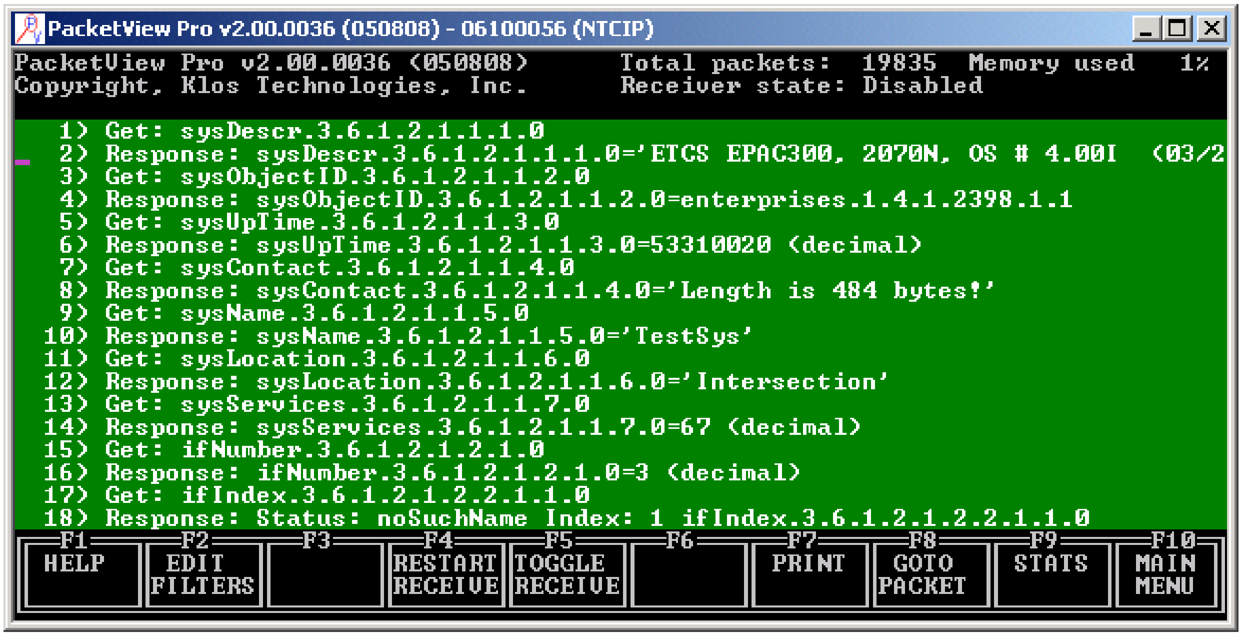
Figure 12. FTS for NTCIP Control Signal Status.

TrafficView

TrafficView is a protocol analyzer that decodes packets of the two protocols specific to NTCIP: PMPP and STMP. It is an option to PacketViewPro™[12], which is a Windows-based, general-purpose protocol analyzer for use with a local or wide area network (LAN or WAN). As such, it is supports both asynchronous (e.g., RS-232) and synchronous (e.g., Ethernet) interfaces. Another option to PacketViewPro is TimeStamper™[13]. TimeStamper is a Global Positioning System (GPS)-based time source for Windows that provides very accurate time. Time stamping of packets and this option make it ideally suited for making any timing measurements during performance testing.

TrafficView is able to handle data rates of up to 115.2 Kbps using standard serial ports and a "T" interface that Klos Technologies can provide. Although it still shows its roots in Disk Operating System (DOS), the software now runs under Windows 2000 or XP and is currently being updated for GUI. TrafficView can take advantage of that environment because the software can run on the same computer running another testing tool's software or the management application software. In this case, there is no worry about extra cables to connect or another computer. Figure 13 is a screen image of the packet capture display from PacketViewPro with the TrafficView option. Although not shown in Figure 13, scrolling left will show timestamp, type of packet indicator, address, and community name information. Figure 14 is a detailed view of an individual packet.

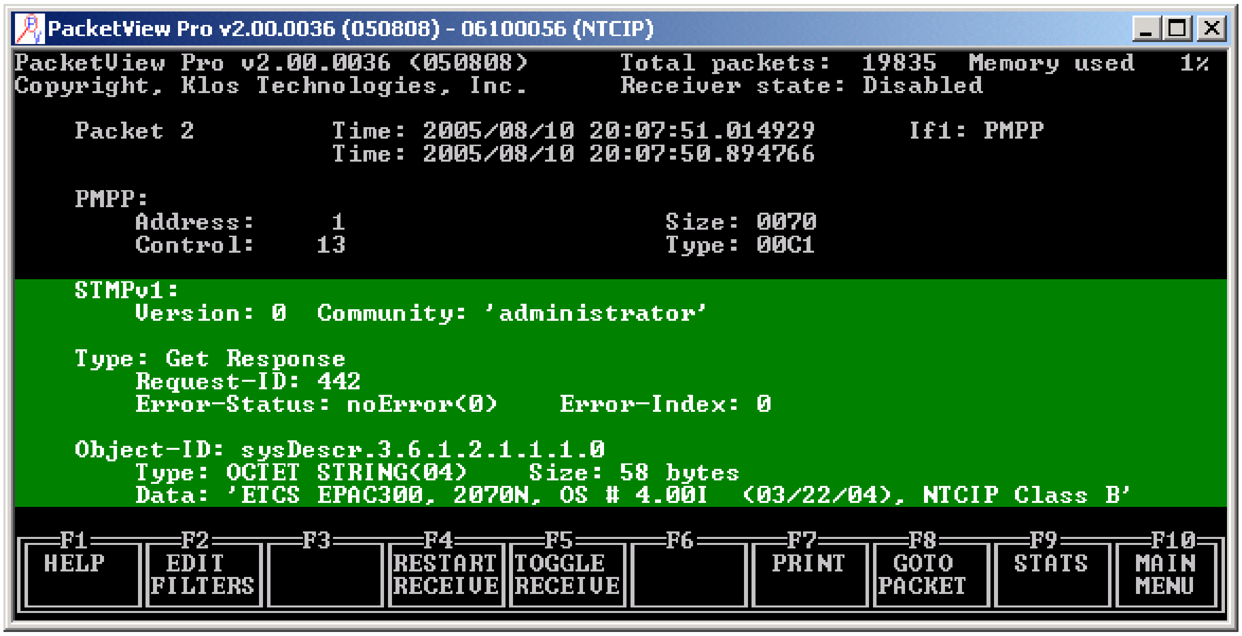
[12] PacketViewPro™ is a trademark of Klos Technologies, Incorporated, Cortland, New York.
[13] TimeStamper™ is a trademark of Klos Technologies, Incorporated, Cortland, New York.



```
PacketView Pro v2.00.0036 (050808)      Total packets:  19835  Memory used   1%
Copyright, Klos Technologies, Inc.      Receiver state: Disabled

   1) Get: sysDescr.3.6.1.2.1.1.1.0
   2) Response: sysDescr.3.6.1.2.1.1.1.0='ETCS EPAC300, 2070N, OS # 4.00I  (03/2
   3) Get: sysObjectID.3.6.1.2.1.1.2.0
   4) Response: sysObjectID.3.6.1.2.1.1.2.0=enterprises.1.4.1.2398.1.1
   5) Get: sysUpTime.3.6.1.2.1.1.3.0
   6) Response: sysUpTime.3.6.1.2.1.1.3.0=53310020 (decimal)
   7) Get: sysContact.3.6.1.2.1.1.4.0
   8) Response: sysContact.3.6.1.2.1.1.4.0='Length is 484 bytes!'
   9) Get: sysName.3.6.1.2.1.1.5.0
  10) Response: sysName.3.6.1.2.1.1.5.0='TestSys'
  11) Get: sysLocation.3.6.1.2.1.1.6.0
  12) Response: sysLocation.3.6.1.2.1.1.6.0='Intersection'
  13) Get: sysServices.3.6.1.2.1.1.7.0
  14) Response: sysServices.3.6.1.2.1.1.7.0=67 (decimal)
  15) Get: ifNumber.3.6.1.2.1.2.1.0
  16) Response: ifNumber.3.6.1.2.1.2.1.0=3 (decimal)
  17) Get: ifIndex.3.6.1.2.1.2.2.1.1.0
  18) Response: Status: noSuchName Index: 1 ifIndex.3.6.1.2.1.2.2.1.1.0
```

Figure 13. PacketViewPro Packet Capture Display.



```
PacketView Pro v2.00.0036 (050808)      Total packets:  19835  Memory used   1%
Copyright, Klos Technologies, Inc.      Receiver state: Disabled

    Packet 2          Time: 2005/08/10 20:07:51.014929      If1: PMPP
                      Time: 2005/08/10 20:07:50.894766

    PMPP:
        Address:      1                     Size: 0070
        Control:     13                     Type: 00C1

    STMPv1:
        Version: 0  Community: 'administrator'

    Type: Get Response
        Request-ID: 442
        Error-Status: noError(0)    Error-Index: 0

    Object-ID: sysDescr.3.6.1.2.1.1.1.0
        Type: OCTET STRING(04)    Size: 58 bytes
        Data: 'ETCS EPAC300, 2070N, OS # 4.00I  (03/22/04), NTCIP Class B'
```

Figure 14. PacketViewPro Detailed Packet Display.

Instrumentation Testing Tools Summary

Both FTS for NTCIP and TrafficView provide the same basic frame display capabilities of message exchanges over serial or Ethernet communication links. FTS for NTCIP makes more effective use of the Windows GUI interface. The one advantage of FTS for NTCIP is that it provides a control signal display. The one advantage of TrafficView is that cabling and an additional computer are not necessary. However, both products make the task of analyzing a message exchange extremely quick and easy. Most importantly, they independently monitor both ends of an exchange so that any fault can be isolated to its source. Table 3 provides a comparison chart of the two products.

Table 3. Instrumentation Testing Tool Comparison Chart.

		Instrumentation Testing Tool	
		FTS for NTCIP	TrafficView
Feature	Information-Level Support	Yes	Yes
	STMP Decode	Yes	Yes
	PMPP Decode	Yes	Yes
	Save to Memory	Yes	Yes
	General Protocol Decode	Yes	Yes
	Raw Byte Stream Data	Yes	Yes
	Signal Pin Display	Yes	No
	Runs in Windows	No	Yes
	Cost	$$$$	$$$$

CHAPTER 5: USER NEEDS

OVERVIEW

User needs should be the basis for any framework for a testing process. To establish a baseline of TxDOT's testing program, we prepared a questionnaire to understand what processes TxDOT already has in place; what various individuals would like to see added to these processes, especially in terms of NTCIP testing; and what other general information is available about testing. We surveyed individuals from the Austin, Amarillo, Corpus Christi, and Houston Districts, as well as the TxDOT Traffic Operations Division. In this chapter, we present the general responses to the questions and summarize some of the key points.

QUESTIONNAIRE AND RESPONSES

The survey contained questions in three areas: Initial Expectations, Current Testing Process, and Expanded Testing. The number of people surveyed consisted of two from the TxDOT Traffic Operations Division and one or two from each of the four district offices. What follows is the list of the questions contained in the survey and a summary of the responses.

Initial Expectations

1. What type of testing do you expect to be performed prior to delivery – burn-in (if so, how long), hardware testing, functional testing, etc.?

Summary of Responses – Division personnel expect that all TxDOT specifications are met. Although no specific paperwork is required, vendors should be able to show that each specific unit was tested. District personnel tend not to have any expectations from vendors; however, they expect that the units sent to the division will receive full environmental and functional testing.

2. Do you use any built-in hardware diagnostics utilities?

Summary of Responses – Internal diagnostics are not used except when a problem crops up.

3. What type of test documentation do you expect from a vendor – vendor test report, test lab report?

Summary of Responses – Except for environmental temperature testing reports, division personnel do not expect any reports to come with the equipment. District personnel do not expect anything but occasionally to see a manufacturer's cabinet checklist where an inspector has gone through a number of physical items related to required cabinet components, wiring, and connections. One district person stated that if he receives any paperwork dealing with test procedures, he sends it to the division.

4. Do you rely on Qualified Product Lists? Whose list? What equipment?

Summary of Responses – Everyone is aware that the TxDOT Qualified Products List (QPL) covers most equipment. There are some exceptions. One example is uninterruptible power supply systems.

Current Testing Process

1. What products do you test? How extensively? Do you use any specific test scenarios?

Summary of Responses – All interviewees reported that units are extensively tested; however, TxDOT typically does not follow formal procedures with specific scenarios when doing this testing. Most testing seems to be "seat-of-the-pants" testing and focuses around the specific application and the field implementation of the device.

2. What products are not tested?

Summary of Responses – Generally, TxDOT does not shop-test radios; instead, the procedure is to install radios in the field location and verify that they can perform their intended function.

3. How do you ensure that a device meets requirements/specifications?

Summary of Responses – Only two interviewees reported testing each device by going through the specifications step by step. The others reported relying on someone else within TxDOT (i.e., the Traffic Operations Division) or the supplier to verify that the equipment fulfilled all the required specifications.

4. Are all requirements/specifications tested?

Summary of Responses – Division personnel were rather emphatic that all requirements/specifications are tested. District personnel tended to focus on installation-specific requirements/specifications.

5. Do you follow a standard set of test procedures? If so, what are they?

Summary of Responses – Except for DMSs, TxDOT does not follow a standard set of procedures for testing devices. The requirements/specifications serve as a general guide in what is to be tested; however, TxDOT does not have formal criteria as to what constitutes compliance to the requirements/specifications.

6. When testing, are there any specific areas that you focus on?

Summary of Responses – Two interviewees mentioned that they focus on the visibility of DMSs and condensation in camera housings. Interviewees also mentioned the following:

- operation of video detectors in fog mode,
- voltage monitoring in the traffic signal controller,
- pedestrian detector inputs in the traffic signal controller,
- synchronization of clocks in the on-street master controller, and
- anything considered a “safety issue.”

7. If a device fails a test procedure, how is it retested – reinitializing the test (i.e., from the beginning of the test) or from the point of failure?

Summary of Responses – When a product fails, retesting generally occurs from the point of failure. Two districts reported that they would attempt to diagnose and repair devices. Two interviewees mentioned a “three failure” rule wherein three failures of any type will prompt a complete retest.

8. What percentage of your time do you spend in testing activities?

Summary of Responses – Although division personnel responses ranged from 5 to 25 percent, this figure may be more of a job responsibility issue. An estimate for the ITS Group was approximately 10 percent or less, while the Traffic Signal Group spends approximately 25 percent of the time conducting testing of controllers. District personnel reported 10-30 percent of time spent on testing.

9. What types of problems do you encounter when testing? What percentage of devices fail?

Summary of Responses – The general response to this question was that 5 percent or less exhibit any type of failure during testing. Cabinets appear to be major source of problems; however, estimates of the percentage of cabinets that failed TxDOT testing ranged from less than 1 percent to 10-15 percent. One interviewee reported that the NEMA TS 2 Specification lead to a dramatic decrease in problems.

10. What types of problems do you encounter after testing?

Summary of Responses – Two interviewees reported transient induced failures. “Lie mode” is another problem. For example, a DMS had the appropriate data entry, but there was no functionality associated with the data. The implementation appeared to support a function because there was a data entry for it. In reality, changing the data had no effect on operation. Other responses were human error and not applicable because maintenance is done by another group.

11. What percentage of devices experience hardware or software failures within 90 days of deployment?

Summary of Responses – All interviewees reported very low failure rates after initial turn-on.

12. What percentage of devices experience hardware or software failures within one year?

Summary of Responses – One-year failure rates were very low. The only problems cited were due to transients and reconfiguration.

13. Do you use different procedures during different phases of deployment (i.e., type acceptance, individual device testing, system acceptance, and after deployment testing)?

Summary of Responses – Type acceptance (essentially QPL) is extensive but loosely defined. After QPL, there are no formal procedures, but most major functions are checked. System acceptance is usually the providence of the districts. District system acceptance is thorough, but scenarios and tests are derived from specifications. In some cases, vendor test procedures are used. The Traffic Engineering Branch covers deployment testing, which samples sites in each of the 25 districts.

14. Do you have any testing sessions scheduled in the next two months? Would you permit someone to observe the testing session?

Summary of Responses – There were no immediate opportunities to observe a testing session.

Expanded Testing

1. Are there any specific problems that might be addressed by testing?

Summary of Responses – Two interviewees mentioned that it would be helpful if there was a test checklist or defined set of procedures. Another useful tool would be an automated input/output checker for traffic signal controllers.

2. Would you accept test procedures written by others including others state DOTs, FHWA, standards development groups, or other public or private entities?

Summary of Responses – As long as procedures meet with the group's acceptance, there would be no problem in accepting procedures written by someone else.

3. Would you accept third-party certification?

Summary of Responses – If certification were performed by a reputable organization, this would go a long way in ensuring correct and quality products. One respondent mentioned the concept of something like an Underwriters Laboratory seal of approval that shows a device has been tested against a set of requirements and has passed all tests.

4. Would a set of multi-tiered test procedures be beneficial? For example, would procedures that are broken down into spot check, major functional, and full functional tests be worthwhile?

Summary of Responses – All three levels need to be addressed. This would follow the general approach to testing that is already in place. One interviewee reported that there used to be a technical inspection that went through all controller functions but that it had to be discontinued because of lack of resources.

5. Given the opportunity, what additional equipment would you like to test?

Summary of Responses – There were numerous responses to this question:

- no additional equipment but rather training,
- automate the environmental test chamber,
- Malfunction Management Unit compatibility card tester/verifier,
- cabinet tester to check inputs and outputs (two respondents),
- standard set of videos of checking video imaging vehicle detection systems (VIVDS),
- automated functional checker for signal controllers,
- documented report on equipment, and
- spread spectrum radio test procedures.

QUESTIONNAIRE SUMMARY AND CONCLUSIONS

Given the responses to the questions related to failures, TxDOT's overall testing program appears to be doing an adequate job. The biggest problem is a lack of formal procedures and no procedures for checking NTCIP in some devices. If NTCIP compliance testing goes beyond just communications and addresses functionality, formal documented procedures will partially resolve the issue. The NEMA TS 2 Standard formalizes a set of electrical and environmental test procedures that apply to most ITS devices. The only additional procedures that seem to be lacking are some type of physical and/or mechanical checklists. One district has a checklist for traffic signal controllers, and it would be worthwhile to promote something similar throughout TxDOT.

There appears to be a need to define sets of test procedures for three phases or levels of acceptance. The phases correspond to the initial checks performed when adding a product to the

QPL, functionality tests of samples sent to the division for lot testing, and run-time or burn-in testing of products at the districts before installation. QPL testing would be the most intensive and perform very detailed testing. Sample testing would entail a moderate level of detail and check all the basic functions but only to a limited degree. Run-time testing would be fully automated procedures to exercise various inputs rather than just running a device with the power on.

None of the interviewees objected to the idea of using test procedures developed by outside parties (i.e., NTCIP or others). The test procedures, however, must be acceptable to TxDOT personnel and meet TxDOT's needs. NTCIP standards will eventually include test procedures. When a User Comment Draft of a standard with test procedures circulates for comment, TxDOT should make a specific point of reviewing it during the review period. If TxDOT feels that more stringent tests or other criteria should be added, their comments would likely be addressed in the standard.

A key requirement of any procedure developed for TxDOT should be automation. TxDOT personnel would like to do more testing, but time and resources are limited. Procedures should always consider how much time a "tester" spends in conducting a testing session. Any test reports should be easy to compile and understand.

Even though the transient protection components in a device or cabinet will never protect equipment 100 percent, performing transient tests at some phase or step in the approval process may prove beneficial. Performing these tests during environmental testing might put too much burden on the personnel performing those tests. Performing them during QPL testing or on an every-hundredth-unit basis, however, may not.

A simple input tester based upon an NTCIP testing tool could prove useful to district personnel when checking traffic signal controllers and introduce them to the benefits of NTCIP. Rather than simply letting a traffic signal controller run with the phases on recall, an input tester could randomly enter vehicle, pedestrian, and preemption calls and provide a visual display of the signal status. The NTCIP testing tool would serve as a software equivalent of a "suitcase tester." Running the tool would not require any external hardware. It would simply connect to the system port of a traffic signal controller that is running inside a cabinet. No changes to cabinet wiring would be required.

CHAPTER 6:
TXDOT TESTING FRAMEWORK

OVERVIEW

This chapter begins by describing the basic steps in testing for conformance to NTCIP. The next section deals with TxDOT specifications. It covers how the specifications define NTCIP requirements, describes the concept of and possible need for a TxDOT MIB, and discusses a special need to develop testing procedures for Texas Diamond phasing operation. The third and fourth sections address testing processes used by TxDOT and testing performed by TxDOT personnel and contractors. The fifth section deals with reporting test results. With different groups using different testing tools to generate testing reports, the format of those reports presents some issues. The concept of a requirements traceability matrix is also discussed. The next section of the chapter deals with the subject of regression testing, what the term means, and when it should apply. The chapter concludes with a discussion about a number of recommendations.

NTCIP TESTING

Testing for conformance to the NTCIP specifications consists of the following steps:

- checking to see if the all of the parameters, controls, and status information (objects) are present,
- checking to see if the parameters and controls can be set to the appropriate values, and
- checking to see if all status information is set properly.

The process of checking to make sure all the parameters, controls, and status information are present and accessible is accomplished by performing a “MIB Walk.” A MIB Walk involves stepping through every occurrence or instance of any object in a device to see if all parameter, control, and status objects are present in the device. Given a starting object identifier, the procedure is to send a “GetNext” command to a field device. The response to the command is the object identifier of the next supported parameter, control, or status object. Using that object identifier, the procedure sends another “GetNext” command to retrieve the next supported object. The process repeats until a “noSuchName” error indicates the end of the supported objects. If a field device returns a sequence of object identifiers for all the objects that are called for in a specification, then the device is compliant as far as object presence is concerned.

The process of checking to make sure all the parameters and controls can be set to the appropriate values is accomplished by trying to set an object to all possible values. If a specification states that an object is supposed to accept a value of 0 to 255, a Set command is sent to an object in the field device with a value of "0," followed by another Set command to set the object to the value of "1," and so on, up to 255. If a field device accepts all the values for all the objects without returning an error, then the device is conformant as far as supported values are concerned. Normal practice is to include some negative testing (i.e., trying to set a parameter to values outside the established range) during this process. Sending a value outside the acceptable range (e.g., –1 or 256) should return an error. An acceptable alternative to testing for all possible values is to test the lower and upper bound values (e.g., 0 and 255), a random sample of values between the two bounds (e.g., 2, 17, 43, 119, and 212), and a couple negative test values (e.g., –1 and 256).

These first two checks are simple to perform, and even a device with tens of thousands of parameters and controls takes a relatively short time to test. From a communications perspective, one might consider that NTCIP testing is complete because all objects are accessible and the parameters and controls can be set to the appropriate values. From a functional perspective, however, the question of whether the status information is correct remains. This step is the most time consuming and complex.

The process of checking to make sure all status information is set properly is accomplished by creating a condition or conditions wherein a status object (or a physical attribute) indicates the proper value under those conditions. For example, if the local time is equal to Greenwich Mean Time plus an offset to account for time zone and adjusted for daylight savings time, setting these objects to specific values should result in the local time object having the correct time. If a traffic signal controller is supposed to start the green of the coordinated phase an offset time after receiving a synchronization pulse, setting specific coordination parameters and then sending a synchronization pulse should result in the coordinated phase turning green at the offset time later.

The number of these condition tests is a function of the number of status objects implemented in a device and the number of possible values that each might have. A person can identify status objects in a MIB by looking at the ACCESS field of an object definition. All status objects are "read-only." A read-only entry in the ACCESS field means that a user can read the value of the object but the device determines the value. While the process of setting the conditions and waiting for something to happen is time consuming, automation can significantly reduce the amount of human involvement. However, there are cases where status is physical in nature. For example, a CCTV should physically move left when commanded to do so. A human tester must verify that the camera has actually moved.

SPECIFICATIONS

During October 2004, a search of the TxDOT Specifications section on the *TxDOT Expressway* website was conducted to identify specifications that contain a reference to NTCIP (*30*). The following describes how TxDOT specifications, special specifications, and special provisions define NTCIP requirements. Although many one-time use specifications reference NTCIP, this report focuses on statewide use specifications.

Traffic Signal Controllers

The TxDOT Material Specification – DMS-11170, Fully Actuated, Solid-State Traffic Signal Controller Assembly – defines NTCIP requirements as suggested in the NTCIP Guide (*31*, *32*). There is one concern, however. The NTCIP requirements primarily reference "NTCIP 1201:1996 and NTCIP 1202:1996." Both standards have undergone various WG updates, and future versions will be the ones that incorporate test procedures. Some elements of the test procedures may involve objects not found in the 1996 versions.

Closed Loop Systems/On-Street Masters

Although DMS-11170 mentions closed loop systems (CLS)/on-street masters, CLS masters do not have any NTCIP requirements stated within that specification (*31*). TxDOT Special Specification 6401 – Closed Loop Traffic Signal System does not have any NTCIP requirements either (*33*). TxDOT Departmental Specification TO-4055 – Closed Loop System On-Street Master Unit includes NTCIP requirements (*34*). The requirements, however, are essentially those of a traffic signal control. The current WG draft of NTCIP 1210-FMS does include a reference to a number of generic objects such as time of day and reports configuration

(18). It does not include phase-, preempt-, and ring-related objects. NTCIP 1210-FMS has more than 100 objects that express the unique functionality of an on-street master.

CLS Monitoring/Central Control Software

Although mentioned in DMS-11170, CLS monitoring/central control software does not have any NTCIP requirements stated within that specification. Likewise, Special Specification 6401 does not include any NTCIP requirements *(33)*.

Dynamic Message Signs

TxDOT Special Specification 6026 – NTCIP for Dynamic Message Signs defines NTCIP requirements in the prescribed manner *(35)*. There is one concern, however. The specification cites NTCIP 1101 – Simple Transportation Management Framework *(36)*. The NTCIP standards group does not plan to reaffirm NTCIP 1101. NTCIP 1102 – Octet Encoding Rules (OER), NTCIP 1103 – Transportation Management Protocol, and NTCIP 8004 – Structure and Identification of Management Information will replace it (*37*, *38*, *39*).

Spread Spectrum Radios

TxDOT Special Specification 6006 – Spread Spectrum Radio for Traffic Signals states, “Provide updates of the spread spectrum radio software free of charge during the warranty period, including the update to NTCIP compliancy” *(40)*. This statement is ambiguous as to what is specifically required. The specification should reference specific standards and list any object definitions that are required. (Even without an NTCIP standard specifically for spread spectrum radios, numerous manufacturers support SNMP, various Internet MIBs, and some of the underlying communications protocols referenced in NTCIP standards.)

Video Imaging Vehicle Detectors

TxDOT Special Specification 8970 – Video Imaging Vehicle Detection System states that, “The update of the VIVDS software to be NTCIP compliant shall be included” *(41)*. This statement is ambiguous as to what is specifically required. The specification should reference specific standards and list any object definitions that are required. NTCIP 1209 – Data Element Definitions for Transportation Sensor Systems is now a recommended standard *(42)*.

Other ITS Field Devices

There does not appear to any statewide TxDOT specification that includes a reference to NTCIP requirements for the following devices:

- closed circuit television pan/tilt/zoom controllers,
- data collection and monitoring devices,
- electrical and lighting management systems,
- environmental monitoring,
- highway advisory radio,
- ramp metering controllers,
- video switching equipment,
- non-video vehicle detection devices and systems, and
- distributed traffic control systems.

General

Two aspects of TxDOT special specifications and special provisions would add to the NTCIP testing framework and any type of testing. The format of the specification documents should use a numbering system so that requirements can be cross-referenced. Specification documents should also include a test case traceability matrix. The following requirements from a TxDOT specification and a test case traceability matrix illustrate the number style used in NTCIP documents and a matrix.

The basic articles and hierarchy of organizational elements as cited in the for Style Guide for Construction and Maintenance Specifications (Style Guide for the 2004 Specifications Book) is a alternative numbering scheme *(43)*. The following (without the numbered headings and items) is copied from DMS-11170. (Additional style changes may also be necessary to conform to TxDOT formatting guidelines.) Adding the reference numbers facilitates testing by ensuring that a requirement traces to a test procedure that checks the requirement.

---

5. Test and Acceptance

5.3 Burn-in each controller cabinet assembly for a period of 48 hr. at a temperature of 60°C or for a period of 96 hr. at a temperature of 23°C.

8. Hardware Design Requirements – NEMA Controller

8.1 Provide a controller unit that is completely solid state and digitally timed. All timing must be referenced to the 60-Hz power line.

10. Clock-Calendar Programming Requirements

10.1 Structure and Interrelationship of Programs

The structure and interrelationships of each type of program must be in accordance with the following paragraphs.

10.1.1 A day plan must consist of the following:

Hour : Minute	Action 1	(time to implement: action to implement)
:	:	:
Hour : Minute	Action 10	(time to implement: action to implement)

where each *action* is unique. There must be a minimum of *10 actions per day plan*.

10.1.2 There must be a minimum of *15-day plans*.

10.1.3 Each *action in a day plan* must consist of a group of the following *objects*:

- pattern, consisting of:
  - cycle length,
  - offset,
  - split,
  - MUTCD flash (on-off), and
  - free operation
- sequence,
- special functions 1–8 (on-off),
- auxiliary functions 1-3 (on-off),
- mode of operation (a means of changing operating modes by time of day),
- max II,
- gap / extension II, and
- phase omits.

Any or all of these may be selected within a single action.

10.1.4 Transfer into and out of FLASH must be in accordance with the Texas MUTCD. It must be possible to program each phase and overlap to flash either yellow or red via the front panel of the controller unit. This must be accomplished by flashing the load-switch driver outputs simultaneously.

10.1.5 An *entry* must consist of time period implemented: *day plan*, *months*, *dates* of the month, and *days* of the week.

10.1.6 A minimum of 255 *entries* must be programmable.

10.1.7 There must be a copy feature that allows the transfer of entries between day plans.

10.1.8 Other programming schemes that meet the functional intent are acceptable but require approval in writing by the TRF Signal Operations Engineer.

The following requirements-to-test-case traceability matrix (Table 4) now uses the requirement numbering scheme to cross-reference a test case or procedure to check it. The concept of a requirements-to-test-case traceability matrix comes from Institute of Electrical and Electronics Engineers (IEEE) Standard 1220 – IEEE Standard for Application and Management of the Systems Engineering Process (*44*). The matrix forces one to consider the wording of a requirement and whether it is testable. The matrix provides a synopsis of the requirement and a cross-reference to where the formal wording appears. Rather than restate the specifics of the requirement, the specification reference column provides a cross-reference to the wording. The requirement column is a short description of the requirement. The matrix links the requirement to the identifier and short description of the test procedure that confirms that an implementation meets the requirement. NTCIP 8007-TEST requires standards that include test procedures to incorporate a requirements-to-test-case traceability matrix (*15*).

Table 4. Requirements-to-Test-Case Traceability Matrix

TxDOT Requirements-to-Test-Case Traceability Matrix			
DMS-11170 Reference	Requirement	Test Case ID	Test Case Title
5.3	Manufacturer burn-in	N/A	**See Manufacturer Documentation**
10.1.1	Day plan structure	NTCIP 1201-TC004	Scheduler
10.1.2	Number of day plans		
10.1.3	Day plan action structure	NTCIP 1202-TC003	ASC Action Table
10.1.4	Transfer to and from flash	NTCIP 1202-TC013	MUTCD Flash
10.1.5	Time period entry	NTCIP 1201-TC004	Scheduler
10.1.6	Number of time period entries		
10.1.7	Day plan copy	N/A	**Manufacturer Specific**
10.1.8	Day plan functional equivalent	N/A	**Manufacturer Specific**

Texas MIBs

A MIB is a text file that describes a collection of parameters, controls, and status information associated with some type of function or entity. For use within NTCIP center-to-field communications, each parameter, control, and status is defined using a special notational format that indicates the name of the object, the data type (syntax) and possible range, access rights, a support indicator, its description, a possible default value, and a numerical identifier. For use in other domains, the format sometimes includes extra fields for additional information. The following is an example of an object definition from NTCIP 1201-GLO:

```
globalTime OBJECT-TYPE
    SYNTAX Counter
    ACCESS          read-write
    STATUS          mandatory
    DESCRIPTION
        "<Definition>The number of seconds since the epoch of 00:00:00 (midnight) January
        1, 1970 UTC (a.k.a. Zulu or GMT).
        <DescriptiveName>Controller.globalTime:quantity
        <DataConceptType>Data Element
        <Unit>second"
    DEFVAL          { 0 }
::= { globalTimeManagement 1}
```

The MIBs that appear in NTCIP standards are written for purposes of defining and showing conformance to the standard. In some respects, they are generic in that they address a type of device, not a specific instance of a device. They are not written to show compliance to a specification, which tends to be very specific about an implementation. Several TxDOT specifications, however, have indirectly created a TxDOT MIB to show compliance. By including a set of constraints such as those defined in the NTCIP compliance section of DMS-11170, the MIB is effectively changed to reflect a compliance perspective. For example, the NTCIP 1202-ASC MIB states the number of phases can vary from 0 to 255 (*4*). By adding a “Minimum Project Requirement” for 16 phases, a MIB that meets TxDOT specifications should state that the number of phases can vary from 16 to 255 and nothing less.

The distinction between conformance and compliance becomes important when one considers that a MIB is also used for testing purposes. Testers and testing tools use the information in a MIB to check valid ranges. If the range of maxPhases (number of phases in a traffic signal controller) is specified as 0 to 255, a value of 8 is valid. If a “TxDOT MIB” specified the range of maxPhases as 16 to 255, a value of 8 is not valid.

Range specifications also have an impact on how long it takes to execute a test procedure. If a parameter defines a user's text description and the length of the description is defined as SIZE (0..40) characters long, a test procedure might use that information to check whether an implementation can accept a description that is 0 characters long, 1 character long, 2 characters long, …and 40 characters long. If a specification states that the description has a range of SIZE (20..40) characters, the same test procedure would take half as long because anything less than 20 characters is not checked. While this example may seem trivial, some ITS devices contain tens of thousands of objects. An NTCIP-conformant traffic signal controller contains somewhere in the neighborhood of 60,000 objects. Performing several checks per second, a full-range check of all the objects might take several days to complete. Even if only the lower and upper bound values and several sample values were tested, the sheer number of object definitions has an impact on execution time.

Another aspect of a TxDOT MIB is that it should only contain those object definitions that are required. NTCIP device MIBs only include parameter, control, and status objects specifically related to the type of device. They include mandatory and option groups of objects and individual objects. A device MIB does not include all the object definitions related to an implementation. Other MIBs define generic or global objects and objects related to communications protocols.

An implementation actually uses several standard MIBs. In addition to those, a manufacturer may define proprietary objects. From a TxDOT perspective, the concern is to test for only those objects that are required. Any NTCIP-defined mandatory or optional objects or any manufacturer-specific objects that are not required should not be included in the MIB used for testing. For example, NTCIP 1201-GLO contains definitions for auxiliary inputs and outputs (*22*). A traffic signal controller does not need or use them. If the standard MIB is used for checking, a testing tool will check for their presence and will indicate the object is not supported even though it is not required. Other objects that are required may be labeled the same way. Using a standard MIB would require the person performing the test to determine if the non-support meant not required or whether the object was actually missing. Defining a MIB that includes only required objects would eliminate any confusion.

Texas Diamond

A TxDOT MIB might be useful to standardize the control and functionality of operations that are specific to TxDOT. NTCIP 1202-ASC defines the parameters, controls, and status information for multi-phase, multi-ring traffic signal controllers (*4*). It does not address any objects associated with Texas Diamond operation. Some manufacturers use proprietary objects to invoke this operation. Each one would require a unique procedure to set up and/or activate the operation. Because all traffic signal controllers provided to TxDOT are required to support the Texas Diamond operation, it may be worthwhile to define a set of common objects that everyone should implement.

It is unlikely that NTCIP 1202-ASC will ever include test procedures for checking the Texas Diamond operation (*4*). Because the Texas Diamond operation is essentially a state-machine driven by external vehicular actuations, a test procedure to check most, if not all, of the functionality could be defined. If such a test procedure is ever to be written, it would likely be up to TxDOT to develop it.

TXDOT TESTING PROCESS

The terminology used by testing domain experts to describe the different phases or stages of testing usually consists of the terms unit, integration, system, and acceptance. Rather than use those words, it may be more meaningful to relate testing to the activities that take place within TxDOT. For a bidder to provide equipment under a contract, the equipment must usually be on the TxDOT's Qualified Products List. To qualify for the list, equipment must pass QPL testing. Assuming that it is on the list, a bidder provides equipment that meets any contract-specific specifications or provisions to the district office. The district sends a sample of the equipment to division for environmental and, possibly, QPL testing. This round of QPL testing only takes place if the sample is not already on the QPL list. At the same time, the district checks and configures the equipment for its specific installation requirements. Once the equipment passes these checks, it is placed in the field and checked to see if it operates properly in the system.

Although this process is somewhat simplified, it describes the basic process. Figure 15 provides a label for TxDOT testing activities and illustrates the process a unit may go through before actual use in the field.

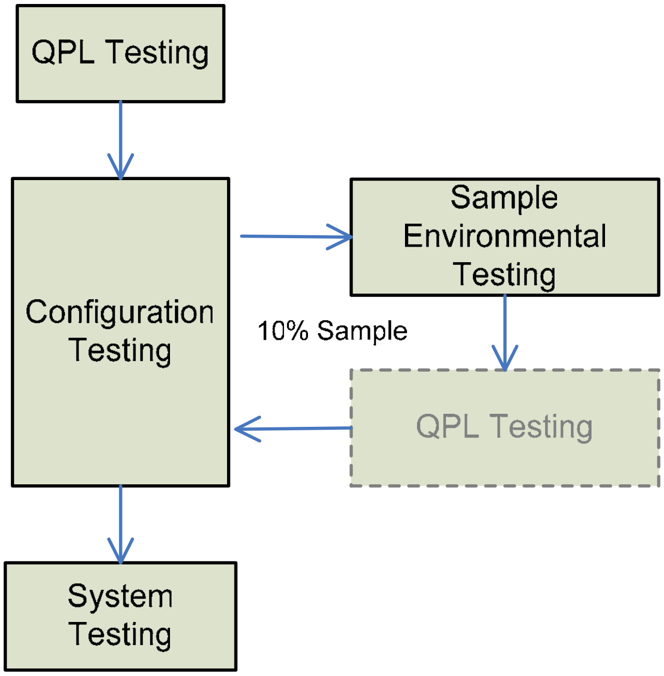


Figure 15. TxDOT Testing Activities.

QPL

The focus of QPL testing is compliance to TxDOT specifications for a specific device. Compliance to the specification entails a 100 percent check of all requirements. It is exhaustive and covers:

- hardware design,
- conformance to external standards,
- functionality, and
- documentation.

QPL testing differs from configuration testing in the level of detail. QPL testing uses the device's specification as a guide to ensure that equipment meets all of the requirements. Not only are all requirements checked, but a number of functional tests are also performed. The functional tests in QPL testing tend to cover more of the atypical uses and fault conditions. For example, during QPL testing, a test procedure for SNMP might test how equipment handles all data types and the various permutations. During configuration testing, a test procedure for SNMP might also test the typical data types but only the most likely permutations. As an example, the number nine can be encoded as 1, 2, or 4 bytes or use an indefinite-length form. This is equivalent to encoding the number nine as 9, 09, 0009, or 000000000009. QPL testing would look at all the permutations, whereas configuration testing would only look at the first two. Another example is how date and time might be tested. During configuration testing, a test procedure for checking date and time might involve setting a device's time to a value that falls under daylight savings time and verifying that any display reflects the correct time. QPL testing, on the other hand, would also look at what happens if a value too far in the past is sent to the device. It might also look at what happens if a value too far in the future is sent to the device.

Some of the active testing tools described in Chapter 4of this report can play an integral role in QPL testing and, at the same time, show compliance. Some NTCIP data dictionary standards go beyond defining just parameters and control objects in that they include a significant number of status objects. These status objects represent the outputs or state of a device. Some standards also define a standardized method for downloading a device's database. For an actuated signal controller, it is possible to download a database that defines a phasing sequence, the phase timings, and all the coordination timing data. It is then possible to invoke a particular timing plan and activate specific detector inputs. Because of the status objects, it is then possible to monitor the changing of green, yellow, and red signals to ensure that the controller is operating as expected. An active tester can perform this type of procedure completely unattended. An active testing tool can perform many of the tasks associated with QPL testing and show conformance to NTCIP standards at the same time.

QPL testing does not usually entail environmental testing. Rather, TxDOT asks a manufacturer to provide either a self-administered or a third-party environmental test report. TxDOT personnel review the report to ensure that the elements of the environmental test have been carried out. For QPL testing, a manufacturer's report addresses the following elements:

- operating voltage,
- power interruption,
- temperature and humidity,
- transients,
- vibration, and
- shock.

The test procedures used to carry out these environmental tests are derived from procedures defined in the NEMA TS 1 or TS 2 standard. Some ITS devices are not required to meet the temperature and/or voltage range as specified in the NEMA controller standard, but the steps involved would apply.

Configuration Testing

TxDOT district personnel carry out configuration testing in preparation for installation. The primary focus is on meeting project-specific requirements. Testing consists of going through each feature or functional requirement that relates to the project and checking to see if it operates properly. There are no specific details of these test procedures, so the exact nature and depth of testing is left up to the personnel conducting the test.

NTCIP standards can also play an integral part in configuration testing. A set of active test tool procedures could test the ability of equipment to accept a download of a database or to accept a full set of configuration parameters. Another set of procedures could activate various controls to simulate the various functional controls or inputs that the equipment would receive in normal field operation. If the appropriate status information is available, the same procedure could check the response of the equipment and thus automate the process. Even if the response to a control or input were physical in nature, a set of procedures would provide a consistent and repeatable method for checking that input.

An emulator testing tool would be applicable to configuration testing of ATMS central software. The one-time use Special Specification 1597 – Distributed Traffic Control System references NTCIP verification and testing (45). There is a requirement to:

> "*…develop a test plan that incorporates the use of a third-party testing suite and/or protocol analyzer to determine if a specific object is transmitted from and can be received by the central software. Require the test suite to determine the value that is being passed and be capable of testing the complete range of values…*"

The use of a protocol analyzer to verify these tests would be extremely labor intensive. An emulator testing tool with full scripting or programming capability would be much more suitable because it could be set up to simulate the behavior of field devices and emulate multiple instances of different types of field devices simultaneously.

Sample Environmental Testing

In the case of traffic signal controllers, TxDOT personnel perform sample environmental testing concurrently with configuration testing. Tex-1170-T, Sampling and Environmental Testing of Traffic Signal Controller and Assemblies and Auxiliary Equipment[14] spells out the procedures *(46)*. For NEMA TS 1 controllers, the procedures consist of:

- power line interruption and
- high voltage – high temperature:
    - interval timing and
    - power line interruption.

For NEMA TS 2 controllers, the procedures consist of:

- burn-in;
- low voltage – low temperature:
    - cold soak,
    - interval timing, and
    - power line interruption; and
- high voltage – high temperature:
    - interval timing and
    - power line interruption.

No other ITS device undergoes similar environmental testing unless there has been a specific request to do so. Special Specifications 6504 – Testing, Training, Documentation, and Warranty covers environmental testing and does apply to most devices *(47)*. However, a manufacturer or a third-party independent laboratory conducts the environmental testing procedures.

NTCIP does not appear to have a role in sample environmental testing. The interval timing tests could conceivably be automated using an active testing tool. However, the response

[14] Even though it is not stated in the current version of Tex-1170-T, TxDOT personnel have confirmed that all four variations of voltage and temperature are tested.

time to NTCIP messages is such that interval timing measurements may not be as precise as needed.

System Testing

During system testing, a device is connected to a management application either while it remains at a facility or after the device is installed in the field location and connected along with other components of the system. In either case, the management application defines what procedures are available and run.

The most relevant testing tool applicable to system testing is an instrumentation testing tool. The tool's primary use is in determining if a fault relates to the handling of the communications protocols. An instrumentation testing tool can determine if information is coded and packaged properly. This check applies to both the management application and the field device. It can isolate faults as to either source.

An active test tool running in an office environment can connect to any device installed in the field and simulate a number of the features and functions of any management application. Any procedure written for QPL or configuration testing could operate in that scenario. Any procedure to set the time of day, upload/download a database, check the cycling of a controller's phases, or position a CCTV could be executed by an active testing tool running in an office with the ITS field devices installed in the field.

CONTRACTOR TESTING PROCESS

Special Specification 6504 describes an alternate testing process that is the responsibility of a contractor to perform *(47)*. Numerous one-time use specifications reference this statewide use specification. Some one-time use specifications also include a reference to NTCIP. Figure 16 illustrates the major elements in "Contractor" testing.

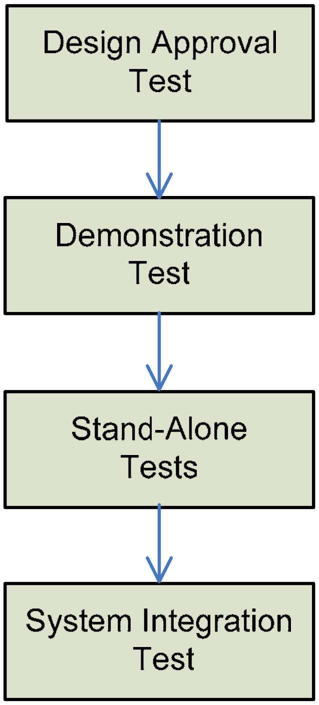


Figure 16. Contractor Testing Activities.

During design approval testing, the contractor either runs environmental tests directly or has an independent testing laboratory conduct them. During demonstration testing (conducted prior to installation), the contractor performs a physical inspection of the equipment and performs operational tests to ensure compliance to the specifications. After the contractor installs the equipment but before connection to any other components of the system, stand-alone testing looks at functional operations. After connection, system integration testing demonstrates that all control and monitor functions are operating properly.

Special Specification 6504 states that the contractor is responsible for conducting the tests *(47)*. TxDOT personnel do not perform the tests, but a reserve clause allows someone from TxDOT to observe the tests. The TxDOT engineer is responsible for overall approval and final acceptance. Although it is not specifically stated, the contractor develops the test plan, the general test procedures, and any procedures related to checking for conformance to NTCIP.

The researcher recommends that for consistency and ease of understanding, any test plan that involves NTCIP should have test procedures that use the NTCIP 8007-TEST format *(15)*.

TEST RESULTS

One issue that TxDOT may want to consider is the format of test results. Even though test procedures defined within NTCIP standards will be uniform and consistent, each testing tool will likely report results in its own specific manner. Contractor test reports will also differ from one to another. This situation leads to the case where reports from vendors, contractors, third-party laboratories, and TxDOT's own internal testing could be quite different in organization and appearance. Without some uniformity and consistency in results reports, they may be difficult to understand and make comparisons.

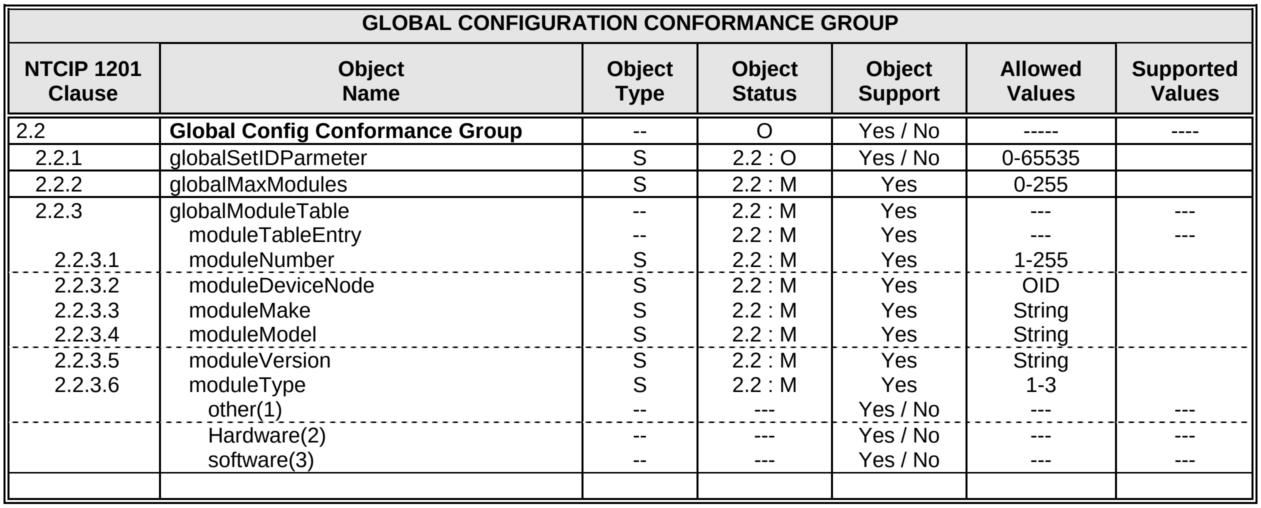
Because the format of test procedures defined by NTCIP 8007-TEST includes a column for results and space for adding notes, the researcher recommends that any results report use the same format as the procedures (*15*). Figure 17 is an example of how the test procedures could be used to report the results. The only difference between this report and the original table appearing in NTCIP 8007-TEST (see Figure 1 in Chapter 3) is that the results column now shows the actual outcome (highlighted) of the test procedure.

Test Case: **TC004**	*Title*: Description: Pass/Fail Criteria:	**Change Time with Administrator Community Name** The Test Case verifies that the administrator can change the time (i.e., one sample object) in the DUT. The DUT shall pass every verification step included within the Test Case in order to pass the Test Case.
Test Step Number	*Test Procedure*	*Results*
1	**Get** globalTime.0	Pass
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 are in seconds, this has the effect of setting the clock one hour ahead.	Pass
4	**Get** globalTime.0	Pass
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
6	**Set** globalTime.0 to the current time.	Pass
Test Case Result:		Pass
Test Case Notes:	It is not intended to check the accuracy of the clock, and it assumes that the DUT supports the *globalTime.0* object. At the end of this procedure, the clock may not be accurate, especially if the test fails.	

Figure 17. Test Procedure Results Format.

Reusing other portions of NTCIP standards would also help address the issue of uniformity and consistency in test results. Two common NTCIP test procedures involve checking support for a list of objects and the values that the objects may take. For example, a test procedure of an NTCIP-compliant traffic signal controller involves a check to see if the tens of thousands of objects called for in the requirements are actually implemented. Another test involves checking to see if the objects can be set to or indicate the appropriate values. Any test procedure would not likely list the individual objects but rather reference them. For the purposes of a test report, however, they do need to be itemized. For this purpose, any conformance group tables provided in an NTCIP standard's Profile (or Protocol) Requirements List (PRL) could be reused.

Most NTCIP Object Definition Standards include a checklist of the objects that may be required. These checklists are provided as conformance group tables in the PRL. The original intention of listing them was to serve as an aid in checking and formulating test procedures. Like the test procedures themselves, the conformance group tables can be reused to provide uniform and consistent test report results. Figure 18 is one of the conformance group definitions defined within the PRL of NTCIP 1209 – Data Element Definitions for Transportation Sensor Systems *(48*).



GLOBAL CONFIGURATION CONFORMANCE GROUP						
NTCIP 1201 Clause	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
2.2	**Global Config Conformance 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	---	---

Figure 18. NTCIP Conformance Group Definition.

In any test result report, the object support and supported values columns would contain the results of the test procedure. Figure 19 illustrates how it might appear in a report. The object support column contains actual responses, and the supported values column indicates the actual range values. Footnotes could explain any anomalies or add clarification. The highlighted areas in Figure 19 indicate the changes and additions.

GLOBAL CONFIGURATION CONFORMANCE GROUP						
NTCIP 1201 Clause	**Object Name**	**Object Type**	**Object Status**	**Object Support**	**Allowed Values**	**Supported Values**
2.2	**Global Config Conformance Group**	--	O	Yes	-----	----
2.2.1	globalSetIDParmeter	S	2.2 : O	Yes	0-65535	N/A [1]
2.2.2	globalMaxModules	S	2.2 : M	Yes	0-255	3
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	1-2 [2]
2.2.3.2	moduleDeviceNode	S	2.2 : M	Yes	OID	Eagle OID
2.2.3.3	moduleMake	S	2.2 : M	Yes	String	40 char
2.2.3.4	moduleModel	S	2.2 : M	Yes	String	40 char
2.2.3.5	moduleVersion	S	2.2 : M	Yes	String	40 char
2.2.3.6	moduleType	S	2.2 : M	Yes	1-3	2-3 [3]
	other(1)	--	---	No	---	---
	hardware(2)	--	---	Yes	---	---
	software(3)	--	---	Yes	---	---

Figure 19. Test Procedure Results Format.

Reusing the test procedures and the PRL section of a standard will add uniformity and consistency to any testing report, eliminating any need to interpret a specific testing tool's report, allowing a person to easily understand the results, and permitting easy comparison of test results performed by others.

Various standards groups include PRL as a method for concisely stating requirements and serving as an aid in testing. In the early days of the Internet, the Department of Defense developed PRLs to ensure interoperability. IEEE incorporates them in a number of their protocol standards. The purpose is to create a checklist of the requirements and provide an indication of whether an implementation meets each requirement. A vendor completes the checklist to show outside parties that he or she meets or does not meet the requirements. A tester can also use the PRL to indicate that he or she has tested the requirement and that a device either passes or fails the test. After a vendor or tester fills in the answers on meeting the requirements, the PRL becomes an Implementation Conformance Statement (ICS). An agency can use an ICS as indication as to whether an implementation meets the requirements of a particular project. A PRL developed for any type of testing would help TxDOT personnel understand requirements and serve as an aid in making sure everything that needs to be checked is checked.

The PRL approach could also be used for purposes of creating general-purpose checklists. PRLs are not limited to referencing NTCIP object definitions. For example, Figure 20 is part of the Physical Layer Conformance Group from NTCIP 2101 – Point to Multi-Point Protocol Using RS-232 Subnetwork Profile *(12)*. Even though the label of the second column is "Protocol Feature," the "features" are actually functional and physical requirements.

Item	Protocol Feature	Base Standard		Profile		Support
		Reference	Status	Clause	Status	
dataRate	Data Rate			2.3.2		
1200	1200 bps		-		m	Yes
Other DataRates	Higher Data Rates (indicate appropriate w/ checkmark) 2400 bps 4800 bps 9600 bps 19200 bps 38400 bps 56800 bps other rate (indicate bps)		-		o	Yes No ( ) ( ) ( ) ( ) ( ) ( ) _________ _
sync async	***Type of Data Communications:*** synchronous asynchronous	EIA/TIA 232-F			x m	No Yes
duplex	***Duplexing***					
halfDupl	Half Duplex	EIA/TIA 232-F	O.6	2.3.3	o.6	Yes
fullDupl	Full Duplex		O.6		o.6	Yes No
232 Circuits	***INTERCHANGE CIRCUITS FOR EIA/TIA 232-E***					
P1	Pin 1 – shield (earth ground)	EIA/TIA 232-F	M	2.3.1	o	Yes No
P2	Pin 2 – transmitted data (transmit data)	EIA/TIA 232-F	M		m	Yes
P3	Pin 3 – received data (receive data)	EIA/TIA 232-F	M		m	Yes

Figure 20. NTCIP 2101 Physical Layer Conformance Group

NTCIP standards use several different formats for PRLs. After looking at the standards and experimenting with different formats, the researcher believes that a modified requirements-to-test-case traceability matrix is the simplest to use. This would be similar to Table 4 on page 58 but include a column to indicate whether an implementation passed or failed the test cases.

REGRESSION TESTING

A testing framework must address the management of hardware and software. While changes in hardware designs are usually quite apparent and self-evident, software changes can be subtle but have significant impact. Regression testing is the term that describes the process of retesting software after some change. The intent of regression testing is to repeat all tests in order to ensure that a device does not fail a test that it previously passed. Regression testing takes place in the context of testing a new release or version of software (perhaps one that fixes a bug). Regression testing ensures that any "bug" fix did not introduce another bug.

Some transportation industry experts advocate full regression testing even after a minor correction. The researcher has seen cases where a seemingly minor change in one area has introduced a major flaw in another. Given enough resources and time, regression testing for even minor changes would be prudent. In the real world, it usually comes down to a judgment call, an understanding of the device, and the nature of the change.

For a manufacturer that uses a three-figure software revision numbering scheme (e.g., X.Y.Z), the rule of thumb should be that any first- or second-level change requires regression testing. A third-level rule might be linked to functionality of what was changed. For example, any change in a traffic signal controller's phasing or preemption area would warrant full regression testing. For a manufacturer that uses a two-figure software revision numbering scheme (e.g., X.Y), the rule of thumb should be that any first-level and every second-level change that includes something other than a trivial change requires regression testing.

RECOMMENDATIONS

Based upon what is currently known and available, the following outlines a number of recommendations. Bear in mind, however, that standardized test procedures are still evolving, new testing tools are coming onto the market, and testing tools are being continuously enhanced.

Consider using an active testing tool with a good scripting tool for QPL testing. This tool should have scripts that duplicate the procedures used in sample functional testing. The scripting capability would allow TxDOT to customize any procedure to its specific needs and develop new ones as needed. This would necessitate either relying on an outside third party or providing training on how to write scripts. However, scripts are quite easy to develop. Scripting languages are much simpler than the Basic programming language and easy to learn and understand. One should keep in mind that the major effort to add a procedure is in defining how to test something and not in actually writing a script to execute it. TxDOT personnel already involved in testing know how to define a test procedure. Someone with exposure to basic programming techniques could become proficient in developing scripts with a one-day training class and three or four days of experience writing them. Committing a test procedure to a script captures a tester's knowledge and automates the process so that it can be repeated many times without incurring additional effort. There is also an added benefit in that the script is sharable with anyone using the same testing tool.

For configuration testing, develop a "test box" that actively exercises a device. An active tester that has a library or ActiveX Control interface could be customized to download a database to a device and toggle its inputs. Instead of simply applying power and letting a device run, "test box" software could randomly activate inputs, change operating controls, and exercise a device the way it operates as if it were installed in the field.

For purposes of consistency and comparison, adopt a standardized format for test results. Reusing the NTCIP test procedures and Protocol/Profile Requirements Lists is a possible approach. Anyone familiar with the details of an NTCIP standard is already familiar with the format.

In the context of contractor testing where the specifications include references to NTCIP, modify Special Specification 6504 or create a Special Provision that addresses NTCIP requirements *(47)*. The NTCIP content could require the use of the Enterprise Test Procedures or that the contractor document procedures according to NTCIP 8007-TEST, and test results are reported using the NTCIP 8007-TEST and/or PRL format (*8*, *15*).

Adopt rules regarding when regression should take place. In TxDOT configuration testing and contractor demonstration testing, add a check of the software version number. If the version number indicates that regression testing is warranted, perform it.

CHAPTER 7:
TESTING FRAMEWORK RECOMMENDATIONS

The testing of conformance to NTCIP Standards ensures that ITS devices will be able to communicate and exchange information when part of an ITS. Conformance testing also ensures that users will be able to take advantage of the standardized protocols in their day-to-day tasks associated with the devices. It is recommended that conformance testing be a part of the QPL testing. A third party can perform the tests to ensure conformance. When TxDOT personnel perform QPL testing, checking conformance can be as simple as reviewing a test report. However, fully integrating conformance testing into TxDOT's current testing processes has several benefits. It provides TxDOT personnel with the confidence that tests were fully executed and streamlines compliance testing to TxDOT specifications. Capitalizing on the capabilities of NTCIP can enhance configuration testing and automate some the tasks associated with configuring and verifying operation intended for specific projects. Running some conformance testing procedures during system testing can help spot any problems before a system is fully connected.

A testing framework is not a single or additional step in the testing process but rather a number of steps leading to an overall quality testing program. Besides the steps involved in integrating NTCIP testing in the QPL compliance, configuration, and system testing phases carried out directly by TxDOT personnel, the framework should address the testing performed by contractors. Ensuring that contractors properly document their procedures and provide understandable results are also part of the framework.

To those ends, the following are the researcher's recommendations on defining a framework for the testing of conformance to NTCIP and integrating it into the current TxDOT testing program.

1. Add a detailed numbering scheme to any new specification, special specification, or special provision so that individual requirements can be cross-referenced.
2. Add something similar to a PRL to any new specification, special specification, or special provision to provide a summary of requirements.
3. Add a requirements-to-test-case traceability matrix to any new specification, special specification, or special provision so that all requirements have an associated test procedure.

4. Develop a test procedure for traffic signal controllers that implements the Texas Diamond phasing operations.
5. Create TxDOT MIBs that reflect TxDOT specifications.
6. Integrate NTCIP conformance testing into TxDOT QPL testing.
7. Develop/adopt a set of test procedures that extensively check conformance to NTCIP during QPL testing. This would include checks for support of all required objects, range/supported values, and scenario-based procedures to check all status objects.
8. Define objective criteria as to when a software change leads to retesting/regression testing of conformance to NTCIP as performed during QPL testing.
9. Use NTCIP PRLs and NTCIP 8007-TEST formatted test procedures as the method for reporting test results.
10. Integrate NTCIP conformance testing into TxDOT configuration testing.
11. Adopt a subset of QPL test procedures to check conformance to NTCIP and exercise the unit during configuration testing. This would include checks for support of all required objects, sampling of range/supported values, and scenario-based procedures to check relevant status objects.
12. Integrate NTCIP conformance testing in TxDOT systems testing.
13. Adopt a subset of QPL test procedures to check performance during system testing. This could include several scenario-based procedures to check the most relevant status objects.
14. Require contractors that supply NTCIP-conformant devices to use NTCIP PRLs and the NTCIP 8007-TEST format to document test procedures that are used during demonstration and stand-alone and system integration testing.
15. Require contractors that supply NTCIP-conformant devices use to an instrumentation testing tool during demonstration and stand-alone and system integration testing so that an engineer may easily verify that NTCIP protocols are being used.
16. Require contractors use NTCIP PRLs or similar requirements lists and NTCIP 8007-TEST formatted test procedures as the method for reporting test results.
17. Perform a cost/benefit analysis of supplying testing tools to division and district personnel.

REFERENCES

1. Setting Standards for the Future, Publication #FHWA-OP-00-009, U.S. Department of Transportation, Washington, D.C., 2002. http://www.standards.its.dot.gov/Documents/itsbro.pdf. Accessed March 8, 2005.

2. TS 1-1998 – Traffic Control Systems, National Electrical Manufacturers Association, Rosslyn, Virginia.

3. TS 2-2003 – Traffic Controller Assemblies with NTCIP Requirements, National Electrical Manufacturers Association, Rosslyn, Virginia.

4. NTCIP 1202 – Object Definitions for Actuated Signal Controller Units, A Joint Publication of AASHTO, Institute of Transportation Engineers (ITE), and NEMA. http://www.ntcip.org/library/standards/default.asp?documents=yes&qreport=no&standard=1202. Accessed July 15, 2005.

5. NEMA Releases Traffic-Messaging Equipment Standard for Dynamic Message Signs. In NEMA News, April 11, 2005. Published by NEMA. http://www.nema.org/media/pr/20050411a.cfm. Accessed July 25, 2005.

6. NTCIP 2301 – Simple Transportation Management Framework Application Profile, A Joint Publication of AASHTO, ITE, and NEMA. http://www.ntcip.org/library/standards/default.asp?documents=yes&qreport=no&standard=2301. Accessed July 25, 2005.

7. NTCIP Exerciser. http://www.ntcip.org/library/software/abstract.asp?AbstractID=66. Accessed August 5, 2005.

8. The Enterprise Pooled Fund. http://enterprise.prog.org/. Accessed July 25, 2005.

9. About the Coalition, Coalition Connection. http://www.i95coalition.org/about.html. Accessed July 25, 2005.

10. NTCIP 2201 – Transportation Transport Profiles, A Joint Publication of AASHTO, ITE, and NEMA. http://www.ntcip.org/library/standards/default.asp?documents=yes&qreport=no&standard=2201. Accessed July 25, 2005.

11. NTCIP 2202 – Internet (UDP/IP and TCP/IP) Transport Profile, A Joint Publication of AASHTO, ITE, and NEMA. http://www.ntcip.org/library/standards/default.asp?documents=yes&qreport=no&standard=2202. Accessed July 25, 2005.

12. NTCIP 2101 – Point to Multi-Point Protocol Using RS-232 Subnetwork Profile, A Joint Publication of AASHTO, ITE, and NEMA.

http://www.ntcip.org/library/standards/default.asp?documents=yes&qreport=no&standard=2101. Accessed July 25, 2005.

13. NTCIP 2103 – Point-to-Point Protocol over RS-232 Subnetwork Profile, A Joint Publication of AASHTO, ITE, and NEMA. http://www.ntcip.org/library/standards/default.asp?documents=yes&qreport=no&standard=2103. Accessed July 25, 2005.

14. NTCIP 2104 – Ethernet Subnetwork Profile, A Joint Publication of AASHTO, ITE, and NEMA. http://www.ntcip.org/library/standards/default.asp?documents=yes&qreport=no&standard=2104. Accessed July 25, 2005.

15 NTCIP 8007 – Testing and Conformity Assessment Documentation within NTCIP Standards Publications, A Joint Publication of AASHTO, ITE, and NEMA. http://www.ntcip.org/library/standards/default.asp?documents=yes&qreport=no&standard=8007. Accessed July 25, 2005.

16. ITS Standards Testing – A State of the Practice Report, NTCIP Testing Forum, http://forum.ntcip.org/upload/ITSTesting-SoP_Final_Forum.pdf. Accessed July 29, 2005.

17. Expanded Controller Interface Device I/O and Software Capabilities, National Institute for Advanced Transportation Technology. http://www.webs1.uidaho.edu/niatt/research/UTC_projects/year6/klk235.htm. Accessed July 25, 2005.

18. NTCIP 1210 – Field Management Stations - Part 1: Object Definitions for Signal System Masters, A Joint Publication of AASHTO, ITE, and NEMA. http://www.ntcip.org/library/standards/default.asphttp://www.ntcip.org/library/standards/default.asp?documents=yes&qreport= no&standard=2103. Accessed July 25, 2005.

19. Florida Department of Transportation – Traffic Engineering Research Laboratory (TERL) DMS Manufacturer Qualification Program, April 2003. http://www.dot.state.fl.us/trafficoperations/pdf/dms_qa_guidelines.pdf. Accessed July 26, 2005.

20. Langston, J. National Transportation Communication for ITS Protocol. Progress Report No. 6: April 2003-June 2003, pp. 13–18. Federal Aid Project CFDA-20.205/Contract No. BC352, Equipment Standards and Testing Development Research. http://rite.eng.fsu.edu/Progress_Reports/QUARTER_REPORT_2002_6.DOC. Accessed August 3, 2005.

21. Interoperability Test Lab, Texas Transportation Institute. http://translink.tamu.edu/facilities/itl.htm. Accessed July 25, 2005.

22. NTCIP 1201 – Global Object Definitions, A Joint Publication of AASHTO, ITE, and NEMA. http://www.ntcip.org/library/standards/default.asphttp://www.ntcip.org/library/standards/default.asp?documents=yes&qreport= no&standard=1201. Accessed July 15, 2005.

23. NTCIP 2100 – Class B Profile, A Joint Publication of AASHTO, ITE, and NEMA. http://www.ntcip.org/library/standards/default.asphttp://www.ntcip.org/library/standards/default.asp?documents=yes&qreport=no&standard=2100. Accessed July 15, 2005.

24. Maryland State Highway Administration Shares Its CHART II Software and Insight Newsletter of the ITS Cooperative Deployment Network. http://www.nawgits.com/icdn/chart2_cds.html. Accessed August 3, 2005.

25. NTester, Trevilon Corporation. http://www.trevilon.com/ntester_download.html. Accessed August 3, 2005.

26. NTCIP 1203 – Object Definitions for Dynamic Message Signs (DMS), A Joint Publication of AASHTO, ITE, and NEMA. http://www.ntcip.org/library/standards/default.asphttp://www.ntcip.org/library/standards/default.asp?documents=yes&qreport=no&standard=1203. Accessed August 4, 2005.

27. ITS Standards Testing – In-House Test Procedures and Scripts, Standards Testing Clearing House. http://www.itstestlab.org/h90w5691/index.htm. Accessed August 2, 2005.

28. Misra, A. “NTCIP Standard: A NTCIP DMS Driver,” ITS World Congress, Turin, Italy, November 2000.

29. The NTCIP Guide, NTCIP 9001, A Joint Publication of AASHTO, ITE, and NEMA. http://www.ntcip.org/library/standards/default.asphttp://www.ntcip.org/library/standards/default.asp?documents=yes&qreport=no&standard=2100. Accessed June 9, 2005.

30. Specifications, TxDOT Expressway. http://www.dot.state.tx.us/business/specifications.htm. Accessed July 29, 2005.

31. DMS-11170, Fully Actuated, Solid-State Traffic Signal Controller Assembly, Departmental Material Specifications 7-115, Section 19. Published by TxDOT. http://manuals.dot.state.tx.us/dynaweb/colmates/dms/@ebt-link;?target=idmatch(s070019). Accessed July 29, 2005.

32. NTCIP 9001 – NTCIP Guide, A Joint Publication of AASHTO, ITE, and NEMA. http://www.ntcip.org/library/standards/default.asphttp://www.ntcip.org/library/standards/default.asp?documents=yes&qreport=no&standard=9001. Accessed August 2, 2005.

33. Special Specification 6401 – Closed Loop Traffic Signal System. Published by TxDOT. ftp://ftp.dot.state.tx.us/pub/txdot-info/cmd/cserve/specs/1993/spec/es6401.pdf. Accessed July 25, 2005.

34. Departmental Specification TO-4055 – Closed Loop System On-Street Master Unit. Published by TxDOT. ftp://ftp.dot.state.tx.us/pub/txdot-info/gsd/pdf/to4055.pdf. Accessed July 25, 2005.

35. Special Specification 6026 – NTCIP for Dynamic Message Signs. Published by TxDOT. ftp://ftp.dot.state.tx.us/pub/txdot-info/cmd/cserve/specs/2004/spec/ss6026.pdf. Accessed July 25, 2005.

36. NTCIP 1101 – Simple Transportation Management Framework, A Joint Publication of AASHTO, ITE, and NEMA. http://www.ntcip.org/library/standards/default.asp?documents=yes&qreport=no&standard=1101. Accessed July 29, 2005.

37. NTCIP 1102 – Octet Encoding Rules (OER), A Joint Publication of AASHTO, ITE, and NEMA. http://www.ntcip.org/library/standards/default.asp?documents=yes&qreport=no&standard=1102. Accessed July 29, 2005.

38. NTCIP 1103 – Transportation Management Protocol, A Joint Publication of AASHTO, ITE, and NEMA. http://www.ntcip.org/library/standards/default.asp?documents=yes&qreport=no&standard=1103. Accessed July 29, 2005.

39. NTCIP 8004 – Structure and Identification of Management Information, A Joint Publication of AASHTO, ITE, and NEMA. http://www.ntcip.org/library/standards/default.asp?documents=yes&qreport=no&standard=8004. Accessed July 29, 2005.

40. Special Specification 6006 – Spread Spectrum Radio for Traffic Signals. Published by TxDOT. ftp://ftp.dot.state.tx.us/pub/txdot-info/cmd/cserve/specs/2004/spec/ss6006.pdf. Accessed July 25, 2005.

41. Special Specification 8970 – Video Imaging Vehicle Detection System. Published by TxDOT. ftp://ftp.dot.state.tx.us/pub/txdot-info/cmd/cserve/specs/1993/spec/es8970.pdf. Accessed July 29, 2005.

42. NTCIP 1209 – Data Element Definitions for Transportation Sensor Systems, A Joint Publication of AASHTO, ITE, and NEMA. ttp://www.ntcip.org/library/standards/default.asp?documents=yes&qreport=no&standard=1209. Accessed July 29, 2005.

43. Style Guide for Construction and Maintenance Specifications. Published by TxDOT. ftp://ftp.dot.state.tx.us/pub/txdot-info/des/specs/styleguide.pdf. Accessed June 30, 2005.

44. IEEE 1220 – IEEE Standard for Application and Management of the Systems Engineering Process. Published by Institute of Electrical and Electronics Engineers, Inc., New York, 1998.

45. Special Specification 1597 – Distributed Traffic Control System. Published by TxDOT. ftp://ftp.dot.state.tx.us/pub/txdot-info/cmd/cserve/specs/1993/spec/es1597.pdf. Accessed July 29, 2005.

46. Tex-1170-T, Sampling and Environmental Testing of Traffic Signal Controller Assemblies: Traffic Signal Controllers and Conflict Monitors. Published by TxDOT. http://manuals.dot.state.tx.us/dynaweb/colmates/tsi/@ebt-link;?target=idmatch(s080001). Accessed July 29, 2005.

47. Special Specifications 6504 – Testing, Training, Documentation, and Warranty. Published by TxDOT. ftp://ftp.dot.state.tx.us/pub/txdot-info/cmd/cserve/specs/1993/spec/es6504.pdf. Accessed June 30, 2005.

48. NTCIP 1209 – Data Element Definitions for Transportation Sensor Systems, A Joint Publication of AASHTO, ITE, and NEMA. ttp://www.ntcip.org/library/standards/default.asp?documents=yes&qreport=no&standard=1209. Accessed July 29, 2005.