

Module 52—Applying Your Test Plan to the Electrical and Lighting Management Systems Based on the
NTCIP 1213 ELMS Standard v03

Ken Leonard: ITS standards can make your life easier. Your procurements will go more smoothly and you'll encourage competition, but only if you know how to write them into your specifications and test them. This module is one in a series that covers practical applications for acquiring and testing standards-based ITS systems.

Ken Leonard: I'm Ken Leonard, the Director of the U.S. Department of Transportation's Intelligent Transportation Systems Joint Program Office. Welcome to our ITS Standards Training Program. We're pleased to be working with our partner, the Institute of Transportation Engineers, to deliver this approach to training that combines web-based modules with instructor interaction to bring the latest in ITS learning to busy professionals like yourself. This combined approach allows interested professionals to schedule training at your convenience without the need to travel. After you complete this training, we hope you'll tell your colleagues and customers about the latest ITS standards and encourage them to take advantage of these training modules, as well as archived webinars. ITS Standards training is one of the first offerings of our updated Professional Capacity Training Program. Through the PCB Program, we prepare professionals to adopt proven and emerging ITS technologies that will make surface transportation safer, smarter, and greener. You can find information on additional modules and training programs on our website at www.pcb.its.dot.gov. Please help us make even more improvements to our training modules through the evaluation process. We look forward to hearing your comments and thank you again for participating and we hope you find this module helpful.

James Frazer: Welcome to course T306, Applying Your Test Plan to the Electrical and Lighting Management Systems Based on the NTCIP 1213 ELMS Standard v03.

James Frazer: I'm your instructor today. I possess an in-depth knowledge of commercial and industrial sales and marketing processes, as well as that of control system integration, networks, and protocols. I've proposed, engineered, and managed large projects throughout the U.S. and Europe. I have a degree in Mechanical Engineering and a Master's degree in Business Administration.

James Frazer: Our learning objectives for today include describing ELMS testing, describing the ELMS test plan application, identifying the relevant elements of an ELMS test plan, and finally, describing the adaptation of a test plan to a project-specific application.

James Frazer: So let's delve into our first objective, describing ELMS testing.

James Frazer: The testing life cycle, the role of test plans, and the testing to be undertaken for electrical and lighting management systems is our first focus, including why we test ELMS; the purpose of an ELMS test plan; components of an ELMS test plan, including test design specification, test case specification, test procedure specification.

James Frazer: To confirm that an ELMS will work as intended, the testing process provides objective evidence that the system satisfies the system requirements: does, in fact, solve the right problem and satisfies the user needs, as defined by the stakeholder community.

James Frazer: Continuing on why we test, let's delve a little bit further into testing and the systems life cycle. We've seen this slide in courses A306a and A306b—the prerequisites to T306—but let's look at it in a little more detail. The system's lifecycle Vee diagram that shows testing to be undertaken at various

stages, as well as the types of testing to be utilized. Testing is done on the right hand side of the Vee model, and it includes unit device testing, subsystem verification, system verification and deployment, and system validation.

James Frazer: We use this to confirm that an ELMS system will work as intended. Just as a review of the Vee model: remember that we start using the Vee model with a facilitation and extraction of User Needs from the stakeholder community. These are refined into measurable, functional System Requirements. Later, we move on—on the right hand side of the Vee model—to System Verification and lastly, System Validation. Notice that Unit/Device Testing is dependent upon Detailed Design. Similarly, Subsystem Verification is dependent on High Level Design. System Verification and Deployment is dependent upon System Requirements. And lastly, System Validation is dependent on the basic Concept of Operations. It's important when looking at this diagram to focus upon the traceability of testing to the requirements and the foundational user needs. The system life cycle diagram, finally, shows testing to be undertaken at various stages of the development process, which includes test planning, testing documentation—meaning the test plan—and the entire testing process. Note that test planning is done earlier than the test design documents. The gold arrows define the traceability connection between the tests and the user needs and the requirements. The purpose of a test plan is meant to answer the question, "Does the system conform to the requirements?" This test plan is a document describing the scope—basically the technical management of the process—the technical approach, the resources needed, and the schedule or the time frame to complete the process. The test plan identifies the items to be tested, the features to be tested, the specific testing tasks, and the risks—requiring a contingency plan.

James Frazer: It's also very important to remember that testing determines whether the system conforms to the requirements and whether it satisfies its intended use and user needs according to IEEE-829-2008. This standard—also known as the 829 standard for software and system test documentation—is an IEEE standard that specifies the form of a set of documents for use in eight defined stages of software testing and system testing. Each stage potentially produces its own separate type of document. The standard specifies the format of these documents but does not stipulate whether they must all be produced, nor does it include any criteria regarding adequate content for each document. That's a matter of judgment for domain experts. The documents within 829 are a Master Test Plan, a Level Test Plan, a Level Test Design, a Level Test Case, Test Procedures, Test Logs, and Anomaly Reports. We'll delve into this in more detail in future slides.

James Frazer: A test plan provides the description of the overall approach to testing all of the requirements to be verified. The test plan outlines the scope, the approach, the resources, and the schedule of testing activities. Notice the Test Plan Specification is related to the Master Test Plan. The Test Design Specification is one level down in granularity and supports the Test Design Unit Test, the Test Design Integration Test, Test Design System Acceptance, and Test Design Periodic Maintenance. The individual Test Case Specifications are one level lower in that hierarchy and are dependent upon the tests above. And lastly, the Test Procedures define the processes for the test case. Additional information on this can be found in course T321, which is a prerequisite in the testing curriculum.

Module 52—Applying Your Test Plan to the Electrical and Lighting Management Systems Based on the
NTCIP 1213 ELMS Standard v03

James Frazer: Test Plan Specification. Test plan specifications detail objectives, target markets, the internal beta team, and processes for a specific test for a software or hardware product. Test plan specifications contain a detailed understanding of the comprehensive work flow.

James Frazer: Test Design Specification. A test design breaks apart testing into smaller efforts and describes a test design specification—the specification that outlines the requirements to be tested—and identifies which test cases cover which requirements. It's important here to remember the hierarchical structure that we reviewed in a previous slide of the test plan to the test design specification to test case and test procedures that is required to develop a fully systems-engineered test plan.

James Frazer: A test case is a set of conditions under which a tester will determine whether the system is working as it was originally intended to do. It identifies and specifies the inputs and the outcomes and conditions for execution of a test, and is included in a document called the Test Case Specification as part of the overall test plan. It identifies a specific input and/or output that needs to be tested and records the purpose of the test, a description of the test, the input and output test specification, the environmental needs, and it references the test procedure and describes the results of the test.

James Frazer: The Test Procedure Specification defines a process that produces the test result. It is a technical operation that consists of determining the characteristics of a given product, process, service according to a specified procedure.

James Frazer: With that, it's time for an activity.

James Frazer: Please answer the following question: Which is not a component of an ELMS test plan? Your answer choices are A) Test facilitation; B) Test design specification; C) Test case specification; or D) Test procedure specification. Please select the correct answer.

James Frazer: Let's review our answers. The correct answer is A—test facilitation is not part of an ELMS test plan. B is incorrect because test design specification is in fact part of an ELMS test plan. C is incorrect because the test case specification is also part of an ELMS test plan. And lastly, D is incorrect because the test procedure specification is in fact part of an ELMS test plan.

James Frazer: With that, we have concluded Learning Objective number 1—describing ELMS testing. Our next objective is describing the ELMS Test Plan application.

James Frazer: Describing the ELMS Test Plan Application.

James Frazer: The steps in developing an ELMS Test Plan include identifying requirements to be tested and those not to be tested for each testing phase; identifying the test methodology and the approach; introducing and describing the requirements to a test case traceability matrix—the RTCTM; planning the logistics of testing; estimating the level of effort and time for testing; evaluating testing risks; and defining a comprehensive project plan closeout.

James Frazer: Developing a Sample Test Plan: Identifying the Requirements to Test. The requirements are found in the Protocol Requirements List—the PRL—in the NTCIP 1213 v03 standard. For those who would like a little additional information and some background, Module A306b—the prerequisite to this course—identifies how to define ELMS requirements for your project-specific implementation. You also

can examine the Student Supplement for the PRL and a list of requirements. When you're developing your test plan, it's important to remember that every requirement should be tested during at least one test phase using at least one method and by at least one party. The range and extent of agency testing is a risk management issue. It's important to have a substantive conversation about how broad will be the testing program.

James Frazer: Identifying a Test Plan Level. Each test level will have its own test plan. This includes the prototype phase, during design approval, during factory acceptance, when the device is incoming to the site, site acceptance testing, burn-in testing—and there may be others. It's often further divided into NTCIP communication compliance testing, hardware testing, and environmental testing, for example. You can examine more of this in the supplement as well.

James Frazer: As we develop that sample test plan approach, it's important to identify the test methodology, inspections, do a complete analysis, demonstrate the testing, and then proceed with your formal testing. It's important to not only consider but implement testing scenarios both for positive tests, negative tests, as well as testing around boundary levels. It's important to remember that there is a need to test to the standard as well, beyond the functional requirements of the project. Just because the project doesn't require a feature does not mean that the feature will not be needed in the future. If you do have an ELMS system, a later integrator software may need that particular feature and expect that your system supports it, particularly if it's mandatory. If the product says it meets that standard, it should meet all mandatory requirements of the standard, whether they are deployed or not.

James Frazer: This is a subset of the Requirements to Test Case Traceability Matrix. Notice the one-to-one relationship of test cases to requirements. In our first row, notice that the Test Case ID 3.5.4.1.1—Retrieve Luminaire Pole Identifier—is directly traced back to a Requirement ID—3.5.4.1.1.1. Also notice in the bottom two rows, that the requirement Configure Luminaire Color Temperature supports two test cases—Configure the Luminaire Color Temperature and Incorrectly Configure Luminaire Color Temperature. The lesson to be learned there is that multiple test cases may trace back to one requirement, as we see in the bottom two rows.

James Frazer: Identifying and defining the test environment is important. It needs to be defined in terms of hardware and software required—both the test application, data analysis, and the device under test.

James Frazer: To plan the logistics of testing you need to define and determine where will tests be performed; safety during on-site testing—particularly if there's electrical or mechanical dangers—and exactly who is responsible for what: for the power, for tools, for the actual physical tables to set up the test equipment, protection from the elements, as well as local assistance if testing is being done at a remote site. It's also important to define what actually happens if testing is suspended: if something doesn't go according to plan—or even if there's simply a power interruption.

James Frazer: Developing a sample test plan also includes estimating the effort, the schedule, and the budget for preparing the test plan, preparing the test cases, preparing the test procedures, and performing multiple rounds of testing—including performing the actual tests, investigating any problems that the tests have uncovered or revealed, and documenting the results of both the tests, the problems, and their resolution.

Module 52—Applying Your Test Plan to the Electrical and Lighting Management Systems Based on the
NTCIP 1213 ELMS Standard v03

James Frazer: As you develop a sample test plan, it's important to understand the impact of a failure. Failures allow direct tracing back from a test case to a requirement and the underlying user need in order to determine corrective action. This slide shows the first step in that trace back process from a test case back to a requirement.

James Frazer: Continuing on failure analysis, this slide shows the second step in that trace back process from a requirement back to a user need.

James Frazer: Finishing on developing your sample test plan. Project plan closeout is important. It's important to have a plan and to understand the impacts of accepting a failure. You may not be able to address every failure, but it is important to accept the impacts if you are unable to rectify and address it.

James Frazer: With that, it's time for another activity.

James Frazer: Please answer the following question. Which of the following ELMS statements is false? Your answer choices are: A) Every ELMS requirement should be tested; B) You should only need to perform your ELMS test plan once; C) Some ELMS testing may be performed by the manufacturer; or D) ELMS traceability tables can help you assess the impact of a test failure. Please select the correct answer.

James Frazer: Let's review our answers. B is the correct answer. The statement is not true. Testing will often reveal problems. These should be fixed and the device retested. A is a true statement: every requirement should be tested. C is also true: testing may be performed by the agency, the manufacturer, or a third party test lab, for example. And D is true: traceability tables allow you to identify the user needs that will not be completely fulfilled.

James Frazer: With that, we have concluded Learning Objective 2: describing the ELMS test plan application. We will move on to Learning Objective 3: identifying the relevant elements of an ELMS test plan.

James Frazer: Identifying Relevant Elements of an ELMS Test Plan.

James Frazer: Remember that only project-specific requirements are tested. This includes all the mandatory objects and any optional objects that might be selected for your project-specific implementation. In this case, Control Electrical Service is optional, but we've selected "Yes." And similarly, we've selected others and deselected some other options. While testing needs to cover just the project requirements, it is important during design to generate project requirements that meet future needs as well. For instance, if the project requirements include control by central photocell and the standard product support control by time of day, you should also include the time of day in your requirements and tests.

James Frazer: Designing Test Case Specifications and Procedures. It's important to review guidance from IEEE-829-2008 that was introduced previously, as well as NTCIP 8007. The principles described here are defined by both IEEE-829 and NTCIP 8007. We no longer use 8007 as originally targeted but now go by the 829-2008 formats and derive test procedures from those sources. In that way, we can use guidance from 8007, but not directly. Agencies should directly use 829-2008 as guidance. It's important to remember that the final step in this process is the development and review of the results of the testing—

this information is known as the Test Report. We review the guidance from 829. We take some insight from 8007. We apply the guidance to our sample dialogs. There are two key differences between 829 and 8007. The IEEE standard approach is applicable to all ITS standards, including Center-to-Center and Center-to-Field standards. The IEEE standard approach separates the test cases from test procedures, while previous efforts combined both. The IEEE standard allows reuse of test procedures where agencies typically place more efforts. And lastly, the IEEE approach includes a test plan and a method to split testing into test designs and includes test reports. So use IEEE-829-2008, but you can also refresh yourself on the subject matter of 8007.

James Frazer: More specifically, what exactly does IEEE-829-2008 provide? It provides a test plan, test design specification guidance, and guidance on test case specifications and test procedure specifications as well as on test reports—including test logs, test anomalies, and test reports. Testing professionals across the ITS community should be familiar with these definitions and formats. The 829 standard for software and system test documentation is an IEEE standard that specifies a form of a set of documents for use in eight defined stages of software testing and system testing, each stage potentially producing its own separate type of document. The standard specifies the format of the documents but does not stipulate whether they must all be produced, nor does it include the criteria regarding adequate content for these documents.

James Frazer: NTCIP 8007 describes components and how they can be combined for NTCIP testing—including the test case identifier, the purpose, the inputs, particular pass-fail criteria, procedural steps, expected outputs, and features to be tested. It defines the terms that can be used in test steps for NTCIP testing. NTCIP 8007 is a process control and information management series document. It defines the content requirements to be used by other NTCIP working groups when they produce NTCIP test documentation. NTCIP 8007 is intended to promote a consistent look and feel of NTCIP testing documentation throughout the various NTCIP standards. You can think about it informally: NTCIP 8007 as a dictionary of terms of syntax guidance, whereas 829 focuses on formatting test case specifications, procedures, and reports.

James Frazer: This slide is an example of a simple dialog between a Management Station and an ELMS device. A comprehensive set of dialogs is included in the standard.

James Frazer: Designing Test Case Specifications and Procedures, continued. First, you will specify your test case—basically, what you are testing. Then you would specify the test procedure—how you run the test. After you've completed initial steps in the test planning process, you're ready to move on to specifying test cases for each requirement. Once that process is complete, you can then specify the test procedure.

James Frazer: Specifying Each Test Case. This slide is an example of the requirement-to-test-case relationship as well as a detailed test case description. Focus upon the four elements inherent in this example—the Test Case Name, the Description, the Variable, and the Pass/Fail Criteria. Since IEEE-829 is being used, it is useful to make sure that the student understands the differences here. 829 asks for an identifier, an objective, inputs, outputs, environmental needs, special procedural requirements, and inter-

case dependencies. Thus, students need to be able to understand the differences and what may be the same.

James Frazer: Designing Test Case Procedures. Data exchanges should follow defined dialogs, as defined in the standard. Generally speaking, the test case procedures should return the device to its original state. Verification steps should cite the relevant requirement. Typically, test cases do in fact test multiple requirements. We reviewed NTCIP 8007 and how it precisely defines standardized step types. For example, a SET operation includes nine specific verification checks related to the simple network management protocol response packet. All of these data exchanges should follow dialogs as described unambiguously in the standard. Additionally, each of these should cite the related relevant requirements. Remember also that for the management system—if it already exists—existing dialogs should be tested as well.

James Frazer: Steps of a Sample Procedure. As we introduced in the previous slide, step number 1 in our test procedure is configure—determining the number of seconds to advance a clock in the ELMS system. Step 2 is to get—or retrieve—the global time, where the results can be pass or fail. We can get the time or not. Step 3 is recording the response value for global time as start_Time. Step 4 is setting a global time, with the result being a pass or fail. Step 5 is delaying for 15 seconds. Step 6 is getting the following object—globalTime.0—which again has a pass or fail result. Step 7 is verifying that the response value for the selected object is equal to the original time we retrieved plus 15 seconds and that total being equal to the second retrieved object. We have four Pass/Fail steps here and we should successfully negotiate each one of those with a Pass.

James Frazer: The process of adapting the test plan based on selected user needs and requirements is what we will focus upon next. We already have described the components of a test plan. We have examined the major components of test cases and test procedures. Next, let's create a project-specific ELMS test plan.

James Frazer: And it's time for an additional activity.

James Frazer: Please answer the following question. Where can you find definitions for terms that can be used in NTCIP test steps? Your answer choices are: A) IEEE-829; B) NTCIP 8007; C) ISO 9001; or D) The Student Supplement to this course. Please select the correct answer.

James Frazer: Let's review our answers. B is correct. As we previously discussed, NTCIP 8007 defines terms that can be used in test steps for NTCIP testing. IEEE-829 is incorrect: it explicitly defines sample outlines for test documentation, but does not define explicitly the steps for NTCIP, nor does it define the terms. C, ISO 9000, is incorrect: ISO 9000 deals with quality management but does not deal directly with NTCIP testing. And D is incorrect: the Student Supplement provides samples of test procedures but it does not define the test terms themselves.

James Frazer: With that, we have concluded our discussion of Learning Objective 3—identifying the relevant elements of an ELMS test plan. And we'll move on to Learning Objective 4—describing the adaptation of a test plan.

James Frazer: Describing the Adaptation of a Test Plan.

Module 52—Applying Your Test Plan to the Electrical and Lighting Management Systems Based on the NTCIP 1213 ELMS Standard v03

James Frazer: The information you'll be using in creating an NTCIP 1213 v03 test design specification is the NTCIP 1213 v03 National Transportation Communications for ITS Protocol Object Definitions for Electrical and Lighting Management Systems standard. Specific portions of that standard that you'll be using include the Protocol Requirements List—the PRL—and the Requirements Traceability Matrix—the RTM. Once again, testing documentation is not currently included with this standard; thus, you'll need to create this on your own. The standard is 1213 v03. The PRL, the RTM are resident within the standard. The testing documentation must be created.

James Frazer: As a little bit of a background about the NTCIP 1213 standard, it's a center-to-field communications standard—meaning Traffic Management Center to devices out physically in the field on the roadside. It contains systems engineering content—meaning the standard has a PRL and an RTM. It has traceability between the measurable functional requirements and the stakeholder-driven user needs. However, ELMS does not contain test procedures in this current version.

James Frazer: The Protocol Requirements List. It contains user needs; it contains functional requirements; it describes the relationship between the needs and the requirements. Project-specific requirements are identified by the project-level Protocol Requirements List that's project-specific to your application.

James Frazer: Similarly, the Requirements Traceability Matrix contains measurable functional requirements. It contains object dialogs as well. And there is a relationship—traceability—between the object dialogs and the requirements. A project-specific Requirements Traceability Matrix references relevant design content needed to define the inputs and outputs for the test case specification.

James Frazer: This is an example of the electric service's SwitchState dialog. Each dialog includes a numeric Object Identifier ID as well as an Object Name. Each project-specific object will need to be examined.

James Frazer: To create an NTCIP 1213 v03 Test Design Specification, we begin with the PRL—the Protocol Requirements List—by selecting project-specific functional requirements. We do this by specifically circling a Yes or a No in the Support column. For a review in how to understand ELMS user needs, please review prerequisite course A306a—Understanding User Needs for the NTCIP 1213 Standard. Also for a review on how to select project-specific functional requirements, please examine course A306b—Specifying Requirements for Electrical and Lighting Management Systems based on NTCIP 1213 v03.

James Frazer: Using the project-specific requirements we selected in the PRL, we next move to the RTM. In the RTM, we trace those functional requirements to the specific objects. Notice the dependency in our image on the screen between the Object ID and its foundational Functional Requirement.

James Frazer: Next, we develop test case objectives for each requirement. In this example, we have defined one objective addressing the four requirements listed there on the screen. Notice that the test case verifies that the data value of the objects requested are within specific ranges. To verify system interface implements for a positive test case, we examine four objects: 3.5.5.2.2 and its three successive objects—electricalserviceSwitchMode, SwitchModeTime, SwitchState, and PhotocellIndex. The test case

Module 52—Applying Your Test Plan to the Electrical and Lighting Management Systems Based on the
NTCIP 1213 ELMS Standard v03

verifies that the data value of the objects requested are within specified ranges. The object identifier—known as an OID—of each object requested is the only input required. An output specification is provided to show valid value constraints, as per the NTCIP 1205 v01 object definitions.

James Frazer: Next we move to the test case output specification used to specify input OIDs and outputs. We identify value constraints, data type, and valid value ranges.

James Frazer: Next we identify dialogs, inputs, and outputs. Notice how Type and Value Range are defined here as OCTET STRING and a range of 1 to 9.

James Frazer: This is another definition from the standard. Notice how Type and Valid Range are defined here as INTEGER and a range of 0 to 65535.

James Frazer: Step 5 is documenting the value constraints for inputs. From the standard, we enter the value constraints into the Test Case Input Specification. Notice those on the right in the red box.

James Frazer: We also enter the value constraints into the Test Case Output Specification. At this point, it's important to remember that negative testing is also important. In the example above, if 1 to 9 is valid, what happens when a 0 is transmitted? How about 10?

James Frazer: In this slide—Step 6, Completing the Test Case—we are completing a simplified test case. Notice the additional items of Environmental Needs, Tester Reviewer, Special Procedure Requirements, and Inter-Case Dependencies. Environmental needs may be defined in terms of temperature or humidity and the Tester/Reviewer name should be entered. Any special procedure requirements should be noted. Any special inter-case dependencies can also be entered into this test case.

James Frazer: Supporting Objects Not in the Standard. Extending the standard complicates interoperability interchangeability. It's often not achievable, unless all design details are known. Additionally, extensions are relatively custom solutions resulting in increased specification costs, increased development costs, increased testing costs, and costs of integration. It often results in much longer deployment timeframes and can lead to increased maintenance costing as well. They're not recommended unless absolutely necessary and it requires some substantive thought before you pursue these extensions.

James Frazer: Extensions should only be considered when NTCIP features have been determined to be inadequate to meet the needs of the stakeholder community, and that the benefits of extension outweigh the very substantial added costs of adding an extension.

James Frazer: Extended equipment should be designed to appropriately integrate with NTCIP-only deployments and should be designed to minimize any added complexity.

James Frazer: If you do choose to test objects that are not in the standard, please adhere to the relationships between the PRL, the RTM, and the RTCTM—as well as the underlying user needs and measurable functional requirements. The main purpose of test design is to identify the features to be tested by a particular level test—such as a unit test. The features to be tested are included, as we

examined previously in the RTCTM, which is in itself based upon the Requirements Traceability Matrix. This is a very important fact to remember.

James Frazer: Next we'll introduce the Test Procedure Generator, including: What is the TPG? Why is the TPG important? What are the benefits? How do you use the TPG? How does it fit into the testing for NTCIP standards? And where does a user obtain the TPG? The TPG tool is used to automate the test planning process—specifically in the areas of test cases and test procedures. TPG reports describe breakages between user needs, requirements, and design details. If extensions to the standard have been added for the project, the test procedures generated by the TPG can be used to help determine compliance to a project specification.

James Frazer: What is the TPG and how does it work? The U.S. DOT has released version 2.1 of the Test Procedure Generator Tool for use by the NTCIP ITS standards communities. TPG version 2.1 supports development and deployment of NTCIP Center-to-Field device interface standards with systems engineering content. TPG is a Windows-based software tool that uses Microsoft Word to input the NTCIP standards and output test procedures automating that process. TPG supports ITS standards developers as well as deployers: local and state agencies of NTCIP Center-to-Field standards.

James Frazer: For deployers and local agencies, the TPG guides the development of test procedures by loading and processing the standard to be implemented—including the requirements, the dialogs, and the objects. It bases the test procedures on the user-selected requirements in the particular NTCIP Center-to-Field standard—whether it be 1213 or some of the other 1200 series standards.

James Frazer: It uses a standardized and consistent language for test procedures development, including standard keywords, variables, and object names imported directly from the standard that's being used. It does output an XML file that could be consistently interpreted by vendors and testing staff in their test suites. Standards deployers can also use the TPG to create consistent test procedures. A very important fact to remember is that the TPG itself is not a testing tool. It's a test procedure generator that assists you in developing testing processes and procedures—but is not a testing tool itself.

James Frazer: Benefits of the TPG. Agencies can use the TPG to develop consistent test procedures for verifying conformance and compliance. Using the TPG tool also will reduce developmental risks, effort, and the cost of developing standards and test procedures.

James Frazer: As described previously, the TPG tool automates Steps 3 and 4—Test Cases and Test Procedures—of the test planning process. It does not address the Test Plan, Test Design Specifications, Test Execution, or Test Reports.

James Frazer: The role of TPG in testing is that it supports off-the-shelf interoperability. It promotes the systems engineering process by giving users support in creating test procedures. Standardized and easily available test procedures are conformant to the standard and help to eliminate proprietary system elements that might work their way into the system.

James Frazer: Let's take a brief look at the TPG. The first step in using the TPG is to start a new session by loading the NTCIP 1213 v03 standard in Word doc 2010 format into the TPG software.

Module 52—Applying Your Test Plan to the Electrical and Lighting Management Systems Based on the
NTCIP 1213 ELMS Standard v03

James Frazer: This slide represents the graphical user interface of the TPG tool. A tree view—the session panel—is on the left. The session status is in a ribbon at the bottom.

James Frazer: The next step—Step 2—is to create a new test procedure.

James Frazer: Creating a new test procedure begins with defining a test procedure title and description. You notice that the first box in the image is the Test Procedure Title. The second is Test Procedure Description. And the third is Test Procedure Pass or Fail.

James Frazer: Step 3 is to select your requirements by checking the check boxes of project-specific requirements. For this, you would go back to your PRL and decide what optional objects are required for your project-specific implementation.

James Frazer: This slide represents the test procedure as defined so far in our process. Notice the requirements are listed in the Requirement(s) row.

James Frazer: Step 4 is to select your variable objects.

James Frazer: Step 5 is to create your test procedure step.

James Frazer: This slide represents your project-specific test procedure results. Notice in our image, test step number 1 includes CONFIGURE the procedure “determine the enumerated value for the sign type required by the specification (PRL 2.3.2.1 and 2.3.2.3)” as well as subsequent test steps and the results.

James Frazer: That's a brief overview of the TPG. I urge you all to experiment with it. How to obtain it? TPG version 2.1 updates include compatibility with Windows 7 Professional as well as compatibility with Microsoft Office 2010. For more information on how to acquire the TPG, please visit the web site that you see, <https://www.standards.its.dot.gov/DeploymentResources/Tools>. The free download package includes the TPG version 2.1 installation file and the TPG user manual. For more information you can email TPGSupport@noblis.org.

James Frazer: It's time for another activity.

James Frazer: Please answer the following question. Which of the following statements is false? A) TPG version 2.1 supports development and deployment of NTCIP Center-to-Field device interface standards with systems engineering content; B) TPG is a testing tool; C) TPG is a Windows-based software tool that uses Microsoft Word to input the NTCIP standards and output test procedures; or D) TPG supports ITS Standard developers as well as deployers (local and state agencies) of NTCIP Center-to-Field standards. Please select the correct answer.

James Frazer: Let's review our answers. B is the correct answer because it is false: TPG is not a testing tool. A is incorrect because TPG does support development and deployment of NTCIP Center-to-Field device interfaces. C is incorrect because TPG is a Windows-based software tool. D is incorrect because TPG supports ITS Standard developers as well as deployers.

Module 52—Applying Your Test Plan to the Electrical and Lighting Management Systems Based on the
NTCIP 1213 ELMS Standard v03

James Frazer: With that, we have concluded our four learning objectives of describing ELMS testing; of describing an ELMS test plan application; of identifying relevant elements of an ELMS test plan; and describing the adaptation of a test plan.

James Frazer: We have now completed the ELMS curriculum of Module A306b—Specifying Requirements for Electrical and Lighting Management Systems Based on NTCIP 1213 v03. And this course, Module T306—Applying Your Test Plan to the Electrical and Lighting Management Systems Standard Based on NTCIP 1213 v03.

James Frazer: Thank you very much for completing this module. Please use the feedback link below to provide us with your thoughts and comments about the value of this training. Thank you very much for attending.