

User Manual for a Center-To-Center (C2C) Reference Implementation

Contract No. DTFH61-12-D-00051

www.its.dot.gov/index.htm

Final — June 2021



U.S. Department of Transportation

Produced by FHWA Office of Operations Support Contract DTFH61-06-D-00004.
U.S. Department of Transportation
Federal Highway Administration

Notice

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof. The U.S. Government is not endorsing any manufacturers, products, or services cited herein and any trade name that may appear in the work has been included only because it is essential to the contents of the work.

Quality Assurance Statement

The Federal Highway Administration (FHWA) provides high-quality information to serve Government, industry, and the public in a manner that promotes public understanding. Standards and policies are used to ensure and maximize the quality, objectivity, utility, and integrity of its information. FHWA periodically reviews quality issues and adjusts its programs and processes to ensure continuous quality improvement.

<u>Technical Report Documentation Page</u>			
1. Report No. FHWA-JPO-XX-XXX	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle User Manual for a Center-To-Center (C2C) Reference Implementation		5. Report Date June 2021	
6. Performing Organization Code			
7. Author(s) Walter Crear, Robert Rausch, Elizabeth Williams	8. Performing Organization Report No.		
9. Performing Organization Name and Address TransCore ITS 192 Technology Parkway, Suite 500 Norcross, GA 30092 Under Contract to Cambridge Systematics, Inc.	10. Work Unit No. (TRAIS) DTFH61-06-D-00051		
12. Sponsoring Agency Name and Address U.S. Department of Transportation ITS Joint Program Office-HOIT 1200 New Jersey Avenue, SE Washington, DC 20590	13. Type of Report and Period Covered 2010-2021		
14. Sponsoring Agency Code JPO			
15. Supplementary Notes			
16. Abstract The Center-to-Center (C2C) Reference Implementation (RI) User Manual provides guidance and instructions on the use of the C2C RI software application. This manual is intended for an experienced systems professional, with a basic background with Intelligent Transportation Systems (ITS) standards, who wants to use the C2C RI application software to perform conformance testing of a system interface which implements the supported ITS C2C Standards. The C2C RI is a tool for verifying a target system's conformance to designated Intelligent Transportation System (ITS) C2C standards. This user manual provides the background information, concepts, and instructions a user needs to utilize the C2C RI in the verification process. The information is organized in accordance with the IEEE 1063-2001 Standard for Software User Documentation using the task-oriented instructional mode of presentation. The User Manual provides guidance and reference material regarding: <ul style="list-style-type: none"> • Conformance Testing for Selected Standards; • Conformance Test Definition and Setup; • Conformance Test Execution; • Post Test Report Generation; and • RI Tool Options and File Maintenance. 			
17. Key Word Center-to-Center, Reference Implementation, User Manual, software, standards, ITS, IEEE, conformance, testing, verification, documentation, guidance.	18. Distribution Statement No Restrictions.		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 129	22. Price N/A

Table of Contents

1.0 Introduction.....	1
1.1 Identification	1
1.2 Purpose.....	1
1.3 Intended Audience	2
1.4 Scope	2
1.5 Software Purpose	2
1.6 Document Conventions.....	3
2.0 References	5
3.0 Concept of Operations (ConOps).....	6
3.1 Conformance Testing Concepts.....	6
3.2 Conformance Testing with the C2C RI.....	8
4.0 Procedures.....	10
4.1 Installing the C2C RI Application.....	10
4.2 Preparing the C2C RI Test Environment.....	12
4.3 Starting the C2C RI Application.....	13
5.0 Navigating within the RI	14
5.1 RI Operating Modes.....	14
5.2 User Interface Screen	14
5.3 Menu Navigation	15
5.4 Accelerator Key Navigation.....	24
5.5 Guided User Mode	25
6.0 Define a Test Configuration	28
7.0 Execute a Test Configuration	37
8.0 Report Test Results.....	42
8.1 How to Navigate the C2C RI Report Viewer Window.....	42
8.2 How to Generate Test Configuration Reports	42
8.3 How to Generate Test Log Reports.....	52
8.4 How to Export Log Reports to CSV	69
8.5 Report Anomaly	70
9.0 Customizing the C2C RI and Performing Support Functions	72
9.1 Maintaining Configuration and Log Files.....	72
9.2 Setting Software Options	72
9.3 Creating Authorized Custom Test Suites	73
9.4 Creating User Defined Test Suites	75
9.5 Modifying Test Case Data	76
9.6 Test Suite Creation	100
10.0 Error Messages and Problem Resolution.....	109
10.1 Overwriting a Test Log File.....	109
10.2 Cancelling Changes to Test Configuration File	109

10.3	Closing a Test Configuration File	110
10.4	Error Message Displayed When Insufficient Need or Requirement Selected	110
10.5	Saving changes to a Test Configuration File.....	111
10.6	Using the 32-bit Version of the C2C RI Application.....	111
10.7	Deleting Files from the C2C RI Application.....	112
11.0	Glossary	114

List of Tables

Table 1-1. Document Conventions.....	3
Table 4-1. RI System Requirements	10
Table 5-1. Shortcut Keys for Application Usage.....	25

List of Figures

Figure 3-1. C2C RI Test System Architecture	8
Figure 4-1. C2C RI Setup Screen	11
Figure 4-2. C2C RI Setup Default Installation Directory	11
Figure 4-3. Choosing Start Menu Folder Screen	12
Figure 5-1. RI User Interface Window: Main.....	15
Figure 5-2. New Test Configuration Dialog	17
Figure 5-3. Open Dialog	18
Figure 5-4. RI Test Definition Dialog.....	19
Figure 5-5. Reports Window.....	20
Figure 5-6. Maintenance Dialog	21
Figure 5-7. RI Parameters Options Dialog – General Tab.....	22
Figure 5-8: RI Parameters Options Dialog – HTTPClientSettings Tab.....	23
Figure 5-9. Help About Message.....	24
Figure 5-10. Wizard Create/Edit Test Configuration Dialog Initial Screen.....	26
Figure 5-11: Wizard Progress Panel	27
Figure 6-1. Test Definition: Configuration Panel.....	29
Figure 6-2. Test Definition: System Under Test (SUT) Panel.....	30
Figure 6-3. Test Definition: Information Layer Parameters Panel.....	31
Figure 6-4. Test Definition: Application Layer Parameters Panel	32
Figure 6-5. Test Definition: Information Layer Test Cases Panel.....	33
Figure 6-6. Test Definition: Application Layer Test Cases Panel	34
Figure 6-7: Test Definition: Entity Emulation Parameters Panel.....	35
Figure 6-8: Load Entity Data Dialog	35
Figure 6-9: Entity Data Viewer.....	36
Figure 7-1. RI Test Execution Window: Pending Execution	38
Figure 7-2. RI Test Execution Window: Results.....	39
Figure 7-3: User Content Verification Window	40
Figure 7-4. RI Test Execution Window: Log.....	41
Figure 8-1. C2C RI Report Viewer Toolbar	42
Figure 8-2. Report Configuration Window.....	43
Figure 8-3. Test Configuration Report with Completed Inputs.....	44
Figure 8-4. Report Request Progress Window	44
Figure 8-5. Test Configuration Report Completion	45
Figure 8-6. C2C RI Test Configuration Report	46
Figure 8-7. C2C RI Test Configuration Report Details.....	47
Figure 8-8. C2C RI Test Scripts Report.....	48

Figure 8-9. C2C RI Test Scripts Report Details	49
Figure 8-10: Sample Test Case Report.....	50
Figure 8-11: Sample Test Procedure Report.....	51
Figure 8-12. Report Configuration Window.....	52
Figure 8-13. Report Log Window	53
Figure 8-14. Test Log Reporting Screen	54
Figure 8-15. Test Log Report Completion	55
Figure 8-16. Standard Conformance/Compliance Test	56
Figure 8-17. Standard Conformance/Compliance Test Details	57
Figure 8-18: Section 1201 Standard Conformance/Compliance Test.....	58
Figure 8-19: Section 1201 Conformance/Compliance Test Details.....	59
Figure 8-20. C2C RI Test Case Summary Report	60
Figure 8-21. C2C RI Test Case Summary Report Details	61
Figure 8-22. C2CRI Test Case Detail Log Report.....	62
Figure 8-23. C2C RI Test Case Detail Log Report Details	63
Figure 8-24. C2C RI Message Summary Report.....	64
Figure 8-25. C2C RI Message Summary Report Details	65
Figure 8-26. C2C RI Message Detail Report	66
Figure 8-27. C2C RI Message Detail Report Details	67
Figure 8-28. C2C RI Test Script Action Log Report	68
Figure 8-29. C2C RI Test Script Action Log Report Details	69
Figure 8-30. CSV Export Status Report.....	70
Figure 8-31: Report Anomaly Example	71
Figure 9-1. File Deletion Warning.....	72
Figure 9-2. DOS Jarsigner Arguments	74
Figure 9-3. Test Case Data Editor Open Dialog.....	77
Figure 9-4. Test Case Parameters Editor.....	78
Figure 9-5: Test Case Parameters Editor Menu Bar.....	79
Figure 9-6: Test Case Parameter Editor Iteration Bar.....	79
Figure 9-7: Test Case Viewer Showing Iteration Name.....	80
Figure 9-8: Test Case Parameters Editor Groups Tab Bar	80
Figure 9-9: Test Case Viewer Showing Group Names	81
Figure 9-10: Test Case Parameter Editor Showing Attribute Name and Attribute Value	82
Figure 9-11: Test Case Parameters Viewer Showing Attributes Names and Attribute Values	83
Figure 9-12: Test Case Parameters Editor showing Attribute Baseline Parameter	84
Figure 9-13: Test Case Editor Showing Selected Iteration	85
Figure 9-14: Rename Iteration Dialog	85
Figure 9-15: Renaming Iteration dialog showing old and new Iteration Name	86
Figure 9-16: Iteration Showing New Name.....	87
Figure 9-17: Replace Group Dialog	88
Figure 9-18: Replace Group Dialog Showing Old and New Group Name....	88
Figure 9-19: Group Parameters Action Dialog	88
Figure 9-20: Replaced Group Showing Cleared Parameters.....	89
Figure 9-21: Replaced Group Showing Parameters.....	90

Figure 9-22: Add Parameter Dialog.....	91
Figure 9-23: Add Parameter Form Completed with Data	92
Figure 9-24: New Parameter Added to Group	93
Figure 9-25: Test Configuration Editing Screen Showing Selected Test Case	94
Figure 9-26: Test Case Viewer Showing Newly Added Parameter	95
Figure 9-27: Confirm Parameter Deletion	95
Figure 9-28: Selected Parameter with Move Parameter Up and Move Parameter Down Buttons enabled	96
Figure 9-29: Parameter Value in List.....	97
Figure 9-30. Test Case Data File Example	98
Figure 10-1. Overwriting a Test Log File Confirmation Dialog	109
Figure 10-2. Cancelling File Changes Confirmation Dialog.....	109
Figure 10-3. Closing a File Confirmation Dialog	110
Figure 10-4. Insufficient Need and Requirement Error Log.....	110
Figure 10-5. Configuration Not Saved Window	111
Figure 10-6: 32-bit File Size Warning Window.....	111
Figure 10-7. Test Configuration File Deletion Confirmation Dialog.....	112
Figure 10-8. Test Log File Deletion Confirmation Dialog	113

List of Attributes

- Figure 3-1: TransCore ITS, LLC, February 2014.
- Figure 4-1: TransCore ITS, LLC, February 2014.
- Figure 4-2: TransCore ITS, LLC, February 2014.
- Figure 4-3: TransCore ITS, LLC, February 2014.
- Figure 5-1: TransCore ITS, LLC, February 2014.
- Figure p. 15: TransCore ITS, LLC, February 2014.
- Figure p. 16: TransCore ITS, LLC, February 2014.
- Figure 5-2: TransCore ITS, LLC, February 2014.
- Figure 5-3: TransCore ITS, LLC, February 2014.
- Figure 5-4: TransCore ITS, LLC, June 2021.
- Figure 5-5: TransCore ITS, LLC, February 2014.
- Figure p. 20: TransCore ITS, LLC, February 2014.
- Figure 5-6: TransCore ITS, LLC, February 2014.
- Figure 5-7: TransCore ITS, LLC, June 2021.
- Figure p. 24: TransCore ITS, LLC, February 2014.
- Figure 5-8: TransCore ITS, LLC, December 2017.
- Figure 5-9: TransCore ITS, LLC, February 2014.
- Figure 5-10: TransCore ITS, LLC, June 2021.
- Figure 5-11: TransCore ITS, LLC, June 2021.
- Figure 6-1: TransCore ITS, LLC, July 2016.
- Figure 6-2: TransCore ITS, LLC, July 2016.
- Figure 6-3: TransCore ITS, LLC, July 2016.
- Figure 6-4: TransCore ITS, LLC, July 2016.
- Figure 6-5: TransCore ITS, LLC, June 2021.
- Figure 6-6: TransCore ITS, LLC, June 2021.
- Figure 6-7: TransCore ITS, LLC, June 2021.
- Figure 6-8: TransCore ITS, LLC, July 2016.
- Figure 6-9: TransCore ITS, LLC, July 2016.
- Figure 7-1: TransCore ITS, LLC, February 2014.
- Figure 7-2: TransCore ITS, LLC, June 2016.
- Figure 7-3: TransCore ITS, LLC, February 2014.

- Figure 7-4: TransCore ITS, LLC, February 2014.
- Figure 8-1: TransCore ITS, LLC, November 2014.
- Figure 8-2: TransCore ITS, LLC, November 2016.
- Figure 8-3: TransCore ITS, LLC, February 2016.
- Figure 8-4: TransCore ITS, LLC, February 2014.
- Figure 8-5: TransCore ITS, LLC, February 2014.
- Figure 8-6: TransCore ITS, LLC, February 2014.
- Figure 8-7: TransCore ITS, LLC, February 2014.
- Figure 8-8: TransCore ITS, LLC, February 2014.
- Figure 8-9: TransCore ITS, LLC, February 2014.
- Figure 8-10: TransCore ITS, LLC, November 2016.
- Figure 8-11: TransCore ITS, LLC, November 2016.
- Figure 8-12: TransCore ITS, LLC, June 2021.
- Figure 8-13: TransCore ITS, LLC, February 2014.
- Figure 8-14: TransCore ITS, LLC, September 2017.
- Figure 8-15: TransCore ITS, LLC, February 2014.
- Figure 8-16: TransCore ITS, LLC, February 2014.
- Figure 8-17: TransCore ITS, LLC, February 2014.
- Figure 8-18: TransCore ITS, LLC, November 2016.
- Figure 8-19: TransCore ITS, LLC, November 2016.
- Figure 8-20: TransCore ITS, LLC, February 2014.
- Figure 8-21: TransCore ITS, LLC, February 2014.
- Figure 8-22: TransCore ITS, LLC, February 2014.
- Figure 8-23: TransCore ITS, LLC, February 2014.
- Figure 8-24: TransCore ITS, LLC, February 2014.
- Figure 8-25: TransCore ITS, LLC, February 2014.
- Figure 8-26: TransCore ITS, LLC, February 2014.
- Figure 8-27: TransCore ITS, LLC, February 2014.
- Figure 8-28: TransCore ITS, LLC, February 2014.
- Figure 8-29: TransCore ITS, LLC, February 2014.
- Figure 8-30: TransCore ITS, LLC, February 2014.
- Figure 8-31: TransCore ITS, LLC, October 2017
- Figure 9-1: TransCore ITS, LLC, February 2014.
- Figure 9-2: TransCore ITS, LLC, February 2014.
- Figure 9-3: TransCore ITS, LLC, June 2021.

- Figure 9-4: TransCore ITS, LLC, June 2021.
- Figure 9-5: TransCore ITS, LLC, June 2021.
- Figure on p. 80 TransCore ITS, LLC, June 2021.
- Figure 9-6: TransCore ITS, LLC, June 2021.
- Figure 9-7: TransCore ITS, LLC, June 2021.
- Figure 9-8: TransCore ITS, LLC, June 2021.
- Figure 9-9: TransCore ITS, LLC, June 2021.
- Figure 9-10: TransCore ITS, LLC, June 2021.
- Figure 9-11: TransCore ITS, LLC, June 2021.
- Figure 9-12: TransCore ITS, LLC, June 2021.
- Figure 9-13: TransCore ITS, LLC, June 2021.
- Figure 9-14: TransCore ITS, LLC, June 2021.
- Figure 9-15: TransCore ITS, LLC, June 2021.
- Figure 9-16: TransCore ITS, LLC, June 2021.
- Figure 9-17: TransCore ITS, LLC, June 2021.
- Figure 9-18: TransCore ITS, LLC, June 2021.
- Figure 9-19: TransCore ITS, LLC, June 2021.
- Figure 9-20: TransCore ITS, LLC, June 2021.
- Figure 9-21: TransCore ITS, LLC, June 2021.
- Figure 9-22: TransCore ITS, LLC, June 2021.
- Figure 9-23: TransCore ITS, LLC, June 2021.
- Figure 9-24: TransCore ITS, LLC, June 2021.
- Figure 9-25: TransCore ITS, LLC, June 2021.
- Figure 9-26: TransCore ITS, LLC, June 2021.
- Figure 9-27: TransCore ITS, LLC, June 2021.
- Figure 9-28: TransCore ITS, LLC, June 2021.
- Figure 9-29: TransCore ITS, LLC, June 2021.
- Figure 9-30: TransCore ITS, LLC, February 2014.
- Figure 10-1: TransCore ITS, LLC, February 2014.
- Figure 10-2: TransCore ITS, LLC, February 2014.
- Figure 10-3: TransCore ITS, LLC, February 2014.
- Figure 10-4: TransCore ITS, LLC, February 2014.
- Figure 10-5: TransCore ITS, LLC, February 2014.
- Figure 10-6: TransCore ITS, LLC, February 2017
- Figure 10-7: TransCore ITS, LLC, February 2014.

Figure 10-8: TransCore ITS, LLC, February 2014.

Document Revision History

Version #	Date	Purpose
1	4/2/2010	Initial Version for C2C RI Internal Release 1
2	1/21/2011	C2C RI Release 2 updates
3	10/16/2013	C2C RI Release 3 updates
4	1/17/2014	C2C RI Acceptance Test updates
5	4/19/2016	C2C RI Release 3.1 updates
6	7/18/2016	C2C RI Release 1.1 updates
7	5/12/2017	C2C RI Release 2.0 updates
8	12/11/2017	C2C RI Release 2.1 updates
9	5/2020	C2C RI Release 3 updates
10	5/2021	C2C RI Release 3.1 updates

1.0 Introduction

This section provides an overview and background information. It describes the manual's purpose as well as the assumed background of the reader. It presents the topics covered by the manual's scope and the purpose of the accompanying software.

1.1 Identification

The Center-to-Center (C2C) Reference Implementation (RI) User Manual provides guidance and instructions on the use of the C2C RI software application.

1.2 Purpose

The C2C RI is a tool for verifying a target system's conformance to designated Intelligent Transportation System (ITS) C2C standards. This user manual provides the background information, concepts, and instructions a user needs to utilize the C2C RI in the verification process. The information is organized in accordance with the IEEE 1063-2001 Standard for Software User Documentation using the task-oriented instructional mode of presentation.

The User Manual provides guidance and reference material regarding:

- Conformance Testing for Selected Standards;
- Conformance Test Definition and Setup;
- Conformance Test Execution;
- Post Test Report Generation; and
- RI Tool Options and File Maintenance.

The manual is organized in sections that track the sequence an RI user would follow in defining, executing, and reporting on a conformance test. Section 2.0 lists the references related to the development of the RI. Section 3.0 presents the background and conceptual information that underlies the RI and the conformance testing process. Section 4.0 describes procedures commonly used in managing the software installation and navigating the RI's user interface. Section 5.0 presents an overview of the RI modes and how to navigate between the modes. In Section 6.0 the details of defining a test configuration are covered. Section 7.0 describes how to use the output of the test definition to execute tests and record the results. In Section 8.0 the recorded results can be formatted into reports for use outside of the RI. Auxiliary functions for managing the RI installation are discussed in Section 9.0, and Section 10 provides information on the RI's error identification along with techniques for addressing common errors.

1.3 Intended Audience

This manual is intended for an experienced systems professional, with a basic background with Intelligent Transportation Systems (ITS) standards, who wants to use the C2C RI application software to perform conformance testing of a system interface which implements the supported ITS C2C Standards. RI users are expected to be experienced with the testing of software in general, particularly specification-based testing, and testing strategies.

All users are expected to understand the issues involved with establishing a network connection between the system under test (SUT) and the RI. Network configuration and management is a complex issue and is not part of this User Manual.

It is assumed that RI users have a basic knowledge of the ITS C2C Standards – in particular the Traffic Management Data Dictionary (TMD) and Message Sets for External Traffic Management Center Communications (MS/ETMCC). These standards are the initial information layer standards supported by the RI. Users are also expected to be familiar with the NTCIP 2306 application layer standard that defines the protocols and transport mechanisms for communication between centers.

1.4 Scope

This user's manual covers the features available in RI Release 3.0 (*except where noted by a release identifier in italics*). These features include the RI's four operational modes, its options, and its file maintenance capabilities.

- The ability to select project level user needs and requirements is discussed.
- Execution of application level tests is discussed.
- Generation of reports is described.
- The RI can be configured to operate as either an owner center or an external center.

The C2C RI is developed to assist in the determination of an implementation's conformance to C2C standards. To accommodate the need for specific project extensions to the standards, the RI can be extended to incorporate custom test suites that verify SUT conformance to these extensions. The RI's user manual describes how to integrate these tests into the RI. It is the users' responsibility to provide test suites that determine a SUT's conformance with the extensions, and the development of these suites is not described by this manual.

1.5 Software Purpose

The purpose of the C2C RI software is to assist in determining whether a C2C interface conforms to C2C ITS Standards. Conformance testing determines the extent that an item (i.e. interface) implements the individual requirements of a particular standard. Additional information on conformance testing is provided in Section 3.0, Concept of Operations (ConOps).

The important factors which characterize conformance testing are

- The SUT defines the boundaries for testing.

- The tests are executed by a dedicated test system (C2C RI) that controls the testing and observes the SUT's behavior.
- The tests are performed on open standardized interfaces.

The RI conducts its conformance testing of the SUT C2C interface by two means. First it examines the interface description document (Web Services Description Language or WSDL) for conformance to the standards. It then tests the interface by communicating with the SUT. This communication consists of dialogs between the RI and the SUT. The test dialogs are determined by the selection of user needs and their derived requirements for the particular test configuration. Because the C2C RI controls the sequence and contents of the protocol messages sent to the SUT, it can test a wide range of both expected (valid) and unexpected (invalid) behaviors.

Three functions are performed by the RI. Configuration of a test is the first function, and this function allows the selection of user needs and requirements to be tested. It results in a set of test cases that can be used to verify that the SUT conforms to the selected requirements.

Execution of a test is the second function of the RI. When carrying out this function, the RI enables the user to select test cases for execution and then runs those test cases logging the details and results of the test.

The third function is a report function, provided to assist the RI user in preparing test result reports for sharing with other stakeholders. The RI's report function can also be used simply to report a system's needs and requirements in a useful format, even if the RI is not being used to test standards compliance.

The RI can act as a client, obtaining information from a system that provides data about its operations and therefore is known as an owner center (OC). Additionally the RI can mimic an OC and provide data to an external center (EC) in order to verify the EC's conformance to the standards for these message exchanges.

In anticipation of a long life cycle for the RI software, it is written in the Java language. Therefore, the test suites that support the RI's conformance testing are distributed in Java Archive (JAR) files. This approach to the RI's implementation enables portability of the RI and its test suites. This user manual avoids discussing the technology behind the application, but does mention it when an understanding of the technology is essential to the user's operation of the RI. The areas where Java technology is most apparent are recognizing pre-defined test suites and the generation of custom test suites.

1.6 Document Conventions

In this document the following conventions are used:

Table 1-1. Document Conventions

Content	Format	Example
Menu Item or Button Name	Bold+Helvetica	File Browse
Field Name	Normal Times New Roman	<i>Test Configuration Name</i>

1. Introduction

Content	Format	Example
Simultaneous Keystrokes	Key1+Key2	ALT+O

2.0 References

	Concept of Operations for a Center to Center Reference Implementation
	Software Requirements Specification for a Center to Center Reference Implementation
	Software Design Document for a Center to Center Reference Implementation
ITE TMDD	Traffic Management Data Dictionary (TMDD) and Message Sets for External Traffic Management Center Communications (MS/ETMCC) Versions 3.03c, 3.03d, and 3.1
IEEE STD 1016-1998	IEEE Recommended Practice for Software Design Descriptions
IEEE STD 829-1998	Standard for Software Test Documentation
IEEE STD 830-1998	Recommended Practice for Software Requirements Specifications
IEEE STD 1063-2001	Standard for Software User Documentation
NTCIP 2306	Application Profile for XML Message Encoding and Transport in ITS Center-to-Center Communications (NTCIP-C2C XML) Version 01.69r
NTCIP 2304:2002	Application Profile for DATEX-ASN (AP-DATEX) Version 01.08f

3.0 Concept of Operations (ConOps)

This section explains the conceptual background for the use of the C2C RI software. The background information is organized under two topics: Conformance Testing Concepts and C2C RI Usage Concepts.

3.1 Conformance Testing Concepts

To understand the concept behind the C2C RI, it is necessary to first have an understanding of the relationship between conformance testing and compliance testing. This background is discussed in the following paragraphs.

3.1.1 ITS Standards Program Overview

The ITS standards program is a joint effort of the U.S. Department of Transportation, the Federal Highway Administration (FHWA) and its partners known as Standards Development Organizations (SDO) including the Institute of Transportation Engineers (ITE), the National Electrical Manufacturers Association (NEMA), the American Association of State Highway and Transportation Officials (AASHTO), the Institute of Electrical and Electronics Engineers (IEEE), and the Society of Automotive Engineers (SAE). The goal of these organizations is to enable interoperable communication between transportation centers, equipment, and users. To achieve this goal requires several objectives, one of which is having a common set of non-proprietary communications protocols that enable transportation centers to communicate with each other. These protocols are referred to as Center-to-Center (C2C) communications.

With respect to C2C communications, the implementation item is the C2C interface between systems. This interface derives from various ITS standards of which the NTCIP 2306 Application Profile and the TMDD External Message Set are a subset of the available standards. The NTCIP 2306 standard specifies the encoding and transport mechanisms for accomplishing C2C information exchange. The TMDD External Message Set standard defines the message exchanges between centers and the data content of those messages.

Transportation system implementations supporting C2C communications must communicate and behave according to these standards protocol specifications to attain this interoperability. One way to determine whether these transportation system implementations accomplish this need is through an activity known as protocol conformance testing.

At the time of C2C RI development, there is an unresolved conflict between naming conventions in the NTCIP 2306 Application Profile and the TMDD External Message Set. The C2C RI allows TMDD naming conventions for information level testing and uses NTCIP 2306 naming conventions for application level testing (see Section 6.0).

3.1.2 Testing Types

Conformance testing is the process of verifying that an item properly implements a specific standard. The item may be an individual system, a set of software programs, or an individual program. The reference standard may be stand-alone or may be associated with other complementary standards; however, only a single standard is the target of the conformance testing. Other standards will have their own conformance testing suite to determine an item's agreement with the standard.

Protocol conformance testing is a kind of functional testing: a software system is solely tested for conformance with respect to the requirements given in its specification. The idea behind this is that only systems with correctly implemented protocols can communicate successfully with peer systems. Only the observable behavior of the protocol implementation is tested (i.e., the interactions of the system with its peers). No testing is conducted related to the internal details of the system.

Conformance testing can be performed by different parties. First, the supplier of a system tests its product as part of its development. Agencies, or their representative contractors, test products for correct functioning prior to acceptance. Finally, independent third party test laboratories can perform conformance tests for any of the previously mentioned parties.

Compliance testing verifies that an item is built per a specific project's specifications. The project would include the application of numerous standards and incorporate project specific requirements that may restrict or extend options made available by the standard. Therefore, compliance testing verifies a more specific set of requirements than conformance testing. For example, a standard may contain a data element that can range in value from 0-255. A project requirement may limit the range of values to a range of 0-128, so the project compliance is more restrictive than the standard. Conformance testing would verify the data element exists and supports required values in the 0-255 range of the standard, while compliance testing would verify the 0-128 range specified for the project.

3.1.3 Conformance Testing Process

The process of conformance testing is generally achieved in three phases. The first phase is test generation and includes the development of an abstract test suite for a particular standard specification. This test suite is abstract in the sense that tests are developed independently of any particular implementation. It contains all possible tests for a particular standard, including all possible options within that standard. A test suite is a set of test cases that verify an implementation conforms to a specific standard. Each test case specifies the inputs, predicted results, and conditions for executing the test. Each test case focuses on ascertaining an implementation's conformance to one or more requirements of the standard.

The second phase is test implementation. The specifics of the testing environment and the portion of the test system which implements the specification (sometimes referred to as the Implementation Under Test (IUT)), are taken into account. The abstract test cases within the abstract test suite are transformed into tests (often called executable tests) which can be executed or interpreted on an actual test system.

The starting point of this phase is the abstract test suite. A selection from the abstract test suite must be made, because it contains test cases for both mandatory and optional standard requirements. It does not make sense to test for options that are not implemented by the SUT. Typically this information is captured in a Protocol Requirements List (PRL). The PRL contains only protocol dependent information. Before a test can be executed, information about the SUT and its environment

3. Concept of Operations

must also be obtained. This information is called PIXIT (Protocol Implementation eXtra Information for Testing). The PIXIT may contain address information for the SUT or other parameter values which are necessary to execute the test suite.

The selected test cases, with parameter values according to the PIXIT, form the executable test suite, which can be executed on a real SUT.

The last phase is test execution. The tests resulting from the completion of test implementation are executed with a particular SUT, and the resulting behavior of the SUT is observed. This leads to the determination of a verdict about conformance of the SUT with respect to the standard protocol specification. The verdict is either pass, fail or inconclusive. Pass indicates that the test was executed successfully, and the goal of the test was achieved. Fail indicates that the SUT does not conform to the specification. Inconclusive indicates that no evidence of non-conformance was observed; however, the test purpose was not achieved. The results of this phase are documented in test reports.

3.2 Conformance Testing with the C2C RI

The C2C RI is a conformance testing tool for C2C ITS Standards that provides a consistent and repeatable approach to verifying an item's implementation of the standard. Use of the C2C RI follows the conformance testing process previously discussed. The architecture of the C2C RI is illustrated in Figure 3-1 and its usage is described in the following paragraphs.

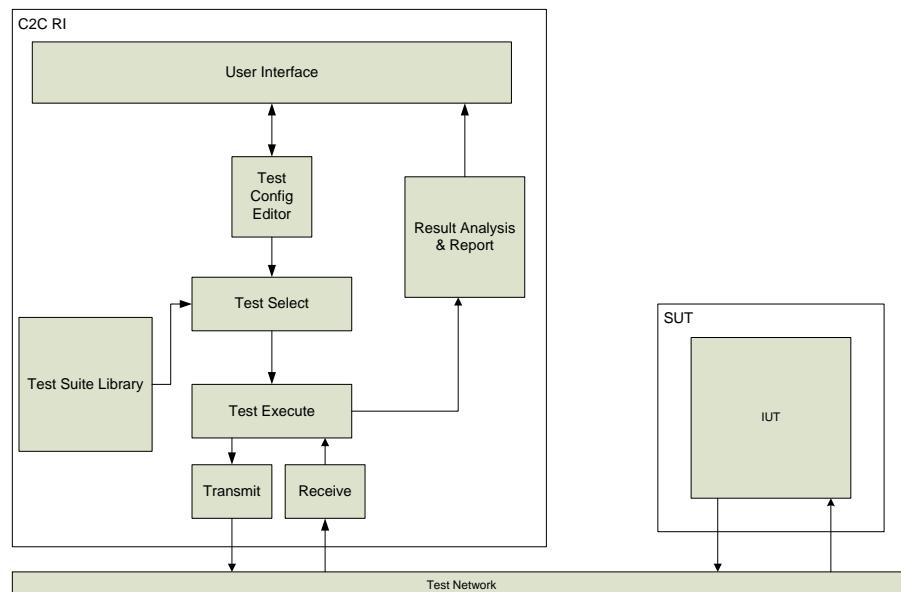


Figure 3-1. C2C RI Test System Architecture
(Source: TransCore ITS, LLC, February 2014.)

3. Concept of Operations

The C2C RI is distributed as a bundled software package. It contains the basic C2C RI application and a set of pre-defined ITS C2C standard test suites. Each test suite represents an abstract test suite for testing a combination of ITS Information Level and Application Level standards. For instance, one test suite might contain a set of tests necessary for testing an implementation which uses the TMDD v3 Information Layer Standard with portions of the NTCIP 2306 Application Layer Standard.

After launching the C2C RI application, the test developer uses the user interface to access the test configuration editor capability. This editor provides the test developer with the means to specify test configuration. The test configuration basically defines an executable test suite. It stores the following information:

- The user selected test options (PRL information);
- Information required to identify the SUT and other testing parameters (PIXIT information); and
- Reference to the user selected test suite (the abstract test suite that will be pulled from the test suite library).

Once the user has defined and saved a test configuration, the user may later select that configuration to be used to run a test.

To execute a test, the user provides a unique test name and selects the test configuration that applies to the test. The C2C RI then executes the test cases within the referenced test suite using the PRL and PIXIT information provided in the test configuration file. Based on the actions specified in the test suites, the C2C RI executes dialogs with the SUT utilizing the applicable information, encoding, and transport protocols. The results of the test cases are stored in a test log.

After the test is complete, the user utilizes the C2C RI user interface to access the application's reporting capability. The user can create a number of reports from the test log. Each report is created as a PDF file. The user may also export the test log as a CSV file, which can be imported and utilized by separate data analysis software applications. Note: The logs created by the program are XML files.

The report files contain an Author ID/Created by ID field. If the user has provided the C2CRIUserName property definition in the options dialog ([Figure 5-7. RI Parameters Options Dialog](#)[Figure 5-7. RI Parameters Options Dialog](#)), this value is used for the Author ID. The Created by ID is the user name that is associated with the login account of the RI user.

4.0 Procedures

4.1 Installing the C2C RI Application

The RI requires the following as a minimum system:

Table 4-1. RI System Requirements

Feature	Minimum (Estimates)	Recommended
CPU	Intel Microprocessor (1GHz)	Intel Microprocessor (>2 GHz)
RAM	512 MB	4 GB
Storage	512 MB of available hard drive space	1 GB
Communications	100 Mbps Ethernet Interface	1000 Mbps Ethernet Interface
Operating System	Microsoft Windows 8	Microsoft Windows 10
Support Software	Zip file management	
Support Software	Java SE Runtime Environment (JRE) Version 11.0.2 or higher (source: http://java.sun.com/javase/downloads/index.jsp)	

To install the C2C RI application, the computer system must have the support software applications installed. The C2C RI installer includes the Java SE JRE.

When updating an existing C2C RI installation, it is necessary to remove the existing installation prior to installing the updated application. First, any data files to be carried over to the updated installation must be backed-up to another directory. Then the uninstall software (uninst.exe) in the existing installation directory can be run to remove the existing installation.

Follow these instructions to install the C2C RI application:

1. Uninstall any earlier versions of the C2C RI application.
2. Copy the c2cri_Release 2_Installer.exe to your PC. Double click on the file to launch the install program.
3. The Main Installation screen displays a figure similar to the one shown in Figure 4-1.
Click Next.



Figure 4-1. C2C RI Setup Screen
(Source: TransCore ITS, LLC, February 2014.)

4. You are prompted to choose a directory in which to install the application shown similar in Figure 4-2. You may accept the default directory or create your own directory path. Click “Next” when done. (Use the default C:\C2CRI directory, since settings in the property file are set there by default.)

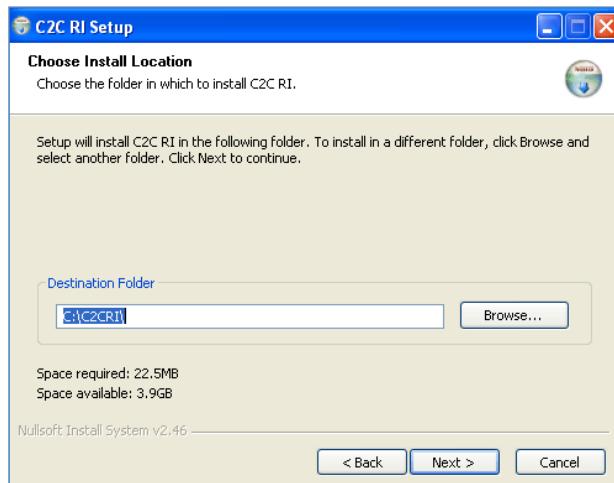


Figure 4-2. C2C RI Setup Default Installation Directory
(Source: TransCore ITS, LLC, February 2014.)

The “Choose Start Menu Folder” window displays a dialog similar to the one shown in Figure 4-3. This window allows you to choose where to install the shortcut in your Start Menu. Choose a window and click the “Install” button to install the software.

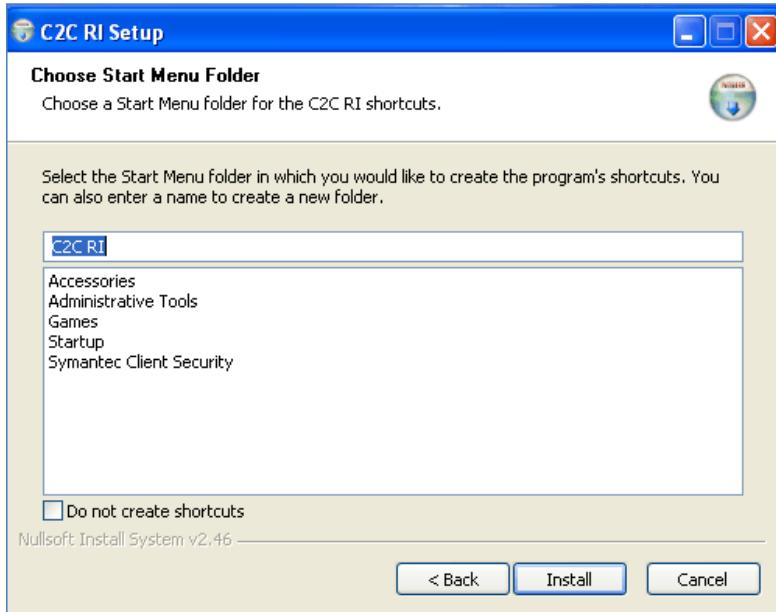


Figure 4-3. Choosing Start Menu Folder Screen
(Source: TransCore ITS, LLC, February 2014.)

5. An installation window displays, showing the status of the installation.
6. Once the installation is complete the “Completing the C2C RI Setup Wizard” window displays. Click the “Finish” button to complete the installation.

4.2 Preparing the C2C RI Test Environment

This section discusses adding the computing system that executes the RI to the environment containing the SUT. Because each target test environment is unique, this section discusses the issues an RI user would be expected to encounter and means to address the issues. Common terminology will be used to describe the relevant staff, but it is left to the reader to identify the proper personnel to contact within each environment.

The basic assumption is that the computer system housing the RI must be installed on a network capable of communicating with the SUT. This may mean that the computer uses DHCP to obtain an IP address or that the administrative network staff must identify an IP address for the RI host system.

This may also involve locating the RI host system inside or outside of a firewall and incorporating appropriate security mechanisms in order to secure access to the SUT.

The communications protocols utilized between the RI and the SUT are determined by the Application Layer Standard that is selected as part of the test configuration. For example, the NTCIP 2306 standard covers the use of a number of TCP/IP transport protocols (FTP, HTTP, SOAP). The particular IP addresses and ports required for communication with the SUT are defined within a Web Service Description Language (WSDL) file that defines the interface.

As part of the definition of the test configuration, the user must specify the location of this WSDL file. To specify the location of the WSDL file in a test configuration file, enter the URL for the file in the Web Service URL field of the editor shown in [Figure 6-2. Test Definition: System Under Test \(SUT\) Panel](#). The example shown applies for the case when the WSDL file is stored locally on the RI host computer. In this case the WSDL file named release2+.wsd is stored in the c:\c2cri\testfiles directory.

In the case of testing a real OC's subscriptions with a test machine (e.g., a laptop) as the EC running the RI, the user must define the test machine EC's URL that is customized for the SOAP Subscriber Callback Listener Service provided in the WSDL. The URL must be included in the data file for each OC test case. The URL form for the subscriber is
[protocol]://[HostAddress]:[Port]/LocationURLPath/[ServiceName]/[PortName]/[OperationName]

In addition, there are a number of user adjustable communications properties defined for this standard that may be edited using the C2C RI options dialog.

4.3 Starting the C2C RI Application

Windows

To launch the application through windows, double-click on the C2C RI shortcut located in the folder where the application was installed.

Command Line

To launch the application from the command line, go to Start>Run and enter “cmd” into the Open field of the run dialog box. Click OK or press enter.

To change to a local directory that contains your C2C RI files, type cd followed by the path to the desired directory after the prompt.

Example: If your C2CRI files are stored in c:\sites\my_site, enter cd c:\sites\my_site.

At the prompt, enter C2CRI and press enter.

5.0 Navigating within the RI

This section describes the four modes of the RI and the tools for navigating between the modes. These tools consist of menus and accelerator keys.

5.1 RI Operating Modes

The RI operates at all times in one of four independent modes. The main mode is the initial mode of the RI at startup. In this mode the banner logo is displayed and no other configuration, test, or report documents are open. Users place the RI into Test Definition mode to select user needs and requirements for configuring a conformance test. Users enter the RI's Test Execution mode to select available tests and perform the test. Users can also set the RI in Test Report mode to prepare final report output. From the main mode, the RI can exit or enter one of the other three modes. This structure prevents users from executing tests with incomplete (i.e., invalid) test configurations.

The following describes how the menus and accelerator keys are used to navigate between these modes and the dialogs presented when navigating between modes.

5.2 User Interface Screen

Launching the RI application software opens a user interface window as shown in Figure 5-1. The window has a menu bar section that shows the different application functions that can be performed. On first use, the user interface also displays a guided user mode, or wizard, on the top right of the screen. More information about disabling, re-enabling, and using the wizard is included in Sections 5.3.3.2 and 5.5. It also shows, in the top right hand corner of the window's title bar, options to minimize the application, maximize/restore the application, and close the application. It is important to note that the RI application will close only when it has completed editing, executing, or reporting on a test in order to preserve information regarding the test.

5. Navigating within the RI

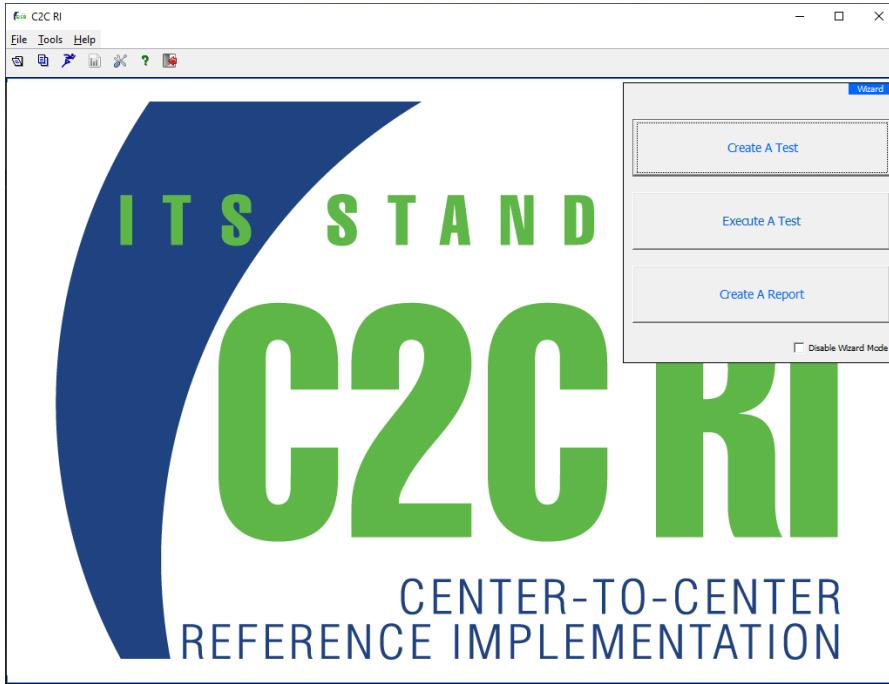


Figure 5-1. RI User Interface Window: Main
(Source: TransCore ITS, LLC, June 2021.)

5.3 Menu Navigation

The menu bar gives users access to the user interface commands for navigating between modes. Users can also carry out all of these commands using the accelerator keys (a combination of ATL+letter) shown in the menus and summarized in Section 5.4: Accelerator Key Navigation.

5.3.1 Menu Bar

The RI user interface menu bar includes these three different pull down menu items for file, tools, and help facilities as shown below.



(Source: TransCore ITS, LLC, February 2014.)

5.3.2 File Menu

The **File** menu includes five items along with five accelerator keys. The **New** and **Open** dialogs are for defining the user needs and requirements of the SUT within the Test Definition mode. The **Execute**

5. Navigating within the RI

dialog is for identifying and conducting an iteration of the test suite for a SUT in the RI's Test Execution mode. **Reports** is for preparing output from the configuration and test execution logs in Test Report mode. These options are shown in more detail in the following sections. **Exit** stops the program but is only available when the RI is in the main mode so that none of the information generated by the three other modes is lost.



(Source: TransCore ITS, LLC, February 2014.)

| **New** opens the dialog shown in [Figure 5-2](#) from which the user creates a new Test Configuration in the system. Accelerator key (ALT+N) may also be used to perform this task.

A name is entered into the *Test Configuration Name* field to uniquely identify the test settings and the file storing the configuration. The name is limited to a total of 25 characters excluding the period and file extension. A name can contain upper and lower case alphabetic, numeric, blank (), underscore (_), or hyphen (-) characters. The file will be stored in the directory specified in the *Path* field. To change this directory the **Path** button is clicked. Note that the default directory can be modified using the **Options** menu selection under the **Tools** menu.

5. Navigating within the RI

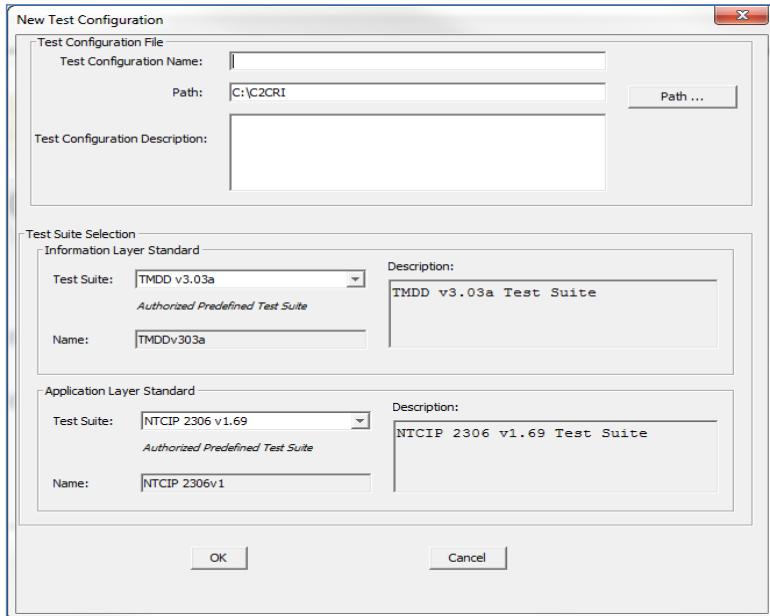


Figure 5-2. New Test Configuration Dialog
(Source: TransCore ITS, LLC, February 2014.)

The *Test Configuration Description* field is provided to enter any details about the test configuration. This can include the relevant sub-profiles to be used or which of several interfaces on a SUT may be the target of the test.

The Test Suite Selection group allows the identification of the individual standards against which the SUT's interface will be tested. The Information Layer Standard defines the applicable dialogs, messages, and data elements, while the Application Layer defines the message encoding and transport mechanisms. These *Test Suite* fields present a list of standards from which a single standard is chosen. Authorized predefined test suites are the set of tests used to verify that the SUT conforms to the standard. These are digitally signed to verify their status as properly authorized suites. Because the standards may be extended, additional test suites may appear with a User Defined Test Suite label presented in red to highlight that the test suite contains custom extensions. When the entry field's list is expanded, the user defined test suites are shown with an asterisk (*) prefixed to the test suite name. User defined test suites are considered extensions and therefore are not digitally signed.

Open opens the dialog shown in [Figure 5-3](#) from which the user can open an existing Test Configuration that has a valid name and a meaningful content. Accelerator key (ALT+O) may also be used to perform this task. The dialog displays RI configuration files with the file extension "ricfg" for selection. It opens in the default configuration file directory (see the [Options Dialog](#) section).

5. Navigating within the RI

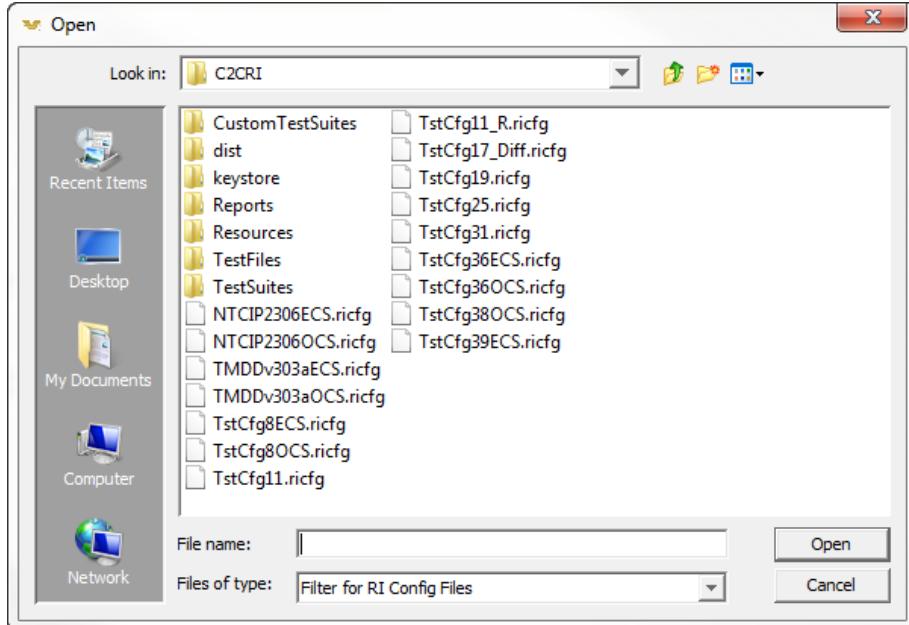


Figure 5-3. Open Dialog
(Source: TransCore ITS, LLC, February 2014.)

Execute opens the dialog shown in [Figure 5-4](#) from which you can select a valid configuration file for conducting a test. Accelerator key (ALT+X) may also be used to perform this task. After the valid configuration file is opened, the user can select one or more test cases and execute them. Note that only valid test configuration files are listed. If the list appears to be missing a test configuration, then the dialog should be cancelled, and the Test Definition mode should be entered in order to verify that the expected test configuration is valid.

The *Test Name* and *Test Notes* fields provide a means to uniquely identify a test suite execution to support repeated executions of a test suite. The test suite could be executed by developers several times during the development process and could be repeated again during the acceptance testing. The contents of these fields can be used to identify these various executions. The *Test Name* field also specifies the file name used to store the results and logs for the specific test execution.

5. Navigating within the RI

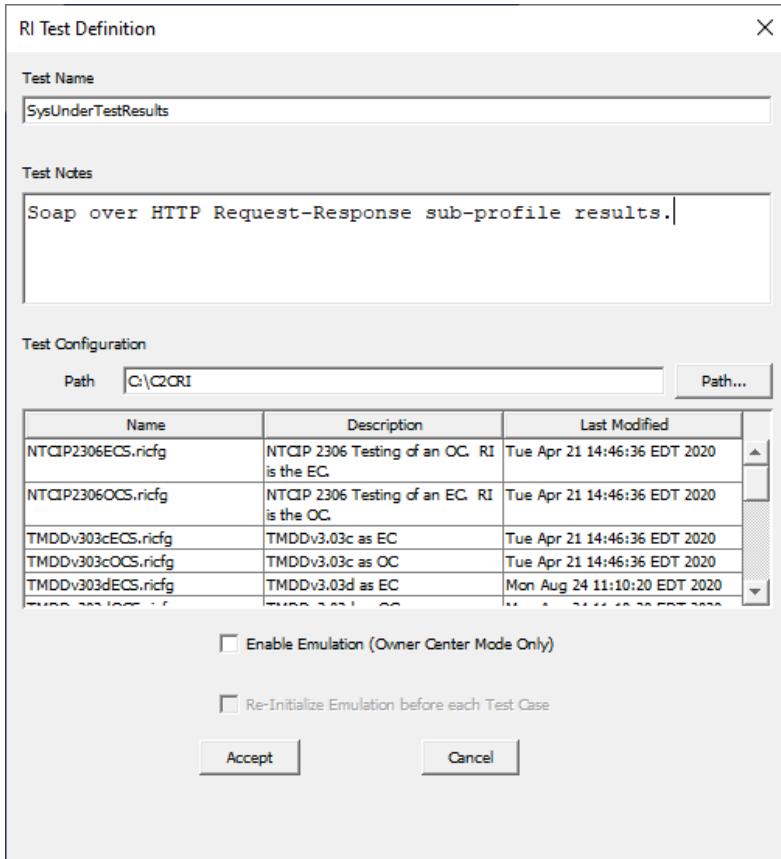


Figure 5-4. RI Test Definition Dialog
(Source: TransCore ITS, LLC, June 2021.)

Reports opens the window shown in [Figure 5-5](#)[Figure 5-5](#), from which the user can enter information for a configuration or log report. A report is created and can be viewed within the RI or later using a system utility that displays PDF formatted files. Accelerator key (ALT+R) may also be used to perform this task.

5. Navigating within the RI

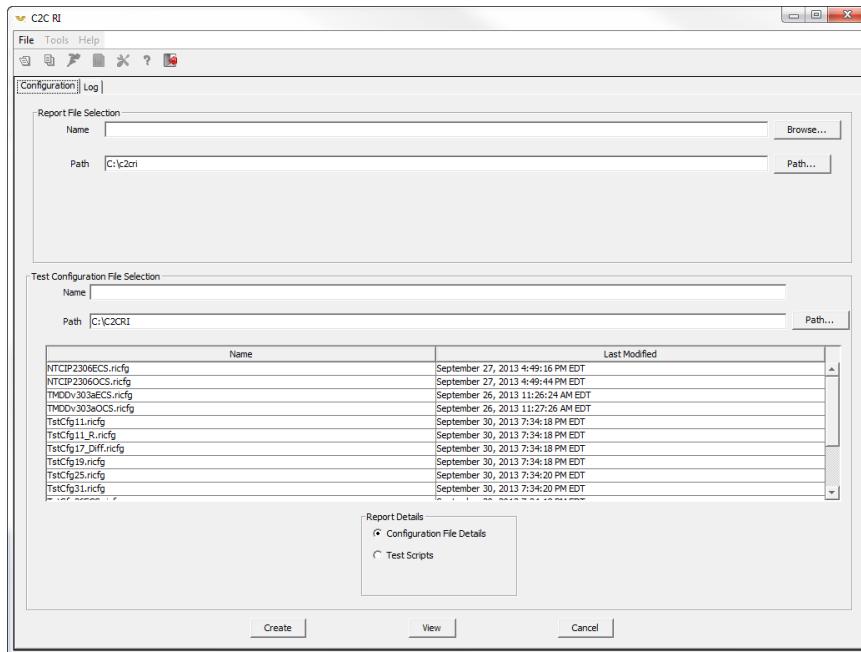


Figure 5-5. Reports Window
(Source: TransCore ITS, LLC, February 2014.)

Exit closes the RI user interface application. Accelerator key (ALT+E) may also be used to perform this task. Exiting the RI must be done from the main mode where the banner logo is displayed. Limiting exiting from the main mode is done so that information will not be lost by premature exits from the other three modes.

5.3.3 Tools Menu

The **Tools** menu includes two items along with their accelerator keys. The tools provide facilities for managing the configuration and log files created by the RI and for setting options controlling the RI's behavior. These are items such as default directories.



(Source: TransCore ITS, LLC, February 2014.)

5. Navigating within the RI

5.3.3.1 Maintenance Dialog

Maintenance opens the dialog shown in [Figure 5-6](#), giving users a choice to delete either configuration or log files that were created in the system. Accelerator key (ALT+M) may also be used to perform this task.

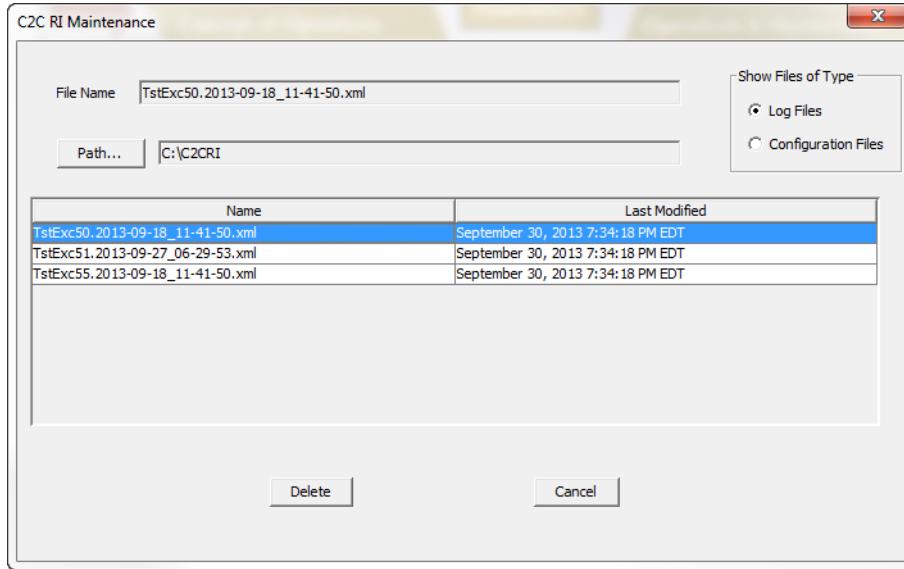


Figure 5-6. Maintenance Dialog
(Source: TransCore ITS, LLC, February 2014.)

5.3.3.2 Options Dialog

Options item opens the dialog shown in [Figure 5-7](#), from which the user can define information for the default settings of different RI parameters. Accelerator key (ALT+P) may also be used to perform this task. All saved options will be applied the next time the C2CRI is started.

The parameters displayed to the user are defined in the c2cri.properties file found in the program installation folder. In order to modify a parameter from this interface, the user must first select the property that needs to be modified. After selection, then changes may be made in the second column (Attribute Value). The C2CRIWizardMode attribute can be set to true or false. Changing the attribute will cause the Wizard panel to appear (true) or not appear (false) the next time the C2CRI is started. The user may click the Save button to store any modification that have been made and continue editing parameters. The user may click the Done button to store the modifications and exit the screen.

HTTPServerListenerHostAddress and HTTPServerPublicationListenerPort, defined in the c2cri.properties file, are on the HTTPServerSettings tab. These are the EC address and port the OC

5. Navigating within the RI

will use when sending subscription publications. They must match the address given to the OC when subscribing. The OC address and port are defined in the WSDL.

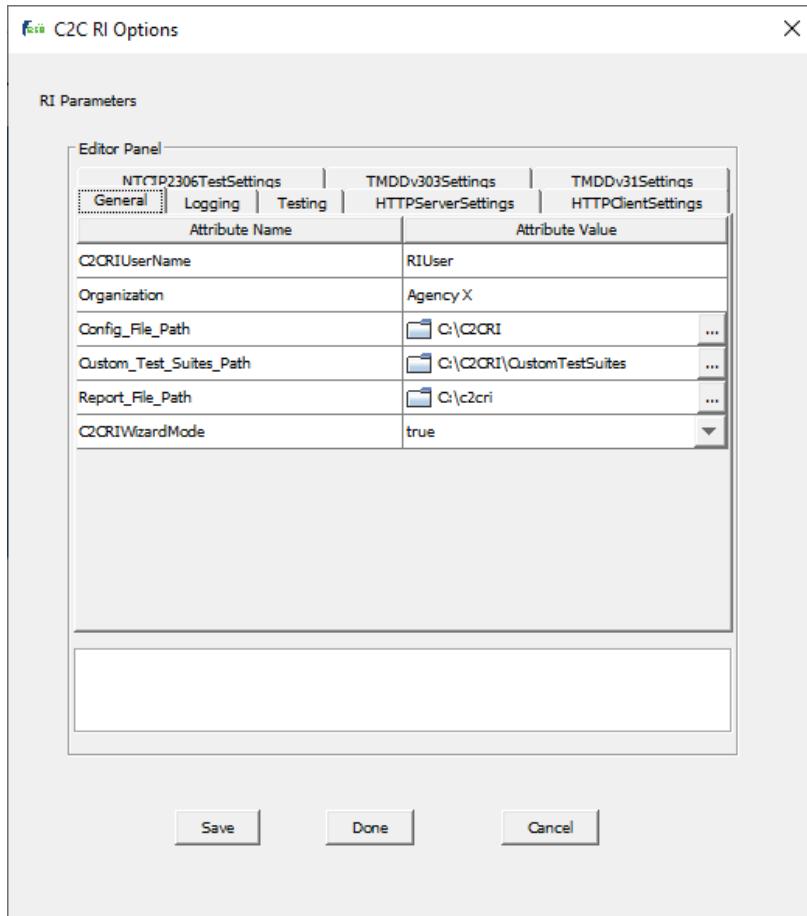


Figure 5-7. RI Parameters Options Dialog – General Tab
(Source: TransCore ITS, LLC, June 2021.)

On the HTTP Client Settings tab (Figure 5-8), the EnsureQuotesOnSOAPActionField attribute name can be set to true or false. Setting it to true will add quotes around the SOAPAction HTTP request header field when it is transmitted, to comply with the SOAP specification (see Section 6 of <http://www.w3.org/TR/2000/NOTE-SOAP-20000508>).

5. Navigating within the RI

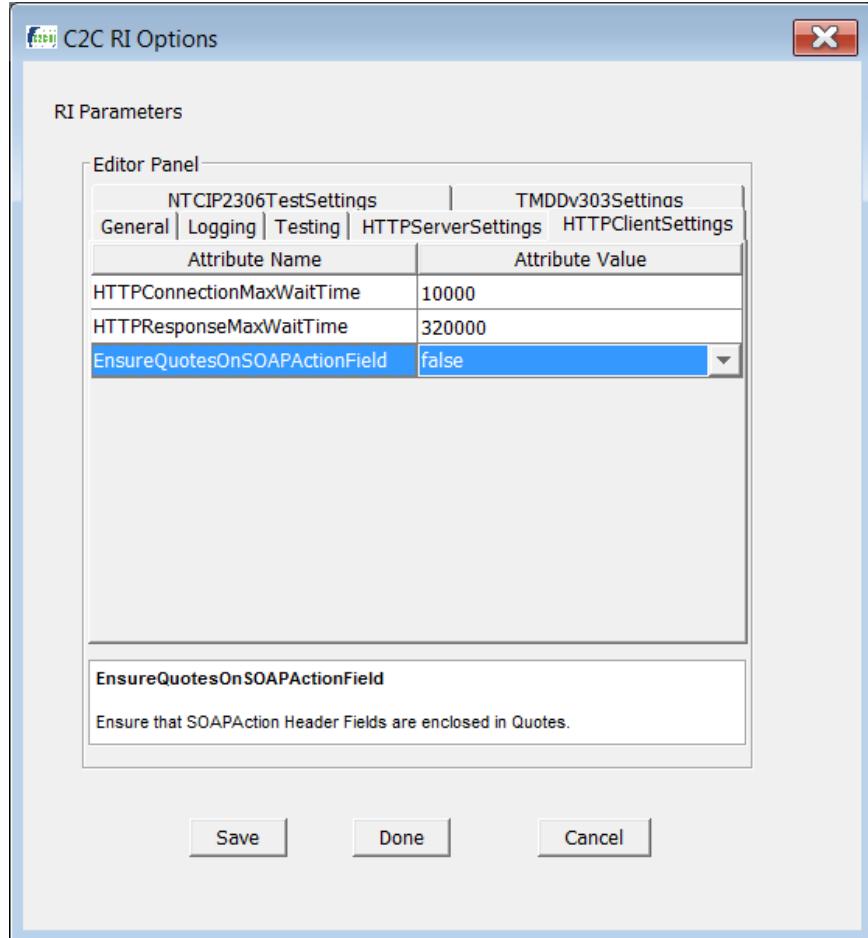


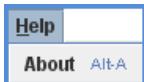
Figure 5-8: RI Parameters Options Dialog – HTTPClientSettings Tab
(Source: TransCore ITS, LLC, December 2017.)

5.3.4 Help Menu

The Help Menu includes one menu item along with its accelerator key.

About opens the dialog shown in [Figure 5-9](#), displaying the version control information for the software. Accelerator key (ALT+A) may also be used to perform this task.

5. Navigating within the RI



(Source: TransCore ITS, LLC, February 2014.)



Figure 5-9. Help About Message
(Source: TransCore ITS, LLC, November 2016.)

5.4 Accelerator Key Navigation

This section describes the various keyboard combinations that you can use to effectively navigate through the C2CRI application. An accelerator (Table 5-1) is a key combination that causes a menu item to be chosen, whether or not it's visible. The user can navigate through each of the application's windows by using the Tab, Space, arrow and Ctrl keys in a manner that is typical for Windows applications.

Each keyboard combination action can be performed directly by using the mouse.

Table 5-1. Shortcut Keys for Application Usage

Menu Item	Accelerator
File	ALT+F
Tools	ALT+T
Help	ALT+H
File->New	ALT+N
File->Open	ALT+O
File->Execute	ALT+X
File->Reports	ALT+R
File->Exit	ALT+E
Tools->Maintenance	ALT+M
Tools->Options	ALT+P
Help->About	ALT+A

These key combinations are not available throughout the RI as they select operational modes. They are available only when appropriate. For example, the file menu accelerator keys are only available when the application is in its main mode when all documents are closed. While in any other mode, that mode must be exited prior to the keys becoming available.

5.5 Guided User Mode

The C2C RI guided user mode, or wizard, (see [Figure 5-1](#)[Figure 5-1](#)) helps a new user navigate through the steps of creating a test configuration, executing a test, and creating and viewing reports. The wizard can be disabled by clicking the “Disable Wizard Mode” checkbox and either disabling the wizard immediately or on restart. Once the wizard is disabled, it can be re-enabled via the Options dialog described in Section 5.3.3.2.

5.5.1 Creating and Editing Test Configuration Files Using the Wizard

Clicking the “Create A Test” button on the wizard opens the Create/Edit Test Configuration dialog shown in Figure 5-10. The user chooses whether to create a new test configuration file or edit an existing file. The wizard walks the user step-by-step through creating or editing a test configuration file, as described in Section 6.0. The wizard progress panel highlights where the user is in the process of creating or editing a test configuration file (see Figure 5-11).

5. Navigating within the RI

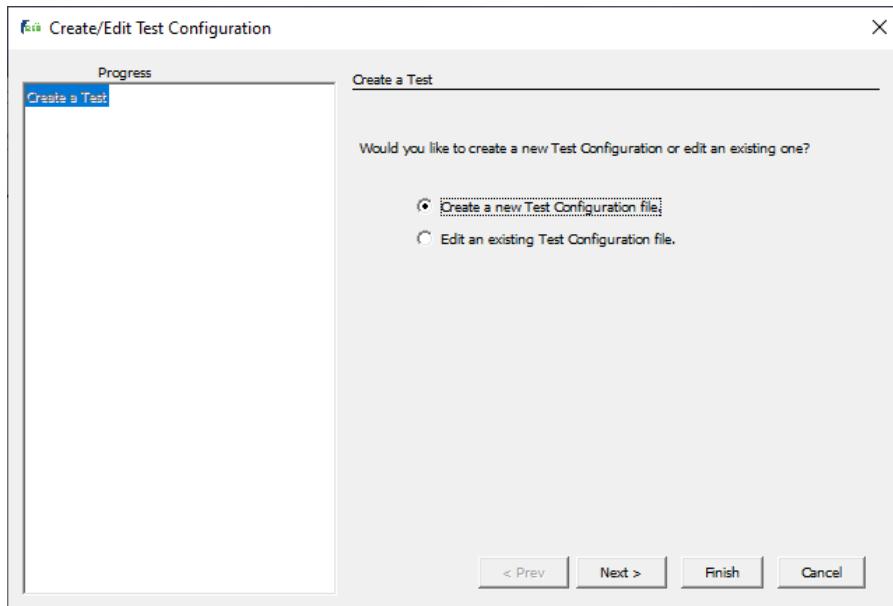


Figure 5-10. Wizard Create/Edit Test Configuration Dialog Initial Screen
(Source: TransCore ITS, LLC, June 2021)

5. Navigating within the RI

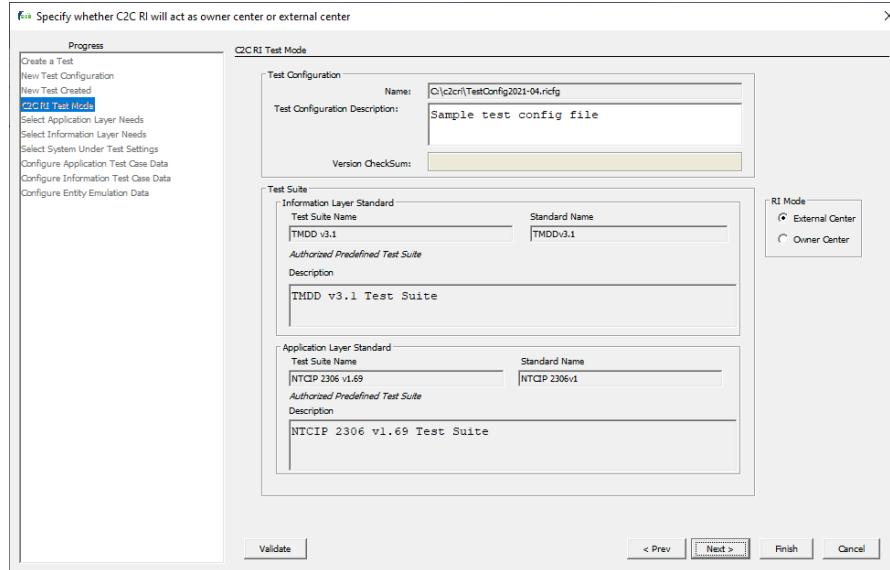


Figure 5-11: Wizard Progress Panel
(Source: TransCore ITS, LLC, June 2021)

5.5.2 Executing a Test Using the Wizard

Clicking the “Execute A Test” button on the wizard opens the RI Test Definition dialog shown in [Figure 5-4](#). From this point, the test execution process is the same as described in Section 7.0.

5.5.3 Creating Reports Using the Wizard

Clicking the “Create a Report” button on the wizard opens the Report Configuration Window shown in [Figure 8-2](#). From this point, the process of creating reports is the same as described in Section 8.0.

6.0 Define a Test Configuration

This section describes how to identify the user needs and requirements for verifying a SUT's conformance with the standards or for reporting needs and requirements without testing conformance. The RI Test Definition mode consists of seven tabbed panels for selection and entry of the data necessary for a complete system description. The contents of each panel are described. The user can complete the panels in any order, but a valid configuration must be completed before the configuration can be used for Test Execution.

To enter the Test Definition mode, create a new test configuration or open an existing configuration file from the **File** menu. The following procedure describes creating a new test configuration starting from the main mode.

7. Select the **File** menu and then select: **New** and the dialog shown in [Figure 5-2](#)[Figure 5-2](#) will be opened.
8. Enter a valid name in the *Test Configuration Name* field. For example, enter **SUT_AnalysisXH_Cfg**
9. Click the **Path** button and browse to the path where you want to save the configuration file.
10. Enter a description for this file in the *Test Configuration Description* field.
11. Select a Test Suite from the drop down list for both the Information Layer Standard and the Application Layer Standard.
12. Click the **OK** button.

Click the (X) button in the top right hand corner of the dialog or click the **Cancel** button to close the dialog at any time prior to Step 6.

Pre-defined test suites are distributed with the RI for various C2C standards. These suites appear in the lists without an asterisk. Each pre-defined test suite contains all user needs (mandatory and optional), requirements, predicates, test cases, test scripts, and other information related to a standard. The RI ensures that pre-defined test suites are authorized versions by inspection of a signature within the test suite's JAR file.

All other test suite JAR files are considered to be custom test suites and therefore are considered to be extensions to the pre-defined test suites. Custom test suites are identified by an asterisk following the test suite name. Custom test suites should also include custom user needs, requirements, predicates, test cases, test scripts, and entity emulation data necessary to verify their project needs. Custom test suites must reference the pre-defined test suite that they extend as described in Section 9.3 Creating Authorized Custom Test Suites.

At successful completion of the dialog the RI will display the Test Definition window as shown in [Figure 6-1](#)[Figure 6-4](#). On the left side of the window is a tree for navigating through the various tabbed

6. Define a Test Configuration

panels. Each panel is also accessible through a tab above the panel. The Configuration Panel lists the standards selected from the test configuration dialog. The only fields that can be modified on this panel are the *Test Configuration Description* notes field and the *RI Mode*.

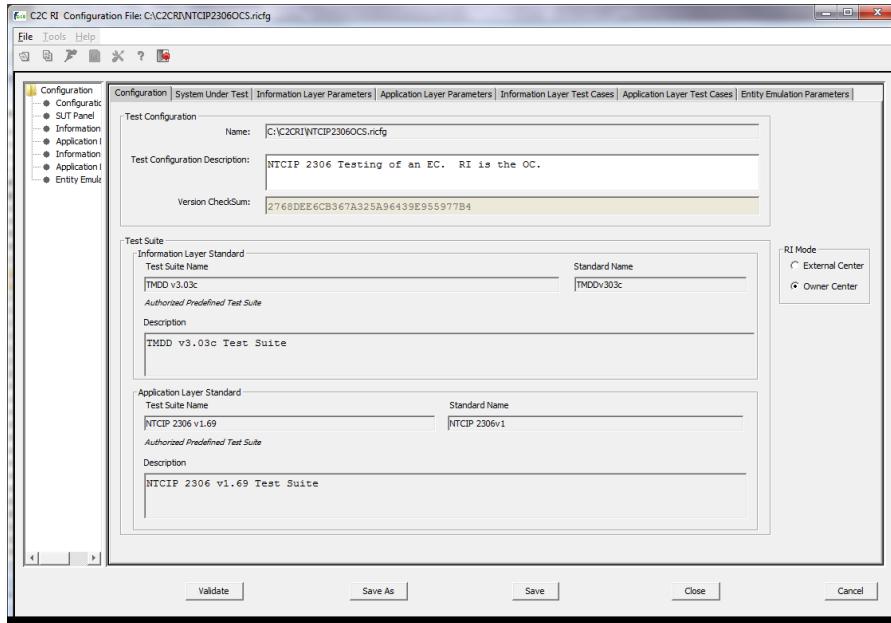


Figure 6-1. Test Definition: Configuration Panel
(Source: TransCore ITS, LLC, July 2016.)

The next panel contains details regarding the System Under Test ([Figure 6-2](#)[Figure 6-2](#)). This data comprises the PIXIT information for the test configuration. These data include the IP address, port number, host name, and Universal Resource Locator (URL) for connecting to the SUT's Web Services – Interface. These entries allow the RI to communicate with the SUT regardless of the SUT being an owner or external center.

Entries for an authorized user account are required on the SUT when secure file transfer protocol (ftp) is utilized. The SUT's administrator must provide these credentials and the RI user must enter them into the fields on this panel.

To use the test configuration only to report needs and requirements, do not edit the default values in the System Under Test panel and select at least one need and requirement on the Application Layer Parameters panel (see [Figure 6-4](#)[Figure 6-4](#)).

6. Define a Test Configuration

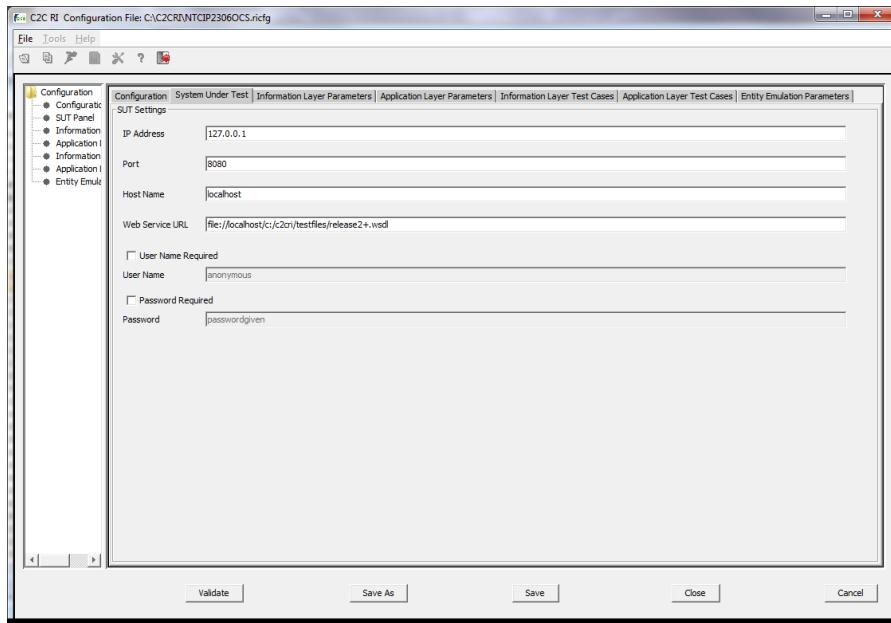


Figure 6-2. Test Definition: System Under Test (SUT) Panel
(Source: TransCore ITS, LLC, July 2016.)

The Information Layer Parameters and the Application Layer Parameters panels behave identically. Each panel has three groups that relate to one another. The *Select Need* group at the top of the panel displays the user needs (UN) defined in the standard's Needs-Requirements Traceability Matrix (NRTM). The middle group titled *Select Requirements* lists the set of requirements defined for the **currently selected** user need in the *Select Need* group. At the bottom, the *Test Parameters* group lists all test items associated with the **currently selected** requirement in the *Select Requirements* group. The important point to remember is that the three tables are related by the current **selection** within the tables.

Figure 6-3 shows these groups for the Information Layer Panel with *User Need* 2.3.3 Need to Provide Information on Organization Centers and Contents currently selected and highlighted. The *Select Requirements* list shows the requirements for *User Need* 2.3.3 and the currently selected requirement is 3.3.3.5.2.1 Organization Name. Because the currently selected requirement doesn't have any test parameters associated with it, the *Test Parameters* list is blank.

6. Define a Test Configuration

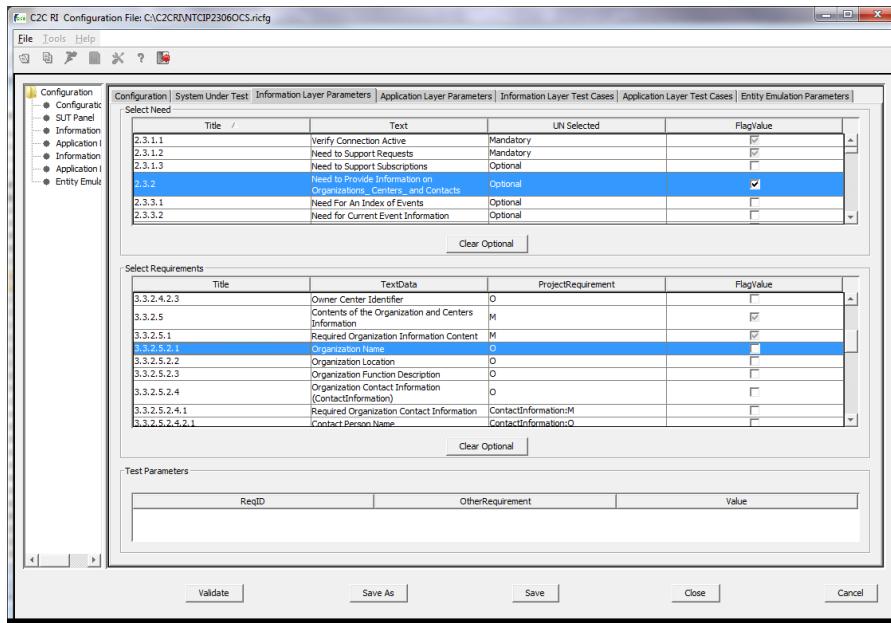


Figure 6-3. Test Definition: Information Layer Parameters Panel
(Source: TransCore ITS, LLC, July 2016.)

This relationship is important to remember when navigating through these three lists. Scrolling through the lists will temporarily cause the selected items to move out of view but the association between the windows will remain intact. Therefore, items unrelated to the currently selected items will appear in the list. This ability to observe unrelated items may cause the unwary user concern while navigating through the various lists. This concept should be kept in mind while navigating the list to avoid declaring the data display to be incorrect.

The Select Need group displays user needs that are designated mandatory by the standard with ‘Mandatory’ in the *UN Selected* column of the table. These mandatory requirements also have check boxes in the *FlagValue* column that are checked and disabled (i.e., greyed out) to indicate their mandatory status. The figure contains Title 2.3.1.1 Verify Connection Active showing the need as Mandatory. The group also displays optional needs, and these are identified by Optional in the *UN Selected* column. The check boxes in the *FlagValue* column show whether the need has been selected (i.e., checked) or is not a need for the configuration (i.e., unchecked).

The mandatory set of user needs consists of six needs. Four of these needs involve the basic verification of communication connections, simple request-response messaging, error responses, and handling of null values. The other two needs require support for distributing the link-node model of the transportation network.

Requirements, like needs, can be either Mandatory or Optional with optional needs indicated by an “O” character in the *Project Requirement* column. Some of the optional requirements are dependent

6. Define a Test Configuration

on other requirements being selected and are referred to as predicates. When predicates exist, they are shown in the *Project Requirement* column as the predicate name followed by a colon and the option character. This can be seen on the 3.3.3.5.2.5.1 Contact Person Name row in the figure. The predicate is Contact Information, and TMDD Volume I Section 5.4.2 describes the requirement that must be selected to include Contact Information (3.3.5.2.4).

The other *Project Requirement* notation that appears identifies option groups. These appear as a numeric group number following the option character and a range number in parentheses. All options within a group have an identical group number. The range number indicates how many of the options are required within the group. An example appears in [Figure 6-4. Test Definition: Application Layer Parameters Panel](#) for the requirements for 2. d and 2. e that are in group 2 as shown by the O.2 and the number of items to be selected is 2 as shown by the (2) notation.

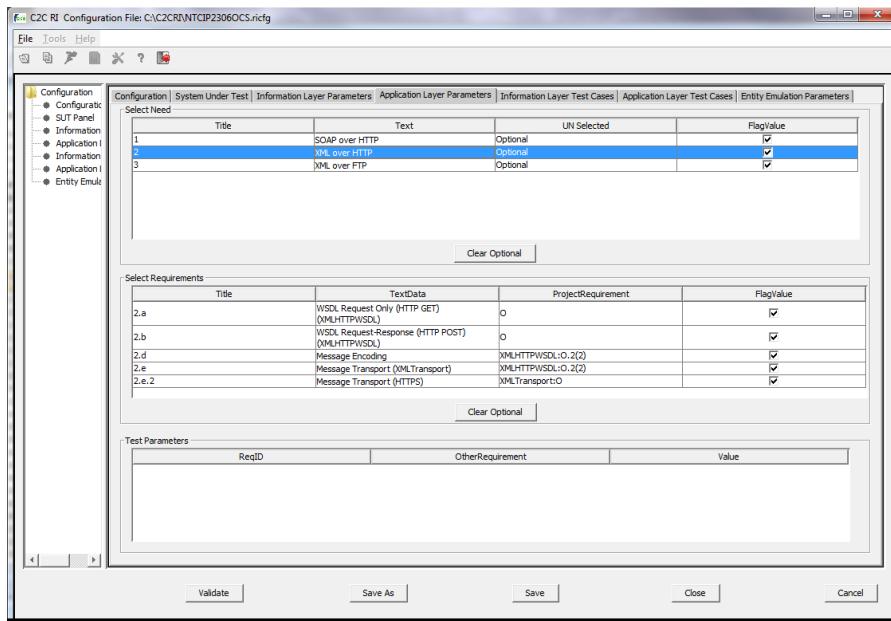


Figure 6-4. Test Definition: Application Layer Parameters Panel
(Source: TransCore ITS, LLC, July 2016.)

The Information Layer Test Cases and the Application Layer Test Cases panels list the test cases made available via the current selection of test suites on the System Under Test Panel. If the user has defined a test case file that should override the default test case definition, then the path and file name for this test case file should be entered in the source column. If the file name entered does not match an existing file on the host system, the source field will revert back to "default."

The individual test cases related to the selected center mode and applicable needs/requirements selections will be available in the Test Execution mode of the RI.

6. Define a Test Configuration

The following figures ([Figure 6-5](#)[Figure 6-5](#) and [Figure 6-6](#)[Figure 6-6](#)) contain a sample of each test case panel.

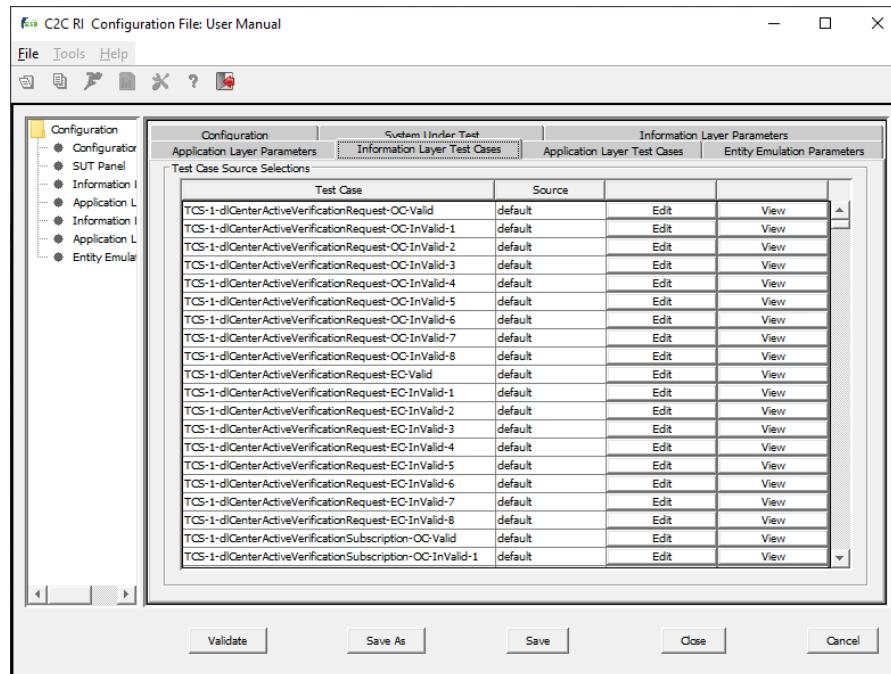


Figure 6-5. Test Definition: Information Layer Test Cases Panel
(Source: TransCore ITS, LLC, June 2021.)

6. Define a Test Configuration

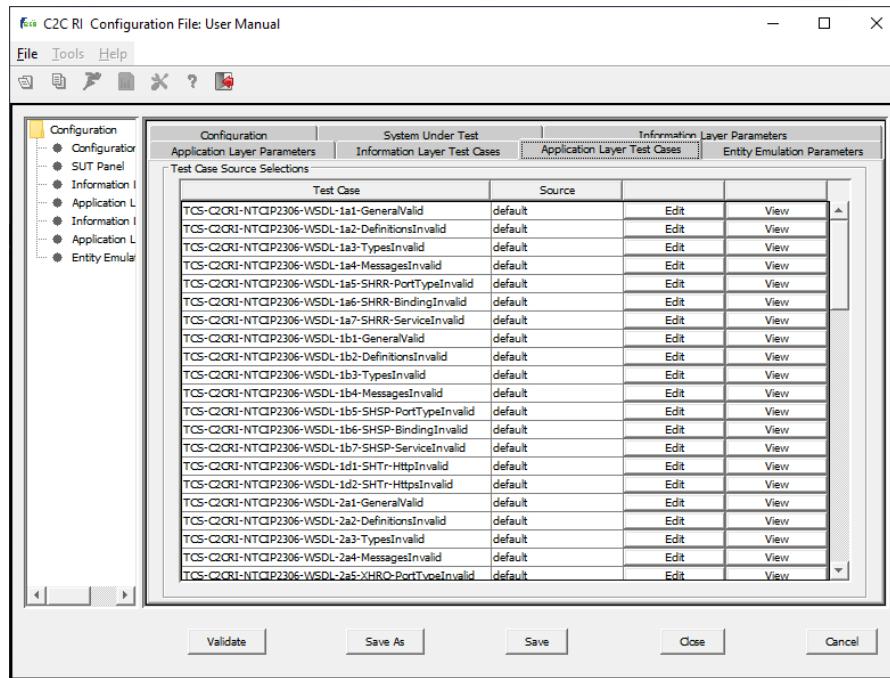


Figure 6-6. Test Definition: Application Layer Test Cases Panel

(Source: TransCore ITS, LLC, June 2021.)

The Entity Emulation Parameters panel (Figure 6-7) shows the test cases made available via the current selection of test suites on the System Under Test Panel. Each element uses TMDD conventions and represents the message that an OC will send in response to an EC request. Each message includes information that the OC has about the entity data type. If the user has defined an entity message that should override the default test case message, select the test case and click **Update** to browse to the desired text file. Select the file, click **Open**, click **Load** (see Figure 6-8), and click **Cancel** and confirm that you want to overwrite the configuration file to close the dialog. The **Source** will then show "Updated." Select an entity data name from the left column and click **View** to open the Entity Data Viewer for the selected entity (Figure 6-9) which shows the message contents.

The *Command Queue Length* is set by the user during testing and identifies how deep the queue must get before commands will be run (range 0-100). If 0, each command will be executed immediately. The queue length is entity specific.

6. Define a Test Configuration

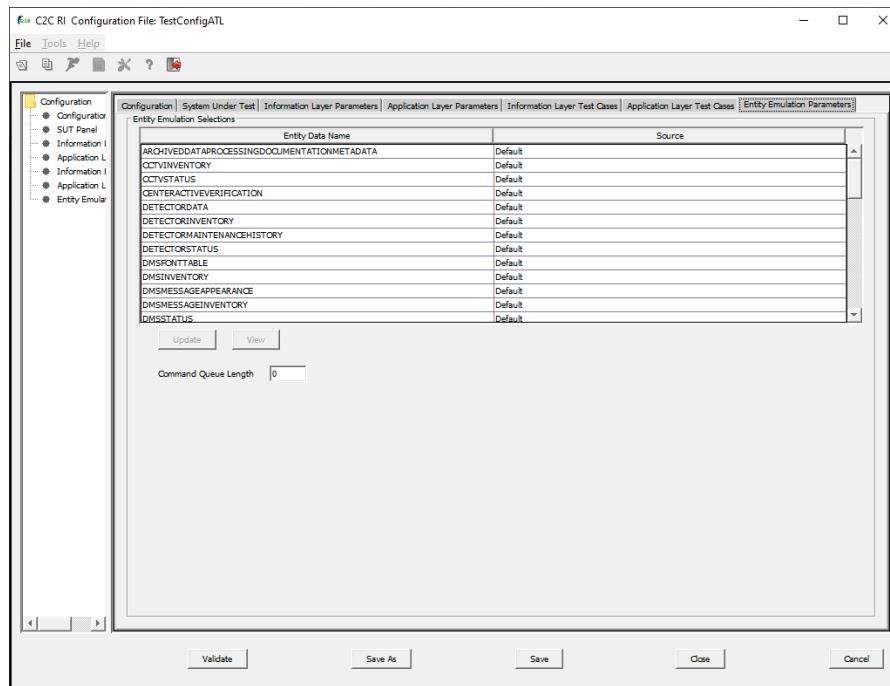


Figure 6-7: Test Definition: Entity Emulation Parameters Panel
(Source: TransCore ITS, LLC, June 2021.)

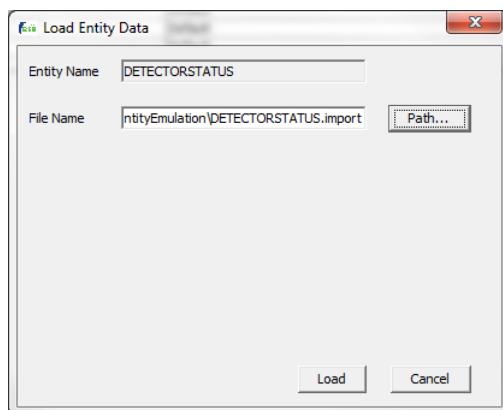


Figure 6-8: Load Entity Data Dialog
(Source: TransCore ITS, LLC, July 2016.)

6. Define a Test Configuration

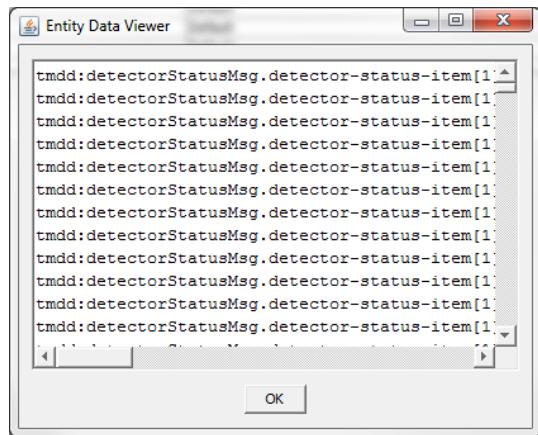


Figure 6-9: Entity Data Viewer
(Source: TransCore ITS, LLC, July 2016)

7.0 Execute a Test Configuration

1. Select the **File** menu and select: **Execute** and the dialog shown in [Figure 5-4](#)[Figure 5-4](#) will be opened.
2. Enter a valid name for the test in the Test Name field.
3. Enter descriptive notes for the test in the Test Notes field.
4. Click the **Path** button to change the current directory to one where existing Test Configurations are saved (optional). When the **Browse** button is clicked, a dialog similar to the one shown in [Figure 5-3](#)[Figure 5-3](#) will be displayed.
5. Select a Test Configuration from the configuration list in the dialog box.
6. Click the **Accept** button and the dialog shown in [Figure 7-1](#)[Figure 7-4](#) will be displayed.

Click the (X) button in the top right hand corner of the window or click the **Cancel** button to close the dialog at any time prior to Step 6.

7. Execute a Test Configuration

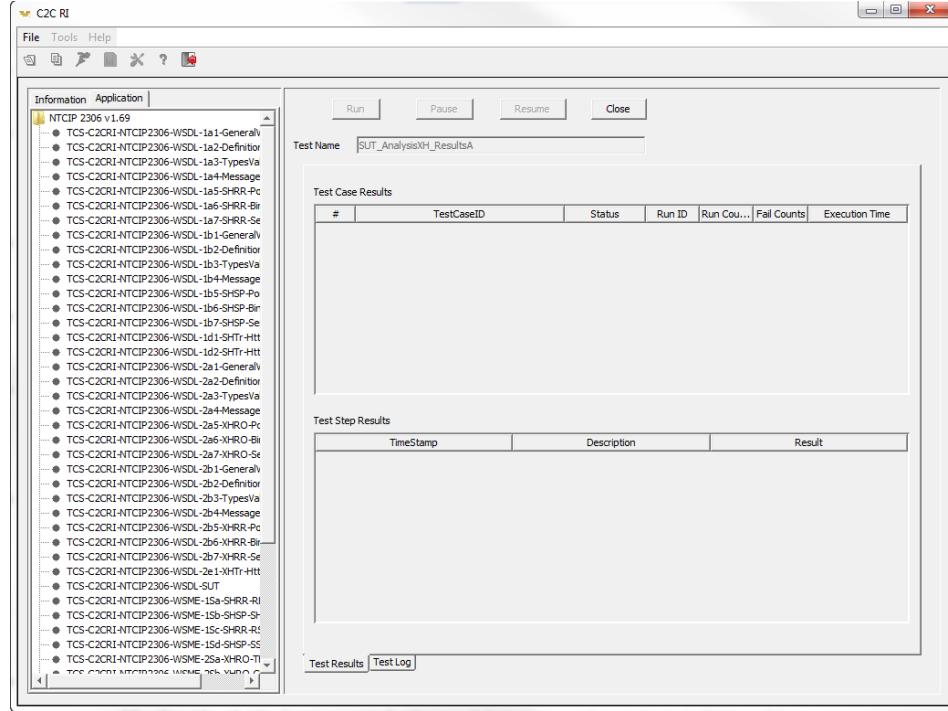


Figure 7-1. RI Test Execution Window: Pending Execution

(Source: TransCore ITS, LLC, February 2014.)

Note: Click the **Pause** button to temporarily stop a running test, and click the **Resume** button to continue executing the test. Click the **Terminate** button to stop an executing test.

- Click the **Run** button to execute the tests selected. The **Terminate** button will display **Close** when the test execution completes as shown in [Figure 7-2](#)[Figure 7-2](#). The *Run ID* field indicates the iteration of the particular test case in the current test. If the user runs the same test case multiple times without closing the test execution window, the subsequent test case results will appear on separate lines with the *Run ID* count incremented. The *Run Counts* field indicates how many times the test case has been run in the current test, and it will be the same count for each test instance. The *Fail Counts* field indicates how many times the test case has failed in the current test, and it will be the same count for each test instance.

Click the **Close** button to close the dialog and also to proceed in performing other RI functions in the user interface.

8. Message Content Verification

When running a TMDD valid information level test, the dialog shown in Figure 7-3 opens when the test is running. The purpose of this dialog is to allow the user to verify the message value contents before proceeding. Verification is optional. The values shown in the Value column are not subject to standards, so the user should check them for reasonableness or against known values (e.g., the

7. Execute a Test Configuration

organization-requesting value). If the user determines that the values are unacceptable, clicking **Fail** will terminate the test. If the user determines that the values are acceptable, it is possible to create value tests for subsequent runs of the same test (see step 9 below). Clicking **Pass** will allow the test to continue.

9. Create Value Tests

It is possible to create value tests for subsequent runs of the same test, testing against selected values that the user determines are acceptable. To create a test, click on the row for the element to be tested. Using *CTRL+click*, you can select multiple rows and create a test for several elements, as long as they are of the same type (e.g., DMS sign type elements). Only Type 1 value tests are supported at this time, and that field is not editable. Click **Create** to create a value test for the selected elements. NOTE: If a row is highlighted yellow, it is an optional requirement element, traced to a TMDD optional element that was not selected in the test configuration dialog. Creating a value test for an optional element may result in future test failures, because inclusion of that element is not required.

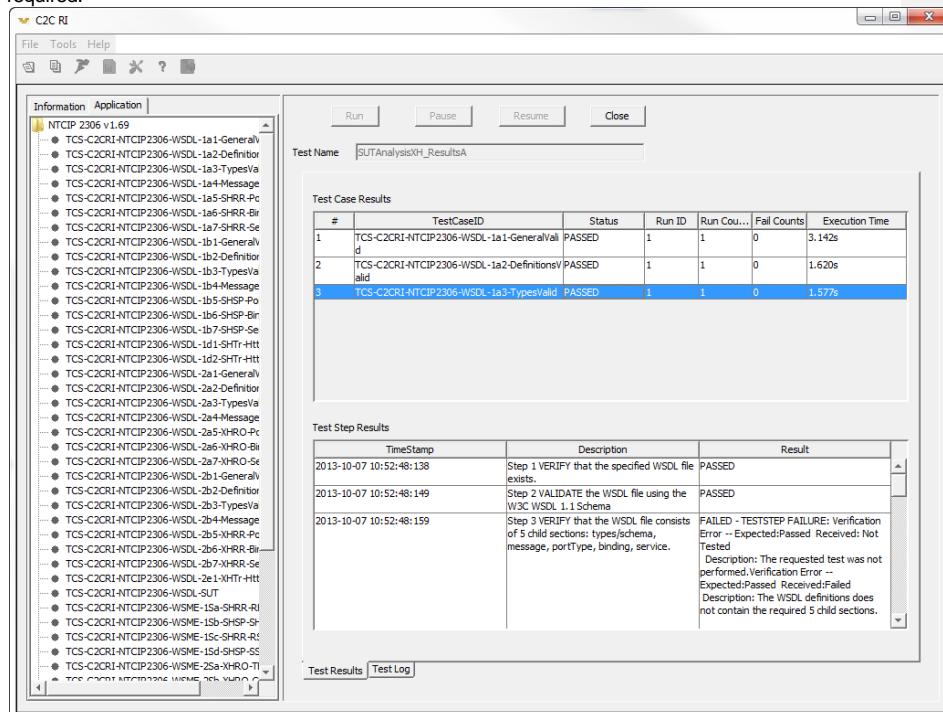


Figure 7-2. RI Test Execution Window: Results
(Source: TransCore ITS, LLC, February 2014.)

7. Execute a Test Configuration

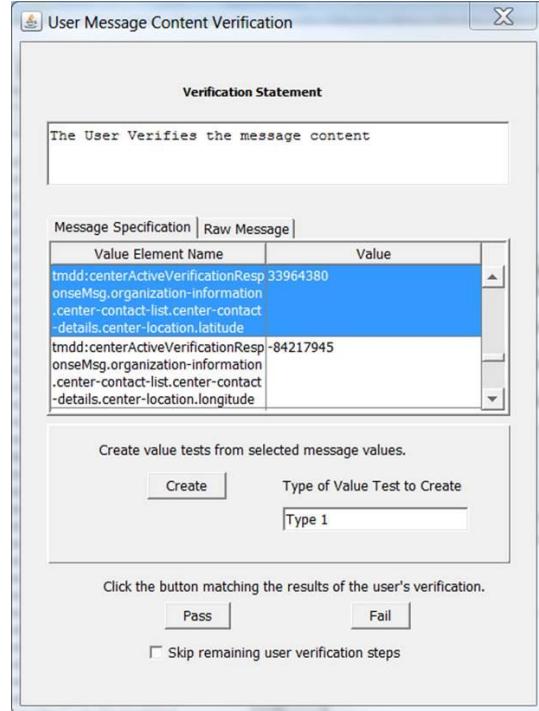


Figure 7-3: User Content Verification Window
(Source: TransCore ITS, LLC, June 2016.)

The Test Execution (Logs) show the status of the individual test steps and the script log describes each operation carried out by the RI test cases ([Figure 7-4](#)[Figure 7-4](#)).

7. Execute a Test Configuration

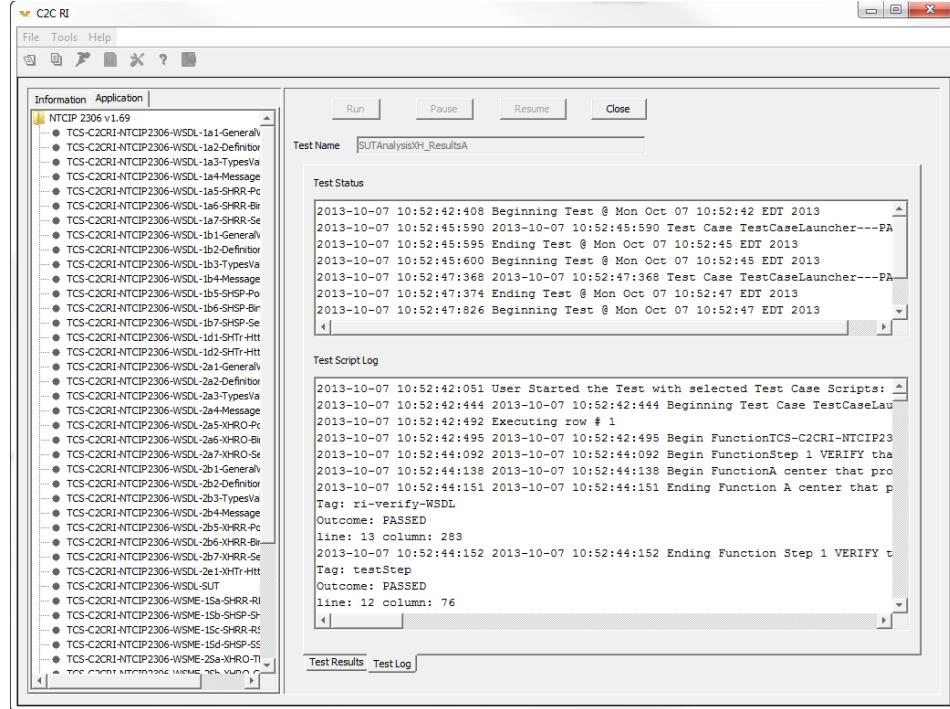


Figure 7-4. RI Test Execution Window: Log
(Source: TransCore ITS, LLC, February 2014.)

8.0 Report Test Results

This section describes how the user can generate different kinds of configuration or log reports to ascertain useful information. Configuration report types include Configuration File Details, Test Procedures, Test Cases, and Test Scripts. Test report types include Conformance/Compliance, Test Case Summary, Test Case Details, Message Summary, Message Detail, and Script Log. In order to generate any of the configuration reports the user must select a Test Configuration file from which the report information will be extracted. Likewise, to generate any of the log reports a test execution log file must be supplied.

8.1 How to Navigate the C2C RI Report Viewer Window

The C2C RI Report Viewer Window has a toolbar that is used to easily navigate the report window. These tools are available for every document generated by the RI and opened in the Report Viewer. The toolbar is shown in [Figure 8-1](#). Each icon has a tooltip which provides basic information about the feature when you hover over it.



Figure 8-1. C2C RI Report Viewer Toolbar
(Source: TransCore ITS, LLC, February 2014.)

From left to right these icons are: Save As..., Print Document, Search Document, Show/Hide Utility Panel, First Page, Previous Page, Current Page number (text field) Next Page, Last page, Zoom Out, Zoom (text field), Zoom In, Actual Size, Fit in Window, Fit Width, Rotate Left, Rotate Right, Pan Tool, text Select Tool, Zoom In Tool, Zoom Out Tool, Select Tool and Link Annotation Tool.

8.2 How to Generate Test Configuration Reports

1. Select the **File** menu and then select: **Reports** and the dialog shown in [Figure 8-2](#) will be opened. **Cancel** will close the Test Reporting dialog.

8. Report Test Results

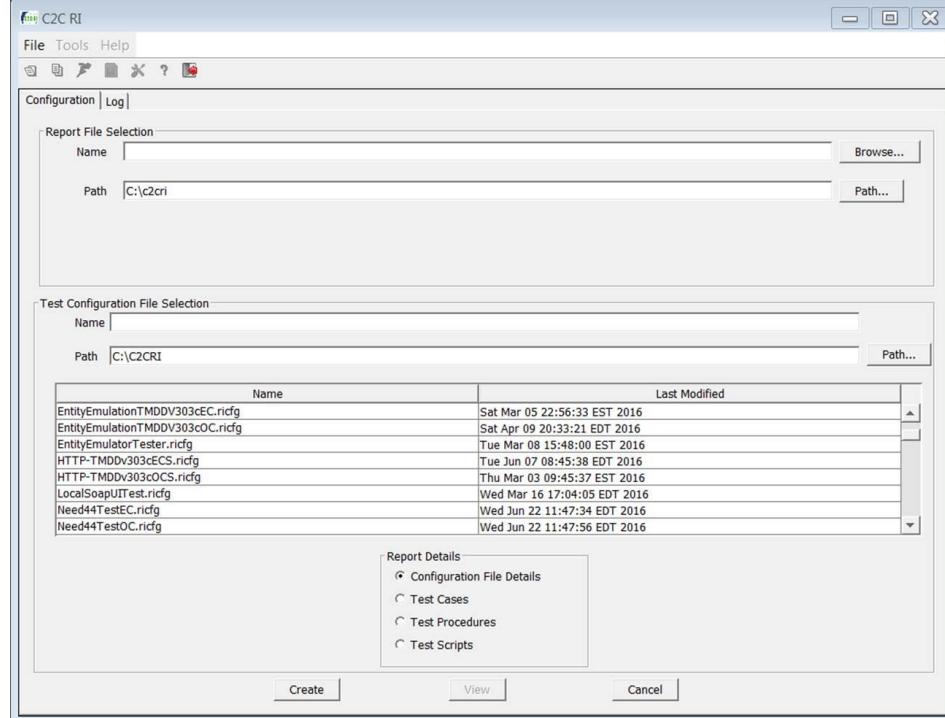


Figure 8-2. Report Configuration Window

(Source: TransCore ITS, LLC, November 2016.)

2. Select the 'Configuration' tab.
3. Enter a valid name for the test report in the Name field inside the Report File Selection section of the screen.
4. Click the **Path** button to change the current directory to one where the report file should be stored (optional). When the **Browse** button is clicked, a dialog similar to the one shown in [Figure 5-3](#) will be displayed. This button can be used to select an existing file name for the report.
5. Enter a different path for the file in the Path field (optional). When the **Path** button is clicked, a dialog similar to the one shown in [Figure 5-3](#) will be displayed.
6. Select a Test Configuration File from the Test Configuration File Selection section of the screen and then the test reporting screen will look similar to [Figure 8-3](#).

8. Report Test Results

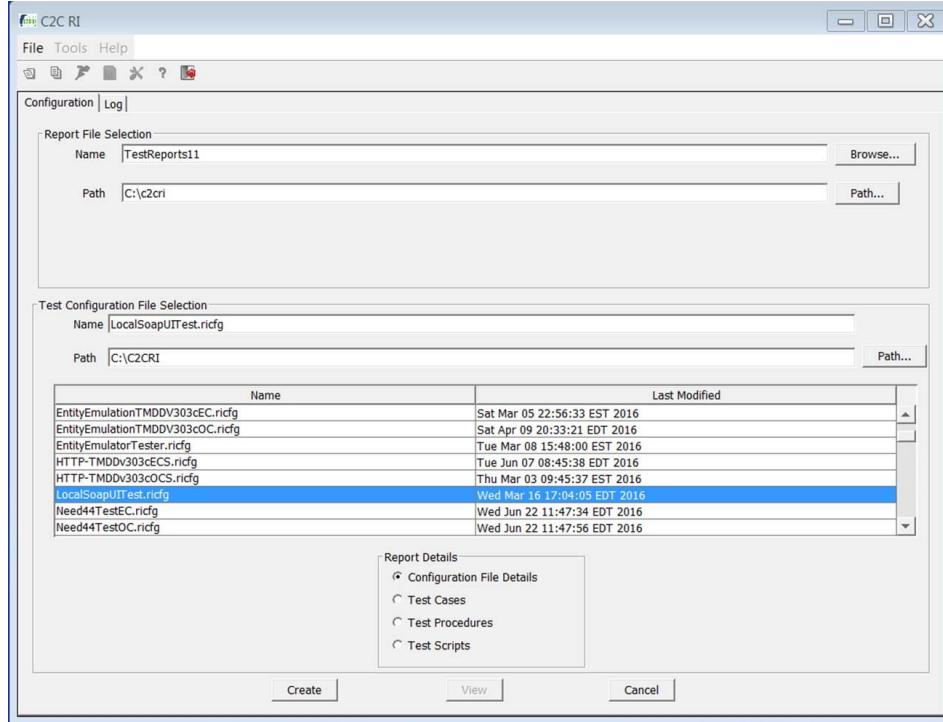


Figure 8-3. Test Configuration Report with Completed Inputs
(Source: TransCore ITS, LLC, November 2016.)

7. Select one of the Report Details type. The user has the option to select any of these reports: Configuration File Details, Test Scripts, Test Cases, or Test Procedures.
8. Click the **Create** button and the processing dialog similar to the one shown in [Figure 8-4](#) will be displayed.

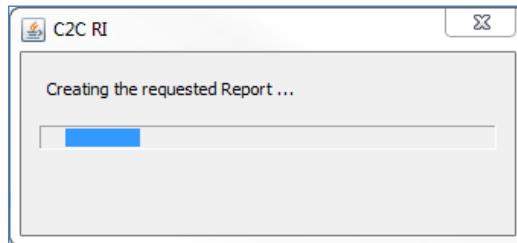


Figure 8-4. Report Request Progress Window

8. Report Test Results

(Source: TransCore ITS, LLC, February 2014.)

Once the requested report is generated the system will display this confirmation dialog in [Figure 8-5](#) to the user.

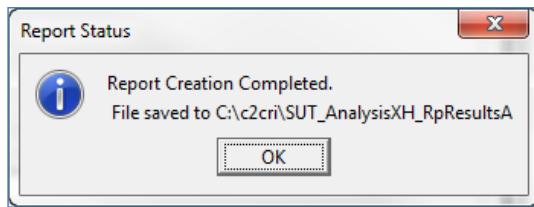


Figure 8-5. Test Configuration Report Completion

(Source: TransCore ITS, LLC, February 2014.)

9. Click the **OK** button on the Report Status dialog.
10. Click the **View** button on the Test Reporting screen.

8.2.1 Configuration File Details Report

If a 'Configuration File Details' report detail type was selected, then the first page of the report will look similar to the information presented in [Figure 8-6](#). This report is useful for reporting a system's needs and requirements without testing for standards compliance.

8. Report Test Results

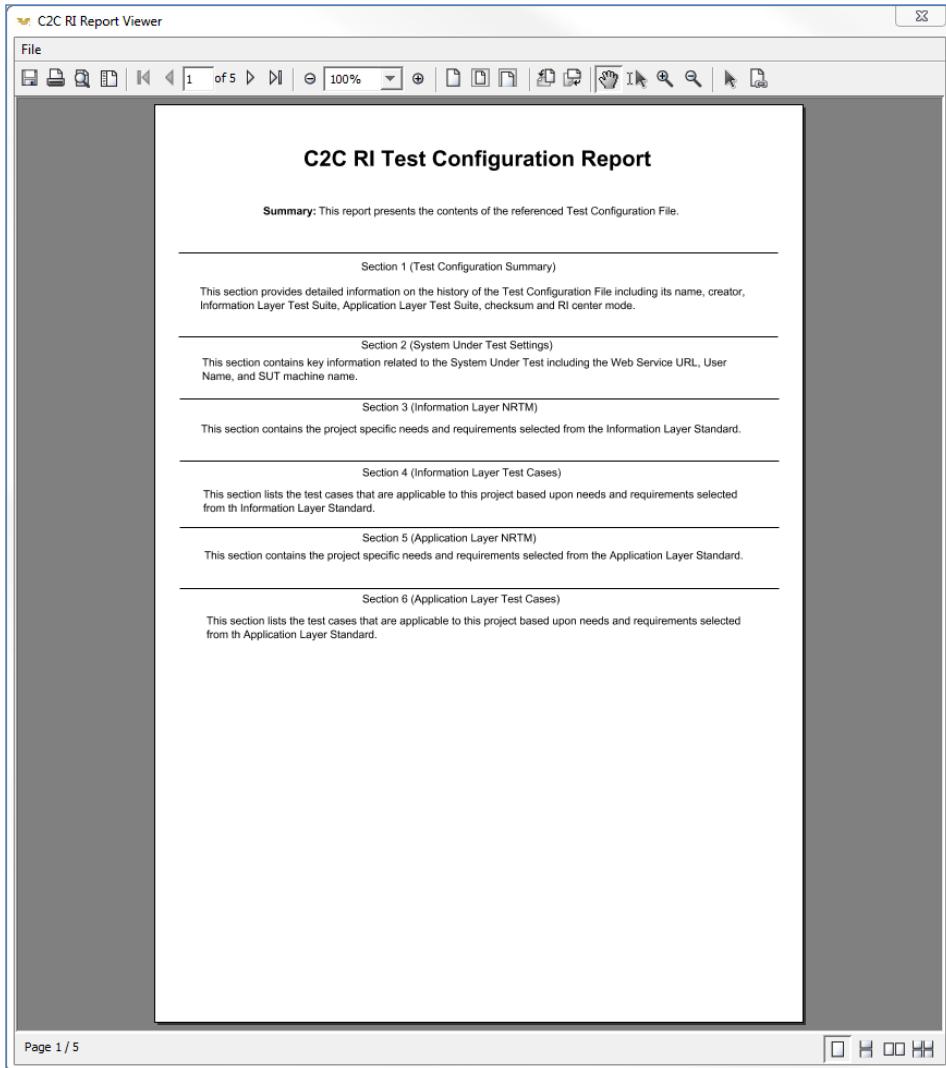


Figure 8-6. C2C RI Test Configuration Report
(Source: TransCore ITS, LLC, February 2014.)

Click the **next Page** icon on the C2C RI Report Viewer to move to the second page of the report and the information that user should be able to view will look similar to the information presented in [Figure 8-7](#)[Figure 8-7](#).

8. Report Test Results

The screenshot shows the C2C RI Report Viewer interface. The main window displays the 'C2C RI Test Configuration Report' from 10/04/2013. It includes sections for configuration file details, SUT settings, and requirement details.

Configuration File Details:

Configuration File Name	C:\C2CR\ISUT_AnalysisXH_cfg.ricfg
Configuration File Creator	tomusikl
Configuration File Creation Date	2013-10-03 17:20:20
Configuration File Description	XML over HTTP Request-Response
Configuration File CheckSum	A8F677181809E1215735FFA28CA08C5A
Selected Test Suite Name	TMDD v3.03a
Selected Test Suite Description	TMDD v3.03a Test Suite

SUT Settings:

IP Address	127.0.0.1
Port	8080
Host Name	localhost
Web Service URL	file:///localhost/c:/c2cri/testfiles/release2+.wsdl
User Name Required	false
User Name	anonymous
Password Required	false
Password	passwordgiven

Information Layer Standard NRTM:

NeedID	Need Text	Need Type
2.3.1.1	Verify Connection Active	Mandatory
Requirement ID	Requirement Text	Other Requirements
3.3.1.1.1	Send Center Active Verification Upon Request	The owner center shall respond within ____ (100 ms - 1 hour; Default = 1 minute) after receiving the request.
3.3.1.1.4	Contents of the Center Active Verification Request	
3.3.1.1.4.1	Required Center Active Verification Request Content	
3.3.1.1.5	Contents of the Center Active Information	
3.3.1.1.5.1	Required Center Active Information	
3.3.1.4.1	Contents of the Error Report	
3.3.1.4.1.1	Required Error Report Contents	
2.3.1.2	Need to Support Requests	Mandatory
Requirement ID	Requirement Text	Other Requirements

04/10/2013 10:11 AM Page 2 of 5

Figure 8-7. C2C RI Test Configuration Report Details
(Source: TransCore ITS, LLC, February 2014.)

8.2.2 Test Script Report

If a ‘Test Scripts’ report detail type was selected, then the first page of the report will look similar to the information presented in [Figure 8-8](#)[Figure 8-8](#).

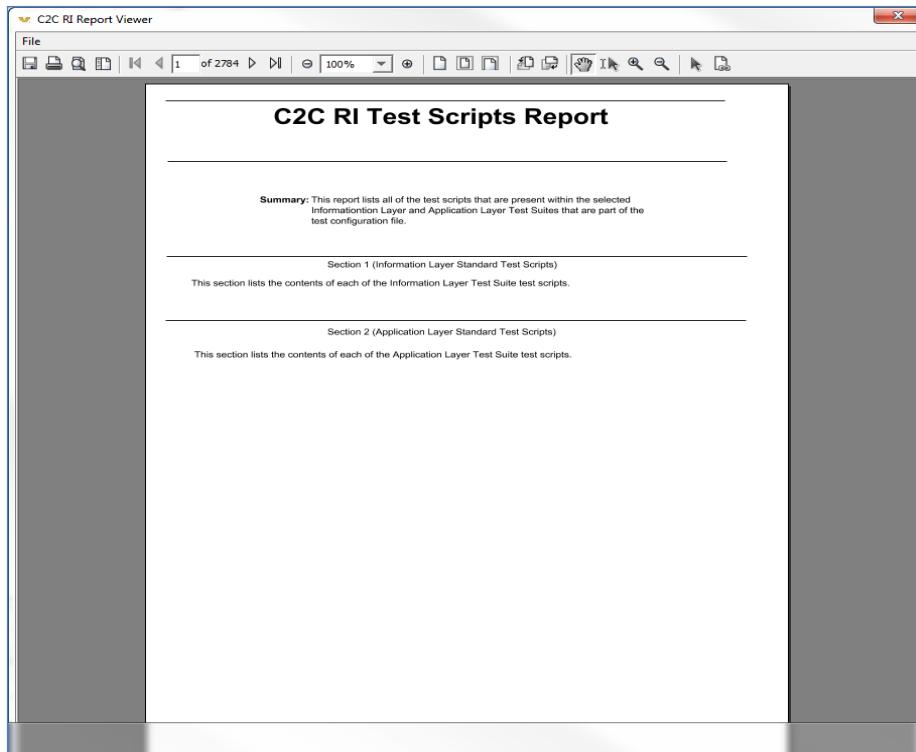


Figure 8-8. C2C RI Test Scripts Report
(Source: TransCore ITS, LLC, February 2014.)

Click the **next Page** icon (any number of times depending on the page number to view) on the C2C RI Report Viewer to move to another page of the Test Scripts Report and the information that user should be able to view will look similar to the information presented in [Figure 8-9](#)[Figure 8-9](#).

8. Report Test Results

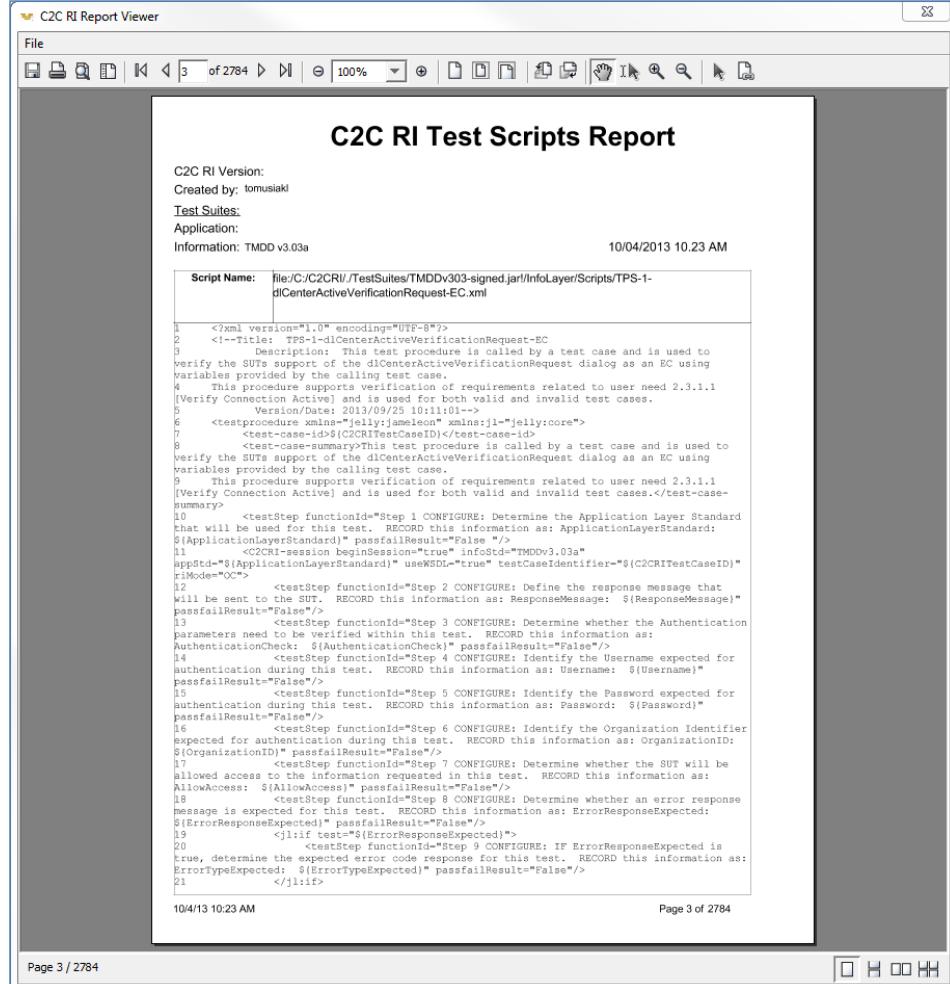


Figure 8-9. C2C RI Test Scripts Report Details

(Source: TransCore ITS, LLC, February 2014.)

Select the **File** menu and then select: **Exit** from inside the C2C RI Report Viewer to close the generated report at any time. Clicking the **(X)** button in the top right hand corner of the window can also be used to accomplish this task.

8.2.3 Test Case Report

If a ‘Test Case’ report detail type was selected, then the first page of the report will look similar to the information presented in Figure 8-10. This report lists all of the test cases that are present within the selected information layer and application layer test suites that are part of the test configuration file.

1. TCS-1-dlCenterActiveVerificationRequest-OC-InValid-1	
1.1 Test Case Specification identifier	
Identifier	Description
TCS-1-dlCenterActiveVerificationRequest-OC-InValid-1	This test case is used to verify the SUTs support of the dlCenterActiveVerificationRequest dialog as an OC using the variable values specified by the Test Plan. This test case supports verification of requirements related to user need 2.3.1.1 [Verify Connection Active]
1.2 Test Items	
3.3.1.1.1 3.3.1.4.1	
1.3 Input Specifications	
Inputs	Procedures
Refer to the Test Case Data Variable Table in the appendix for the test case input parameters.	TPS-1-dlCenterActiveVerificationRequest-OC
1.4 Output Specifications	
Outputs	Procedures
Each input execution shall generate an RI Test Result Status of Passed or Failed associated with the matching expected result shown in the Test Case Data Variable Table in the appendix.	
1.5 Environmental Needs	
1.5.1 Hardware	
See C2C RI SRS	
1.5.2 Software	
See C2C RI SRS	
1.5.3 Other	
None	
1.6 Special Procedural Requirements	
None	
Page 4 of 5788	

Figure 8-10: Sample Test Case Report
(Source: TransCore ITS, LLC, November 2016.)

8.2.4 Test Procedure Report

If a ‘Test Procedure’ report detail type was selected, then the first page of the report will look similar to the information presented in Figure 8-11. This report lists all of the test procedures that are present within the selected information layer and application layer test suites that are part of the test configuration file.

C2C RI Test Procedures Report										
C2C RI Version:	Pre-Release-v2-16									
Created by:	Jim:hardik									
Test Suites:										
Information:	TMDDv3.03c									
Application:	NTCIP 2306v1									
	October 31, 2016									
Configuration File Name:	C:\C2CR\ITRANSCOM Test 2.ricfg									
Configuration File Creator:	Jim:hardik									
Configuration File Creation:	2016-05-31 12:39:14									
Configuration File Description:	Expanded NRTM and Connectivity test.									
Configuration File CheckSum:	0D6C292B22B1E48266D8919F21D04CE7									
SUT Center Mode:	SUT in External Center Mode									
Information Layer Standard Test Procedures										
<p>1. Test Procedures for TPS-1-dlCenterActiveVerificationRequest-OC</p> <p>1.1 TPS-1-dlCenterActiveVerificationRequest-OC</p> <table border="1"> <thead> <tr> <th>Test Procedure</th><th>TPS-1-dlCenterActiveVerificationRequest-OC</th><th>TPS-1-dlCenterActiveVerificationRequest-OC</th></tr> </thead> <tbody> <tr> <td>Description:</td><td>This test procedure is called by a test case and is used to verify the SUTs support of the dlCenterActiveVerificationRequest dialog as an OC using variables provided by the calling test case. This procedure supports verification of requirements related to user need 2.3.1.1 [Verify Connection Active] and is used for both valid and invalid test cases.</td><td></td></tr> <tr> <td>Requirement(s):</td><td>3.3.1.1.1 3.3.1.1.5 3.3.1.1.5.1 3.3.1.1.5.2.1 3.3.1.4.1 3.3.1.4.1.1 3.3.1.4.1.2.1</td><td></td></tr> </tbody> </table>		Test Procedure	TPS-1-dlCenterActiveVerificationRequest-OC	TPS-1-dlCenterActiveVerificationRequest-OC	Description:	This test procedure is called by a test case and is used to verify the SUTs support of the dlCenterActiveVerificationRequest dialog as an OC using variables provided by the calling test case. This procedure supports verification of requirements related to user need 2.3.1.1 [Verify Connection Active] and is used for both valid and invalid test cases.		Requirement(s):	3.3.1.1.1 3.3.1.1.5 3.3.1.1.5.1 3.3.1.1.5.2.1 3.3.1.4.1 3.3.1.4.1.1 3.3.1.4.1.2.1	
Test Procedure	TPS-1-dlCenterActiveVerificationRequest-OC	TPS-1-dlCenterActiveVerificationRequest-OC								
Description:	This test procedure is called by a test case and is used to verify the SUTs support of the dlCenterActiveVerificationRequest dialog as an OC using variables provided by the calling test case. This procedure supports verification of requirements related to user need 2.3.1.1 [Verify Connection Active] and is used for both valid and invalid test cases.									
Requirement(s):	3.3.1.1.1 3.3.1.1.5 3.3.1.1.5.1 3.3.1.1.5.2.1 3.3.1.4.1 3.3.1.4.1.1 3.3.1.4.1.2.1									
Page 2 of 128										

Figure 8-11: Sample Test Procedure Report

(Source: TransCore ITS, LLC, November 2016.)

8.3 How to Generate Test Log Reports

Select the File menu and then select: Reports and the dialog shown in [Figure 8-13](#)[Figure 8-12](#) will be opened.

Select the 'Log' tab and the dialog shown in [Figure 8-13](#)[Figure 8-13](#) will be opened.

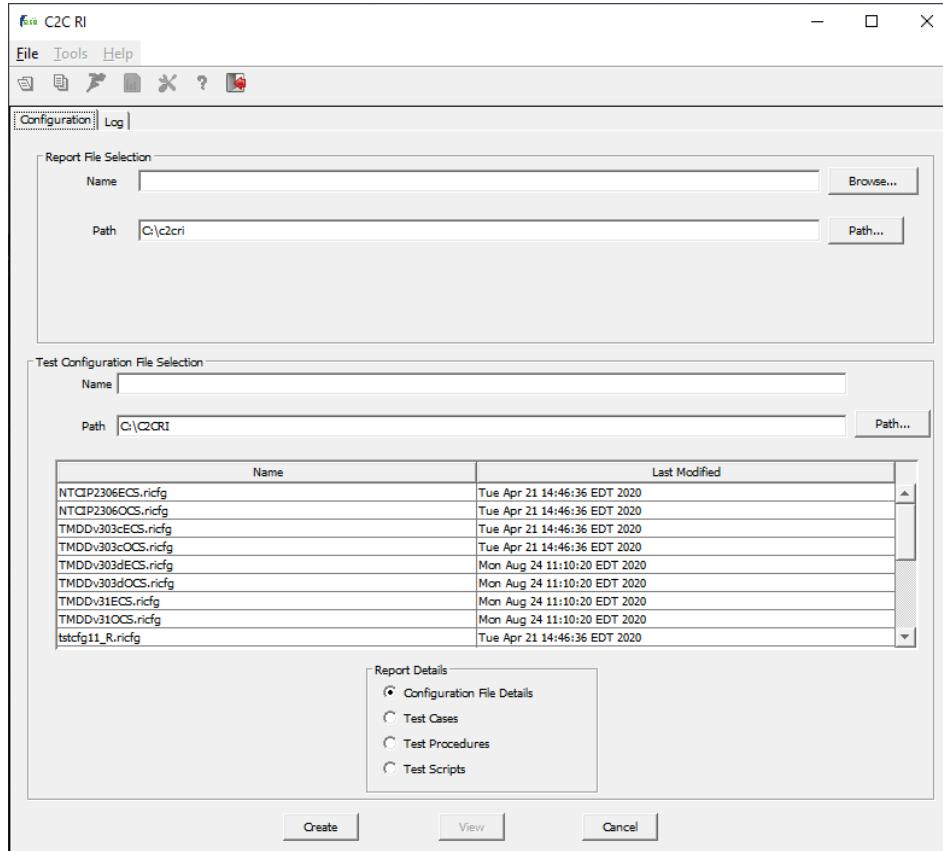


Figure 8-12. Report Configuration Window
(Source: TransCore ITS, LLC, June 2021.).

8. Report Test Results

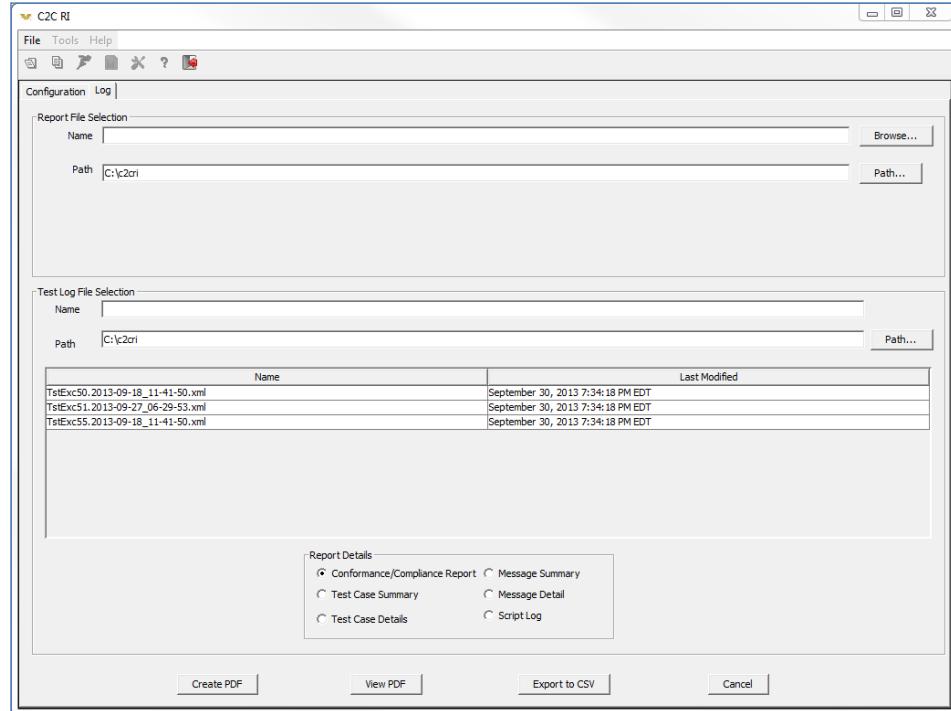


Figure 8-13. Report Log Window
(Source: TransCore ITS, LLC, February 2014.)

Enter a valid name for the test log report in the Name field inside the Report File Selection section of the screen.

Click the **Path** button to change the current directory to one where the report file should be stored (optional). When the **Browse** button is clicked, a dialog similar to the one shown in [Figure 5-3](#) will be displayed. This button can be used to select an existing file name for the report.

Enter a different path for the file in the Path field (optional). When the **Path** button is clicked, a dialog similar to the one shown in [Figure 5-3](#) will be displayed.

Select a Test Log File from the Test Log File Selection section of the screen and then the test reporting screen will look similar to [Figure 8-14](#).

8. Report Test Results

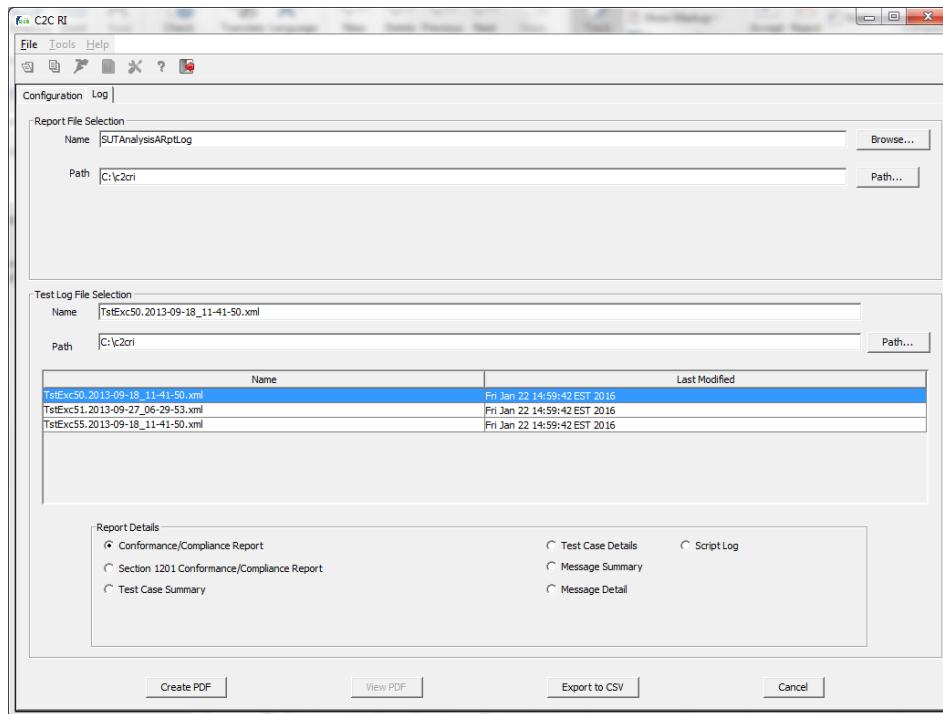


Figure 8-14. Test Log Reporting Screen
(Source: TransCore ITS, LLC, September 2017.)

Select one of the Report Details type. The user has the option to select any of these reports: Conformance/Compliance Report, Test Case Summary, Test Case Details, Message Summary, Message Details and Script Log.

Click the **Create PDF** button and a processing dialog similar to the one shown in [Figure 8-4](#) will be displayed. Once the requested report is generated the system will display a confirmation dialog similar to the one shown in [Figure 8-15](#) to the user.

8. Report Test Results

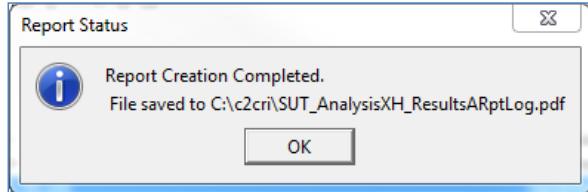


Figure 8-15. Test Log Report Completion
(Source: TransCore ITS, LLC, February 2014.)

Click the **OK** button on the Report Status dialog.

Click the **View PDF** button on the Test Reporting screen.

Test results are presented for three hierarchical categories: test cases, requirements, and needs; each category's result depends on the results of the lower level category. Test cases are the lowest level, and needs are the highest level.

Possible results of test cases are

1. Pass
2. Fail

Possible results of requirement tests are

1. Passed: All test cases related to this requirement were executed and passed.
2. Failed: One or more of the test cases related to this requirement failed.
3. No Test Cases Applicable in This Test Mode: No test cases are defined for this requirement when operating in this test mode.
4. Not Fully Tested: Only a subset of the test cases related to this requirement was tested, and all passed.
5. Not Tested: None of the test cases related to this requirement were tested.

Possible results of need tests are

1. Passed: All test cases related to the requirements that trace to this need were fully tested and passed.
2. Failed: One or more of the test cases related to a requirement that traces to this need failed.
3. Partial Pass: Only a subset of the requirements that trace to this need were tested, and all passed.
4. Not Tested: None of the test cases related to the requirements that trace to this need were tested.
5. No Test Cases Applicable in This Test Mode: No test cases are defined for this need when operating in this test mode.

8.3.1 Conformance or Compliance Report

If the 'Conformance/Compliance Report' report detail type was selected, then the first page of the report will look similar to the information presented in Figure 8-16.

8. Report Test Results

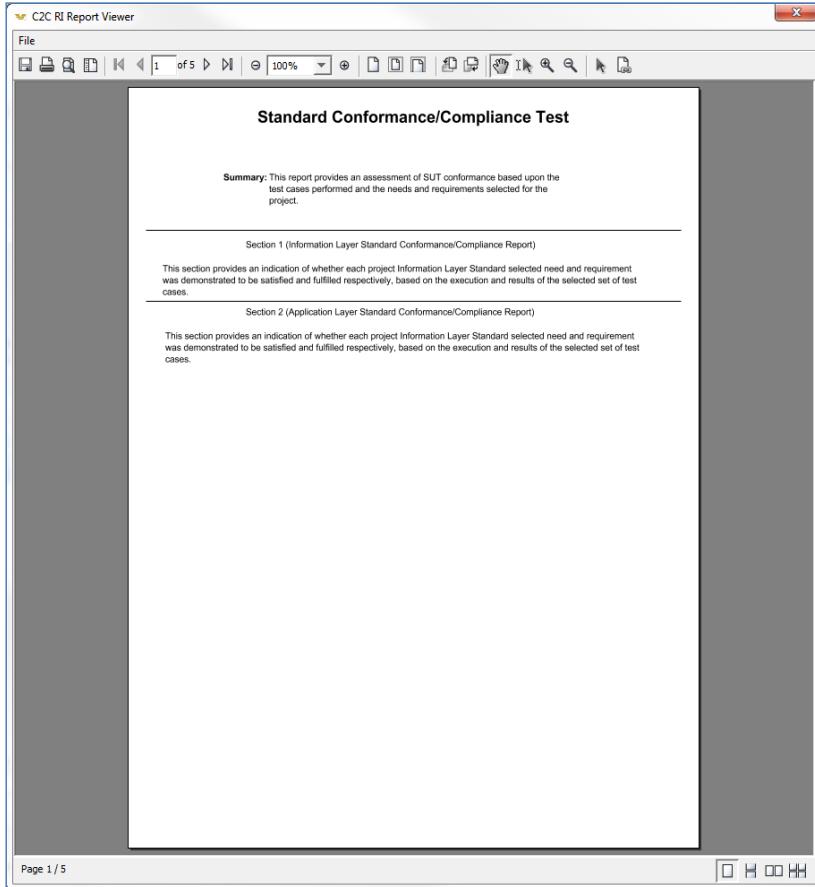


Figure 8-16. Standard Conformance/Compliance Test
(Source: TransCore ITS, LLC, February 2014.)

Click the **next Page** icon on the C2C RI Report Viewer to move to the second page of the Conformance/Compliance Report and the information that user should be able to view will look similar to the information presented in [Figure 8-17](#).

8. Report Test Results

The screenshot shows a window titled "C2C RI Report Viewer". The report details are as follows:

C2C RI Version: Pre-Release 3.0
Created by: Administrator
Date: 10/04/2013 10:44 AM
SUT in Owner Center Mode

Test Suites:
Application: *NTCIP 2306 v1.
Information: *TMDD v3.03a

Configuration File:
Name: C:\C2CRI\Test Config Files\TstCfg25.ricfg
Checksum: 639A57AAAE0375720C5FB6769F06B321

Log File Details:

Log File Name:	C:\c2cn\TstExc50.2013-09-18_11-41-50.xml
Log File Description:	
Log File Creation Date:	2013-09-18_11-41-50

Information Layer Standard Compliance Report

UN ID	User Need	Requirement ID	Other Requirements	Results
*2.3.1.1	Verify Connection Active	*3.3.1.1.1	The owner center shall respond within ____ (100 ms - 1 hour; Default = 1 minute) after receiving the request. : Set to 60000	Not Tested
		*3.3.1.1.4		Not Tested
		*3.3.1.1.4.1		Not Tested
		*3.3.1.1.5		No Test Cases Applicable in this Test Mode
		*3.3.1.1.5.1		No Test Cases Applicable in this Test Mode
		*3.3.1.4.1		No Test Cases Applicable in this Test Mode
		*3.3.1.4.1.1		No Test Cases Applicable in this Test Mode
*2.3.1.2	Need to Support Requests			Need must be indirectly verified.
		Requirement ID	Other Requirements	Results
		*3.3.1.2		Requirement

Source: C:\c2cn\TstExc50.2013-09-18_11-41-50.xml
Page 2 of 5

Figure 8-17. Standard Conformance/Compliance Test Details
(Source: TransCore ITS, LLC, February 2014.)

8.3.2 Section 1201 Conformance or Compliance Report

If the 'Section 1201 Conformance/Compliance Report' report detail type was selected, the first page of the report will look similar to the information presented in Figure 8-18.

8. Report Test Results

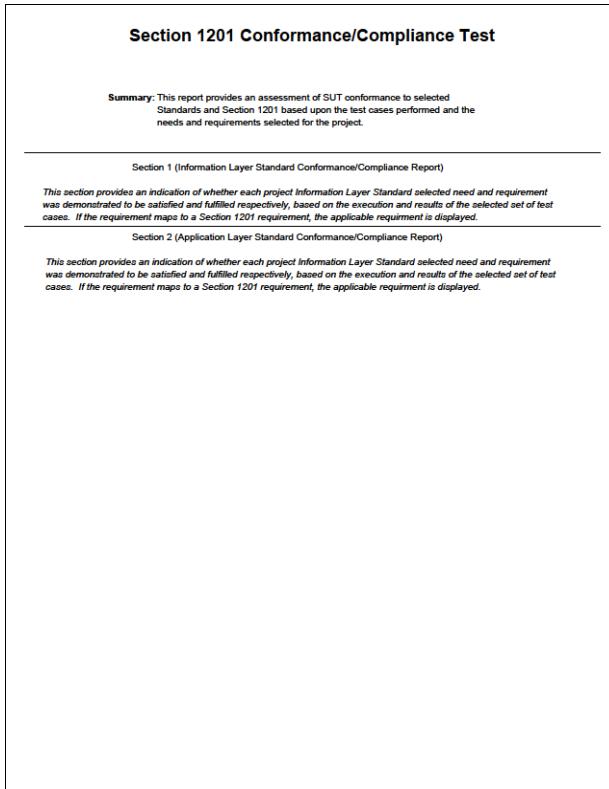


Figure 8-18: Section 1201 Standard Conformance/Compliance Test

(Source: TransCore ITS, LLC, November 2016.)

Click the **next Page** icon on the C2C RI Report Viewer to move to the second page of the Conformance/Compliance Report and the information that user should be able to view will look similar to the information presented in Figure 8-19.

8. Report Test Results

Section 1201 Conformance/Compliance Test Report																			
SUT in Owner Center Mode		12/01/2016 11:31 AM																	
C2C RI Version: Pre-Release-v2-21	Created by: Administrator																		
Test Suites:																			
Application: NTCIP 2306 v1.69																			
Information: TMDD v3.03c																			
Configuration File																			
Name: C:\c2crl\TMDDv303cECS.ricfg																			
Checksum: E34C151465F238DAF6456116C888DBF0																			
<table border="1"><tr><td>Log File Name:</td><td colspan="3">C:\c2crl\TestEC.2016-11-16_00-12-19.xml</td></tr><tr><td>Log File Creator:</td><td colspan="3">Administrator</td></tr><tr><td>Log File Creation Date:</td><td colspan="3">2016-11-16_00-12-19</td></tr><tr><td>Log File Description:</td><td colspan="3"></td></tr></table>				Log File Name:	C:\c2crl\TestEC.2016-11-16_00-12-19.xml			Log File Creator:	Administrator			Log File Creation Date:	2016-11-16_00-12-19			Log File Description:			
Log File Name:	C:\c2crl\TestEC.2016-11-16_00-12-19.xml																		
Log File Creator:	Administrator																		
Log File Creation Date:	2016-11-16_00-12-19																		
Log File Description:																			
Information Layer Standard Conformance Report																			
UN ID	User Need	Results																	
2.3.1.1	Verify Connection Active	Passed																	
	Requirement ID	Related DXFS Requirement	Results																
	3.3.1.1.1	3.4.1.1.1	Passed																
	3.3.1.1.2	3.4.1.1.2	Passed																
	3.3.1.1.3	3.4.1.1.3	Passed																
	3.3.1.1.4	3.4.1.1.4	No Test Cases Applicable in this Test Mode																
	3.3.1.1.4.1	3.4.1.1.5	No Test Cases Applicable in this Test Mode																
	3.3.1.1.5	3.4.1.1.6	Passed																
	3.3.1.1.5.1	3.4.1.1.7	Passed																
	3.3.1.1.5.2.1	3.4.1.1.9	Passed																
	3.3.1.4.1	3.4.4.1	Passed																

Source: C:\c2crl\TestEC.2016-11-16_00-12-19.xml

Page 2 of 217

Figure 8-19: Section 1201 Conformance/Compliance Test Details
(Source: TransCore ITS, LLC, November 2016.)

8.3.3 Test Case Summary Report

If a ‘Test Case Summary’ report detail type was selected, then the first page of the report will look similar to the information presented in [Figure 8-20](#)[Figure 8-20](#).

8. Report Test Results

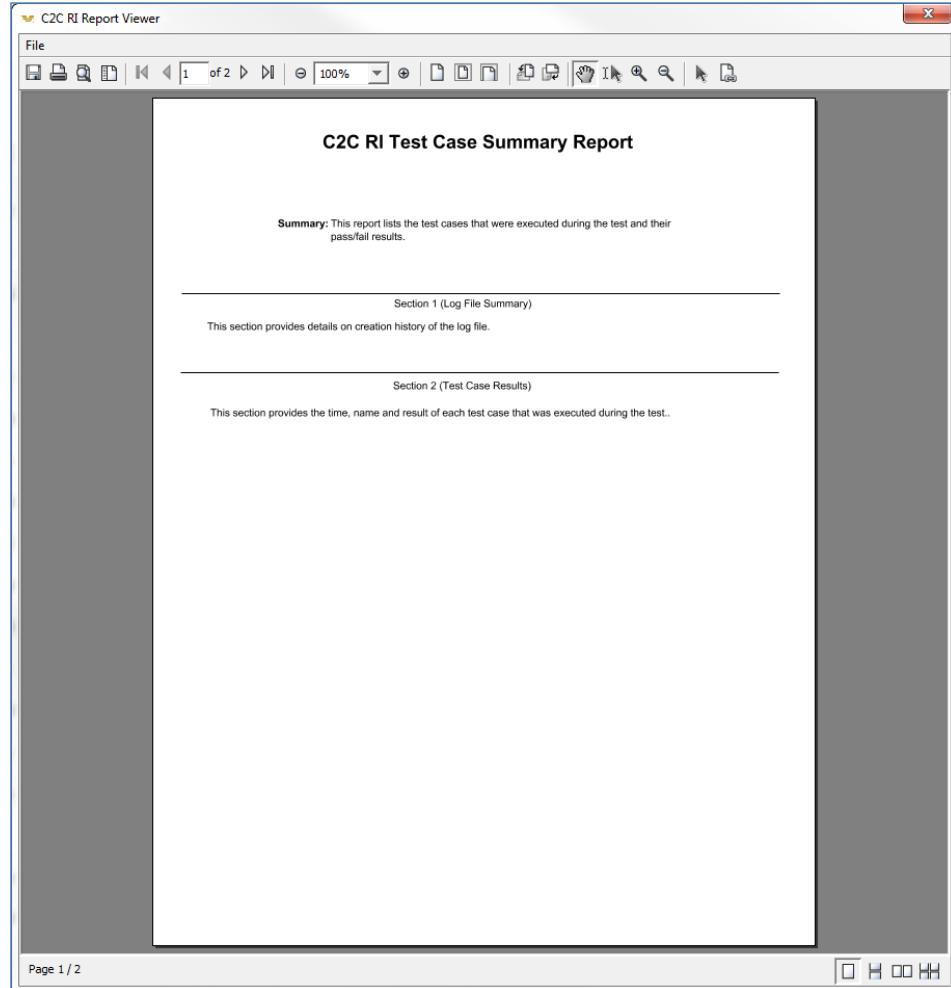


Figure 8-20. C2C RI Test Case Summary Report

(Source: TransCore ITS, LLC, February 2014.)

Click the **next Page** icon (any number of times depending on the page number to view) on the C2C RI Report Viewer to move to another page of the Test Case Summary Report and the information that user should be able to view will look similar to the information presented in [Figure 8-21](#).

8. Report Test Results

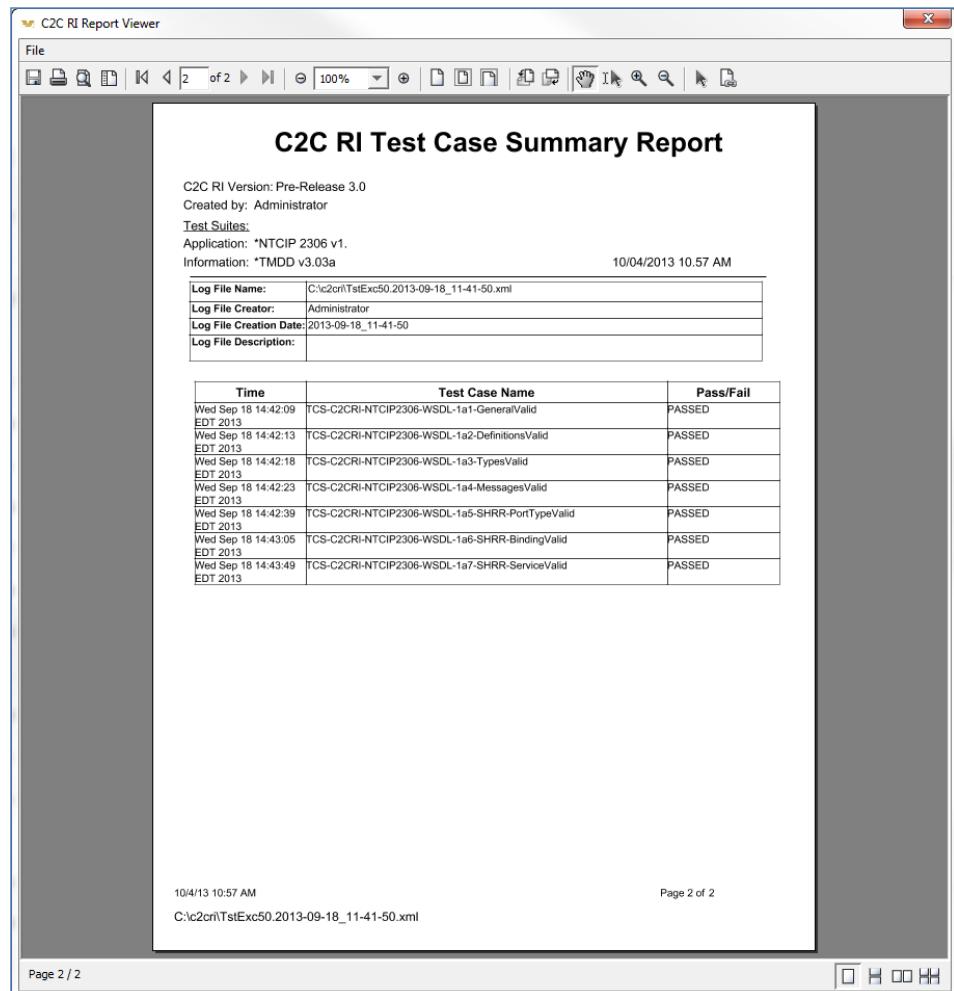


Figure 8-21. C2C RI Test Case Summary Report Details

(Source: TransCore ITS, LLC, February 2014.)

8.3.4 Test Case Details Report

If the 'Test Case Details' report detail type was selected, then the first page of the report will look similar to the information presented in Figure 8-22.

8. Report Test Results

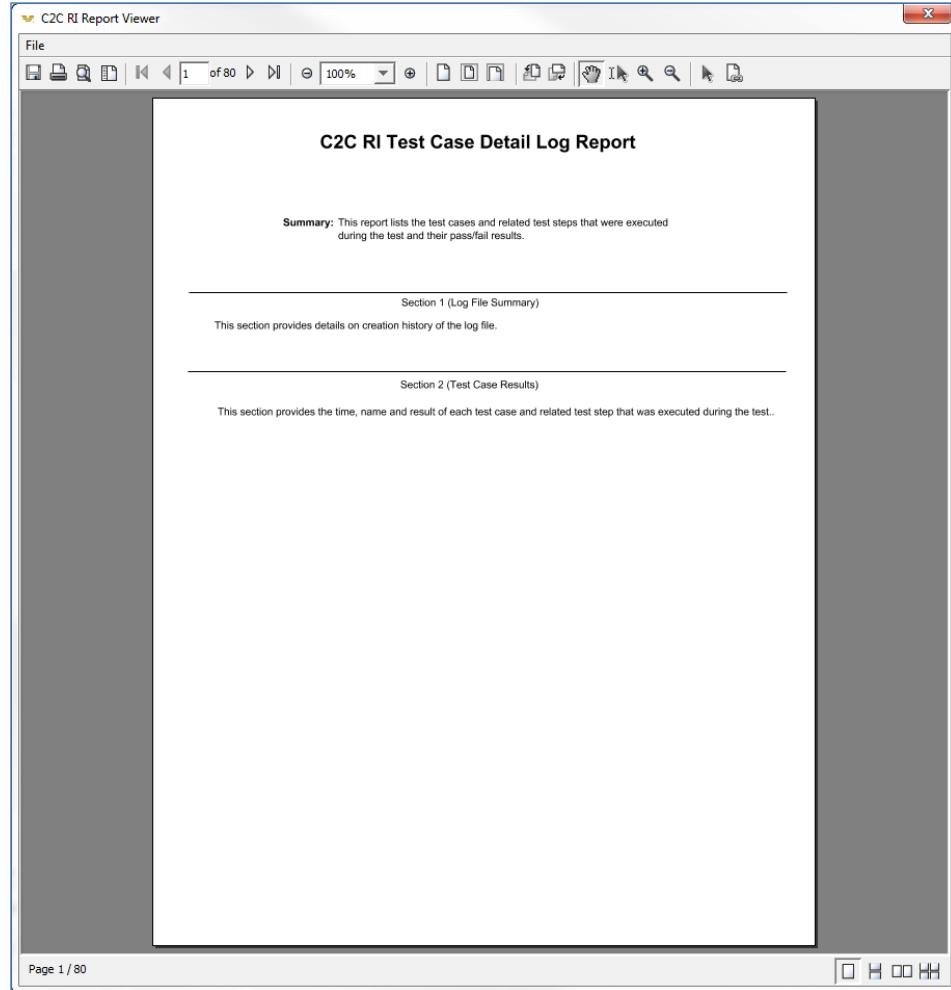


Figure 8-22. C2CRI Test Case Detail Log Report

(Source: TransCore ITS, LLC, February 2014.)

Click the **next Page** icon on the C2C RI Report Viewer to move to the second page of the Test Case Details Report and the information that user should be able to view will look similar to the information presented in [Figure 8-23](#)[Figure 8-23](#).

8. Report Test Results

The screenshot shows the 'C2C RI Report Viewer' window. The title bar reads 'C2C RI Report Viewer'. The menu bar includes 'File' and various icons for navigation and search. The main content area displays the 'C2C RI Test Case Detail Log Report'. The report header includes the following information:

- C2C RI Version: Pre-Release 3.0
- Created by: Administrator
- Test Suites:**
- Application: *NTCIP 2306 v1.
- Information: *TMDD v3.03a
- Date: 10/04/2013 11:22 AM

Below the header is a table with the following data:

Time	Test Case Name	Test Step Description	Pass/Fail	Fail Reason
18/09/2013 14:42.09.261	TCS-C2CRI-NTCIP2306-WSDL-1a1- GeneralValid	Step 1 VERIFY that the specified WSDL file exists.	PASSED	
18/09/2013 14:42.10.731		Step 2 VALIDATE the WSDL file using the W3C WSDL 1.1 Schema.	PASSED	
18/09/2013 14:42.10.758		Step 3 VERIFY that the WSDL file consists of 5 child sections: types/schema, message, portType, binding, service.	PASSED	
18/09/2013 14:42.10.785		Step 4 VERIFY that the definitions tag includes a name attribute.	PASSED	
18/09/2013 14:42.10.823		Step 5 VERIFY that the type section begins with the types tag.	PASSED	
18/09/2013 14:42.10.854		Step 6 VERIFY that the schema tag immediately follows the types tag.	PASSED	
18/09/2013 14:42.10.883		Step 7 VERIFY that the import tag follows the types tag and specifies any namespace being imported and the schemaLocation.	PASSED	
18/09/2013 14:42.10.922		Step 8 VERIFY that each message includes all the level messages defined in the XML Schema that apply to the project implementation for Request-Response transmission pattern.	PASSED	
18/09/2013 14:42.11.048		Step 9 VERIFY that each message type begins with [MSG].	PASSED	
18/09/2013 14:42.11.075		Step 10 VERIFY that the	PASSED	
18/09/2013				

At the bottom of the report, the date '10/4/13 11:22 AM' and the file path 'C:\c2cn\TstExc50.2013-09-18_11-41-50.xml' are displayed. The page footer indicates 'Page 2 / 80' and contains icons for printing and navigating through the report pages.

Figure 8-23. C2C RI Test Case Detail Log Report Details

(Source: TransCore ITS, LLC, February 2014.)

8.3.5 Message Summary Report

If the 'Message Summary' report detail type was selected, then the first page of the report will look similar to the information presented in Figure 8-24.

8. Report Test Results

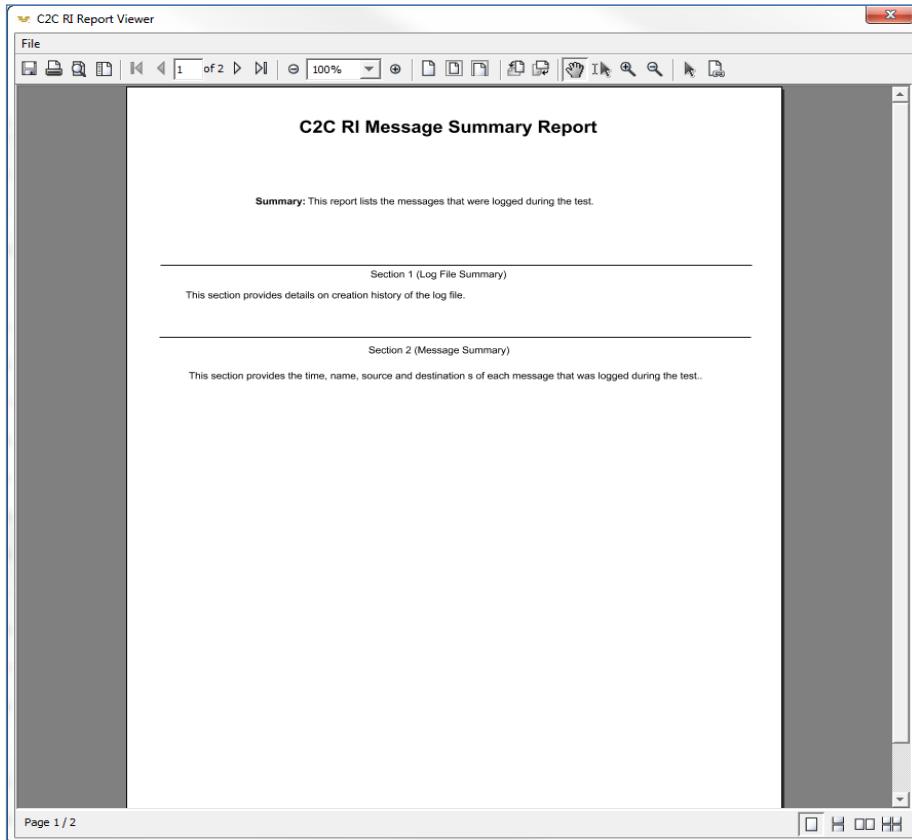


Figure 8-24. C2C RI Message Summary Report
(Source: TransCore ITS, LLC, February 2014.)

Click the **next Page** icon on the C2C RI Report Viewer to move to the second page of the Message Summary Report and the information that user should be able to view will look similar to the information presented in [Figure 8-25](#)[Figure 8-25](#).

8. Report Test Results

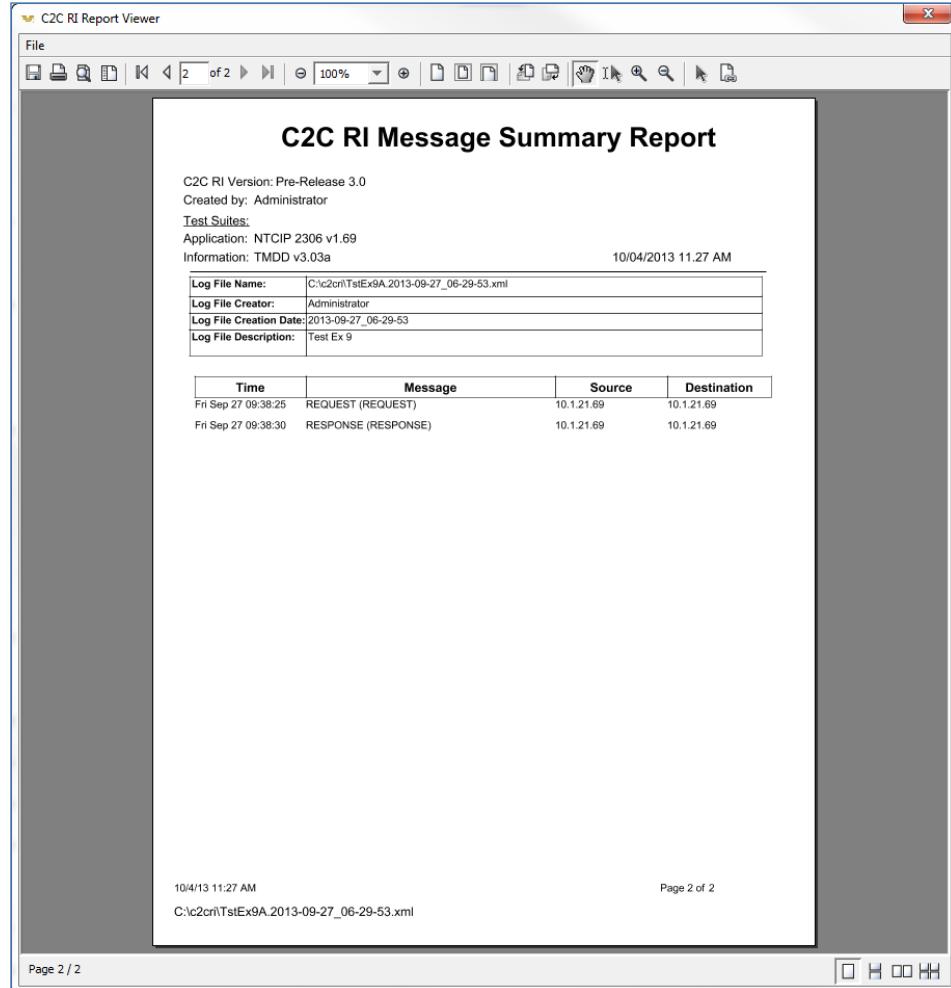


Figure 8-25. C2C RI Message Summary Report Details

(Source: TransCore ITS, LLC, February 2014.)

8.3.6 Message Detail Report

If the 'Message Detail' report detail type was selected, then the first page of the report will look similar to the information presented in Figure 8-26.

8. Report Test Results

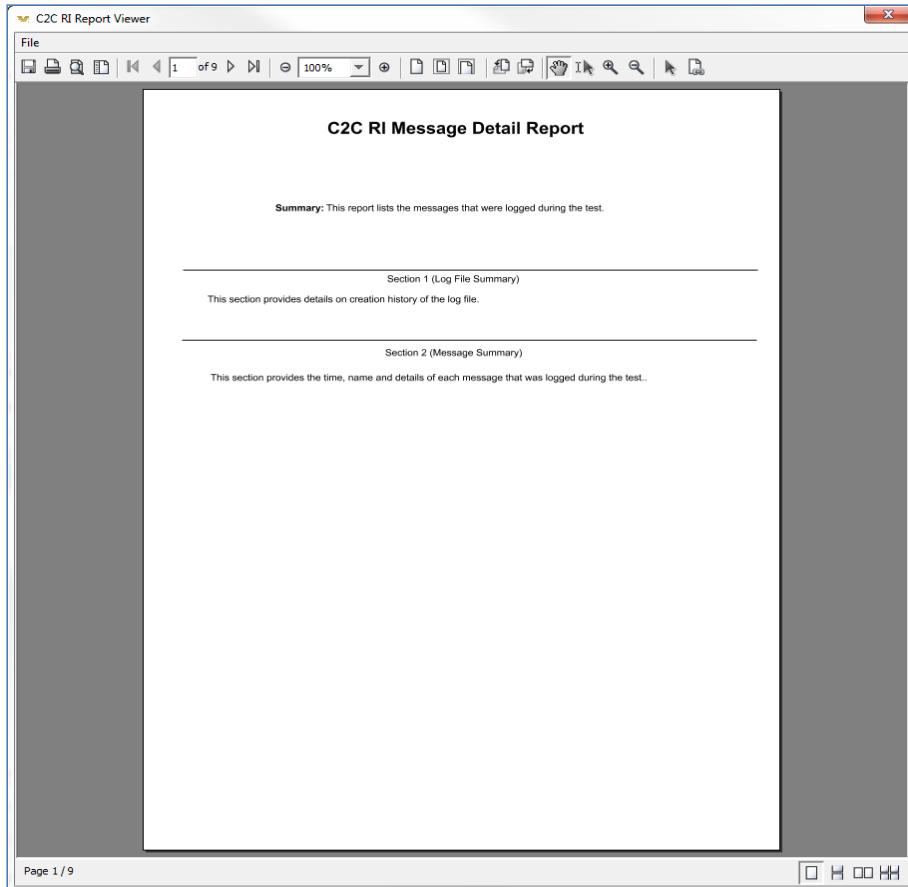


Figure 8-26. C2C RI Message Detail Report
(Source: TransCore ITS, LLC, February 2014.)

Click the **next Page** icon on the C2C RI Report Viewer to move to the second page of the Message Detail report and the information that user should be able to view will look similar to the information presented in [Figure 8-27](#)[Figure 8-27](#).

8. Report Test Results

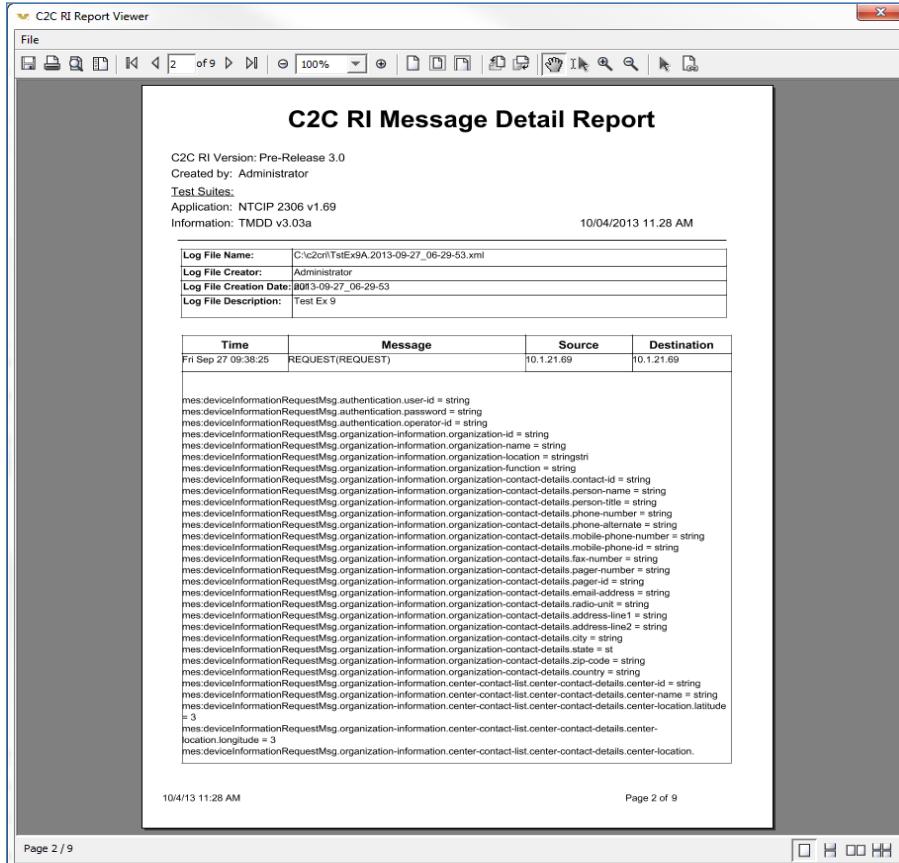


Figure 8-27. C2C RI Message Detail Report Details
(Source: TransCore ITS, LLC, February 2014.)

8.3.7 Test Script Report

If the 'Test Script' report detail type was selected, then the first page of the report will look similar to the information presented in [Figure 8-28](#).

8. Report Test Results

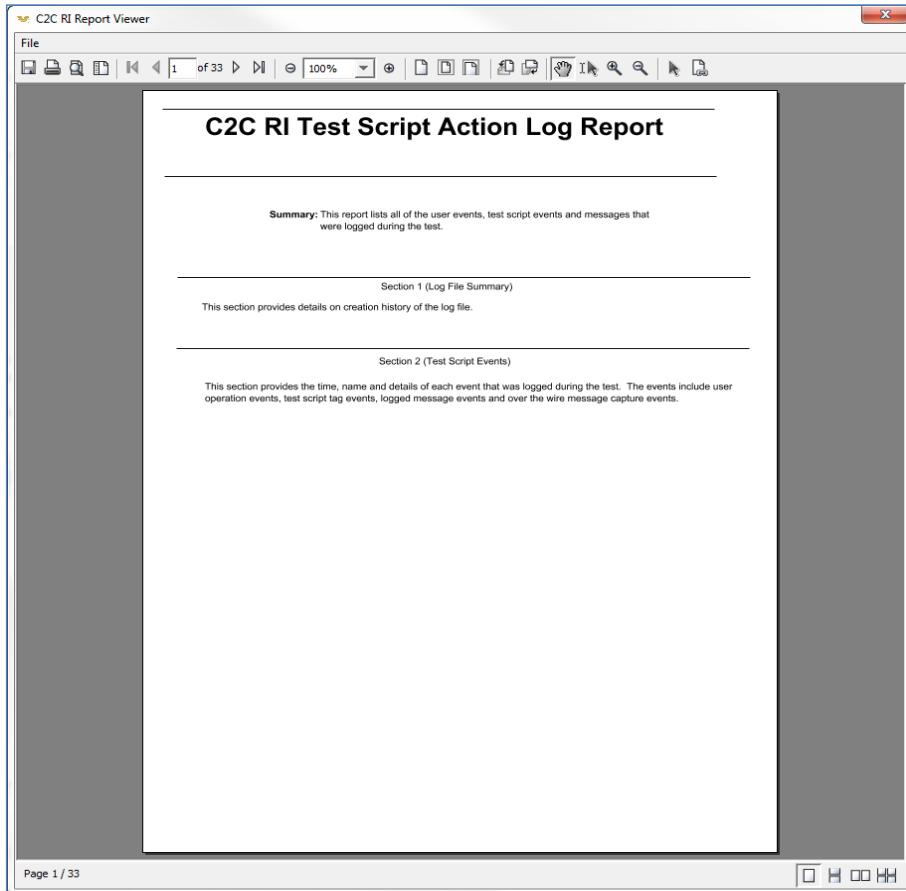


Figure 8-28. C2C RI Test Script Action Log Report
(Source: TransCore ITS, LLC, February 2014.)

Click the **next Page** icon on the C2C RI Report Viewer to move to the second page of the Test Script Report and the information that user should be able to view will look similar to the information presented in [Figure 8-29](#)[Figure 8-29](#).

8. Report Test Results

The screenshot shows the C2C RI Report Viewer window. The title bar reads "C2C RI Report Viewer". The main content area displays the "C2C RI Test Script Action Log Report". The report includes the following information:

- C2C RI Version: Pre-Release 3.0
- Created by: Administrator
- Test Suites:
- Application: NTCIP 2306 v1.69
- Information: TMDD v3.0a
- Date: 10/04/2013 11:30 AM

Below this, there is a table with three columns: TimeStamp, Test Script Action, and Results. The table contains several rows of log entries, each detailing a specific test step, its action, and its outcome (e.g., PASSED). The log file information is also provided at the top of the report.

TimeStamp	Test Script Action	Results
27/09/2013 09:37:24.013	User Started the Test with selected Test Case Scripts-Application Layer: TCS-C2CRI-NTCIP2306-WSME-2Sc-XHRR-THConnect-EC	
27/09/2013 09:37:27.819	testCase - Begin Script: jar!file:/C:/C2CRI/dist/RIGUI.jar!org/fhwfa/c2crl/testmodel/TestCaseLauncher.xml line: 14 column: 1	
27/09/2013 09:37:28.381	testCase - script - Begin Script: jar!file:/C:/C2CRI/dist/RIGUI.jar!org/fhwfa/c2crl/testmodel/TestCaseLauncher.xml functionId: TCS-C2CRI-NTCIP2306-WSME-2Sc-XHRR-THConnect-EC	
27/09/2013 09:37:46.446	testStep - Begin Script: jar!file:/C:/C2CRI//TestSuites/NTCIP2306v01_69Signed.jar!/AppLayer/Scripts/TPS-NTCIP2306-HTTP-POST-EC.xml line: 18 column: 217 functionId: Step 1 CONFIGURE: Identify the WSDL file that will define the one-way Get operation used during the test. RECORD the WSDL File Name value as: WSDLFile =	
27/09/2013 09:37:46.477	testStep - End Script: jar!file:/C:/C2CRI//TestSuites/NTCIP2306v01_69Signed.jar!/AppLayer/Scripts/TPS-NTCIP2306-HTTP-POST-EC.xml line: 18 column: 217 functionId: Step 1 CONFIGURE: Identify the WSDL file that will define the one-way Get operation used during the test. RECORD the WSDL File Name value as: WSDLFile =	PASSED
27/09/2013 09:37:46.493	testStep - Begin Script: jar!file:/C:/C2CRI//TestSuites/NTCIP2306v01_69Signed.jar!/AppLayer/Scripts/TPS-NTCIP2306-HTTP-POST-EC.xml line: 19 column: 253 functionId: Step 2 CONFIGURE: Identify the HTTP related Service defined in the WSDL that will be used for the two-way POST operation during the test. RECORD the Service	
27/09/2013 09:37:46.493	testStep - End Script: jar!file:/C:/C2CRI//TestSuites/NTCIP2306v01_69Signed.jar!/AppLayer/Scripts/TPS-NTCIP2306-HTTP-POST-EC.xml line: 19 column: 253 functionId: Step 2 CONFIGURE: Identify the HTTP related Service defined in the WSDL that will be used for the two-way POST operation during the test. RECORD the Service	PASSED
27/09/2013 09:37:46.508	testStep - Begin Script: jar!file:/C:/C2CRI//TestSuites/NTCIP2306v01_69Signed.jar!/AppLayer/Scripts/TPS-	

At the bottom left, it says "Page 2 / 33" and at the bottom right, there are icons for search, refresh, and exit.

Figure 8-29. C2C RI Test Script Action Log Report Details
(Source: TransCore ITS, LLC, February 2014.)

Select the **File** menu and then select: **Exit** from inside the C2C RI Report Viewer to close the generated report at any time. Clicking the **(X)** button in the top right hand corner of the window can also be used to accomplish this task.

8.4 How to Export Log Reports to CSV

1. Select the **File** menu and then select: **Reports** and the dialog shown in [Figure 5-5](#)[Figure 5-5](#) will be opened.
2. Select the 'Log' tab and the dialog shown in [Figure 8-14](#)[Figure 8-14](#) will be opened.

8. Report Test Results

3. Enter a valid name for the test log report in the Name field inside the Report File Selection section of the screen.
4. Click the **Path** button to change the current directory to one where the report file should be stored (optional). When the **Browse** button is clicked, a dialog similar to the one shown in [Figure 5-3](#)[Figure 5-3](#) will be displayed. This button can be used to select an existing file name for the report.
5. Enter a different path for the file in the Path field (optional). When the **Path** button is clicked, a dialog similar to the one shown in [Figure 5-3](#)[Figure 5-3](#) will be displayed.
6. Select a Test Log File from the Test Log File Selection section of the screen.
7. Select one of the Report Details type. The user has the option to select any of these reports: Conformance/Compliance Report, Test Case Summary, Test Case Details, Message Summary, Message Details and Script Log.
8. Click the **Export to CSV** button and a processing dialog similar to the one shown in [Figure 8-4](#)[Figure 8-4](#) will be displayed. Once the requested report is generated the system will display a confirmation dialog similar to the one shown in [Figure 8-30. CSV Export Status Report](#)[Figure 8-30](#)[Figure 8-30](#) to the user.

Formatted: Check spelling and grammar

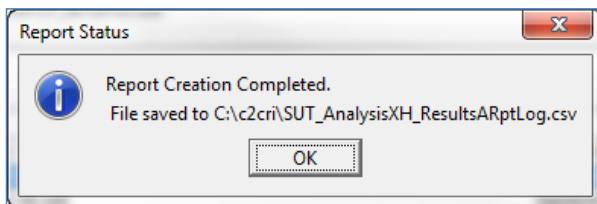


Figure 8-30. CSV Export Status Report
(Source: TransCore ITS, LLC, February 2014.)

9. Click the **OK** button on the Report Status dialog.
10. Move to directory path location where the CSV file was to be stored and check that a CVS file is created using the name that was entered by the user.

8.5 Report Anomaly

There is a known anomaly with the report generator which occasionally cuts off the end of text in a report cell as shown at the bottom of [Figure 8-31](#)[Figure 8-31](#). If you experience this, generate the report again, which usually results in the complete text.

8. Report Test Results

Standard Conformance/Compliance Test Report				
Information Layer Standard Conformance Report				
UN ID	User Need	Requirement ID		Results
		Time	Test Case ID	Error Description
		09.18.17 12:08:56.06 0	TCS-3-PeriodicSubscription- EC-Valid	Errors NOT expected, but found! jar:file:/C:/C2CRI//TestSuites/TMD Dv303d- Signed.jar!/InfoLayer/Scripts/TPS-3- PeriodicSubscription-EC.xml ----- --- CAUSE: jar:file:/C:/C2CRI//TestSuites/TMD Dv303d- Signed.jar!/InfoLayer/Scripts/TPS-3- PeriodicSubscription- EC.xml:61:440: <PUBLICATION- OC> TESTSTEP FAILURE: Error with publication-oc: 'Application Layer Error: No SOAP PUB
		3.3.1.3.2		Passed

Figure 8-31: Report Anomaly Example
(Source: TransCore ITS, LLC, October 2017)

9.0 Customizing the C2C RI and Performing Support Functions

This section describes how to manage the RI's support files and how to define the RI's execution parameters to support custom installation needs. It also provides an introduction to the development of custom test suites.

9.1 Maintaining Configuration and Log Files

1. Select the **Tools** menu and select: **Maintenance** and the dialog shown in [Figure 5-6](#). [Maintenance Dialog](#) will be opened. Click the **Cancel** button to close the dialog at any time.
2. Select either **Configuration Files** or **Log Files** under Show Files of Type option.
3. Click the **Browse** button to browse to a directory where configuration files or log files exist.
4. Select the configuration or log file in the list that you want to maintain.
5. Click the **Delete** button and the dialog shown in Figure 9-1 will be displayed. Click the **No** button to keep the file or the **Yes** button to proceed with the deletion.

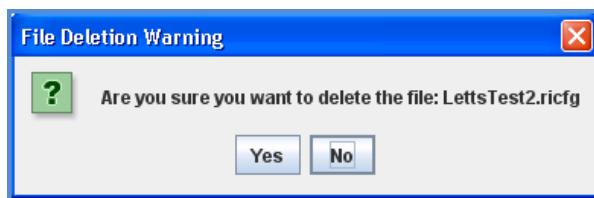


Figure 9-1. File Deletion Warning
(Source: TransCore ITS, LLC, February 2014.)

9.2 Setting Software Options

There are few options available in the RI due to its purpose of providing uniformity in test for conformance to the ITS standards. The available options are comprised of an agency identifier and file locations. The agency identifier is used in producing the RI reports. The file locations are for managing the data used and reported by the RI as well as the RI's installation directories. To manage these data, use the following procedure:

1. Select the **Tools** menu and select: **Options** and the dialog shown in [Figure 5-7](#) will open. Click the **Cancel** button to close the dialog at any time.

2. Select **General, Logging, or Testing** tab under Editor Panel.
3. Click the *Attribute Value* of the item to be modified or select the icon to the right of the value.
4. Set the value on the Testing tab appropriately for the installed system. This value indicates whether testing is between an SUT and the RI (SUT/RI) (the only testing the typical user will ever perform), internal RI testing only (RI) (used only during RI development), or both internal and SUT testing (RI;SUT/RI) (the complete set of test cases).
5. Click the **Save** or **Done** button to save the changes. **Cancel** will close the dialog without saving the new settings.

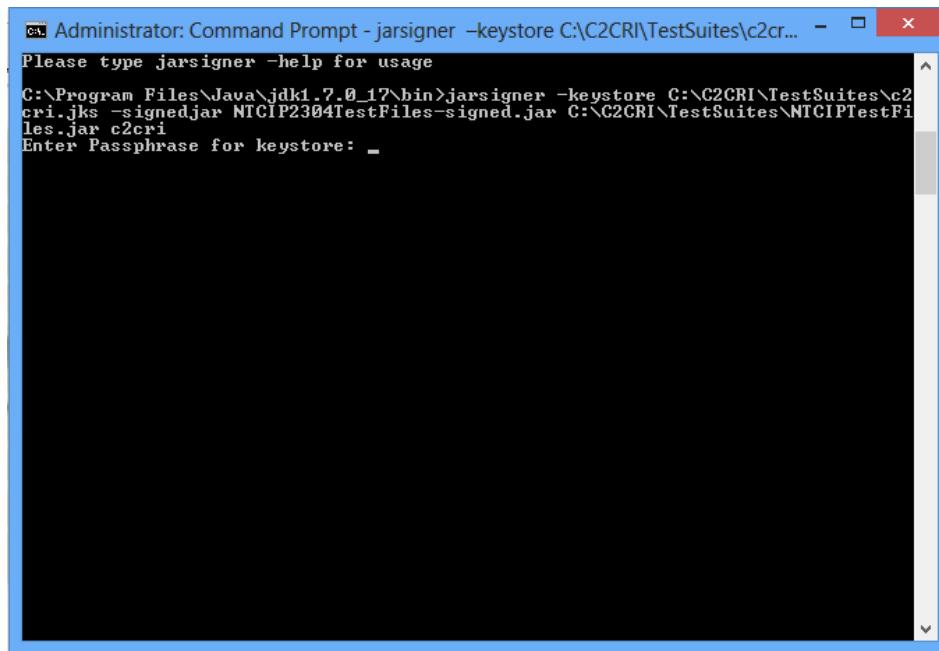
9.3 Creating Authorized Custom Test Suites

The C2C RI application is capable of supporting custom test suites. With this available functionality users can create their own custom test suites which if authorized by the system will be displayed in the C2C RI application. When the new authorized application or information layer test suite is selected a red labeled text is displayed below the drop down list box that reads ‘User Defined Test Suite’ indicating that the RI allowed the user to add a new user defined test suite to the list.

The following list of steps details the process of adding a new authorized custom application or information layer test suite to the system:

1. Close the C2C RI application if it is currently running.
 2. Create a directory folder on any workstation.
 3. Create a sub-folder inside that parent folder with the name ‘AppLayer’ or ‘InfoLayer’ depending on the test suite that is needed.
 4. Create a ‘SuiteSpec.properties’ file inside the parent folder/directory.
 5. Double-click the folder named ‘AppLayer’ or ‘InfoLayer’ and create the following files and folders: a folder name ‘Data’ to hold test case data file, a folder name ‘Scripts’ to hold test procedure scripts, ‘NRTM.csv’ file, ‘Predicates.csv’ file, ‘TestCaseMatrix.csv’ file and ‘TestCases.csv’.
 6. Double-click to open the file named ‘SuiteSpec.properties’.
 7. Modify the: SuiteName, InfoLayerStandard, AppLayerStandard, SuiteDescription, BaselineTestSuiteName and TestSuiteType. Note that the value used for the ‘TestSuiteType’ should be ‘Application’ if an application layer test suite is being created or ‘Information’ if it is for an information layer test suite.
 8. Save the changes to the file named ‘SuiteSpec.properties’ and then close it.
 9. Move back to the location where the parent folder was created and then compress the file.
 10. Right-click the compressed file and select the ‘Rename’ command and then change the extension of the file to a JAR file. For example, ‘MyTestSuite.jar’.
 11. Copy the newly created JAR file to the ‘C:\C2CRI\ TestSuites’
 12. Open the DOS prompt from the RI user workstation.
 13. Enter “cd c:\” in the DOS window.
-

14. Enter “cd C:\Program Files\Java\jdk1.7.0_17\bin” in the DOS window. Note: ‘C:\Program Files\Java\jdk1.7.0_17\bin’ is just an example of the location where the ‘jarsigner.exe’ file is installed on the user’s workstation.
15. Enter “jarsigner –keystore C:\C2CRI\TestSuites\c2cri.jks –signedjar NTCIP2304TestFiles-signed.jar C:\C2CRI\TestSuites\NTCIPTestFiles.jar c2cri” and the command should look similar to what is shown in [Figure 9-2](#). The user will then be prompted to enter the passphrase for the keystore. Note in this example ‘NTCIP2304TestFiles-signed.jar’ is the name of the new JAR file that will be created and ‘NTCIPTestFiles.jar’ is the name of the JAR file (converted from a folder) that will be signed.



The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt - jarsigner -keystore C:\C2CRI\TestSuites\c2...". The window contains the following text:
Please type jarsigner -help for usage
C:\Program Files\Java\jdk1.7.0_17\bin>jarsigner -keystore C:\C2CRI\TestSuites\c2cri.jks -signedjar NTCIP2304TestFiles-signed.jar C:\C2CRI\TestSuites\NTCIPTestFiles.jar c2cri
Enter Passphrase for keystore: _

Figure 9-2. DOS Jarsigner Arguments
(Source: TransCore ITS, LLC, February 2014.)

16. Enter the appropriate passphrase.
17. Move to the ‘C:\Program Files\Java\jdk1.7.0_17\bin’ directory and check if this file is shown: ‘NTCIP2304TestFiles-Signed.jar’.
18. While inside the directory ‘C:\C2CRI\TestSuites’ right-click onto the file named ‘NTCIP2306v01_69Signed.zip’ and select ‘Rename’ and enter the name ‘file named ‘NTCIP2306-Signed.jar’.

19. Move to this location on the user workstation: ‘C:\Program Files\Java\jdk1.7.0_17\bin’ then right-click the file named ‘NTCIP2304TestFiles-signed.jar’ that was signed and created there and select the ‘Copy’ command.
20. Move to this location on the user workstation: ‘C:\C2CRI\TestSuites’ then right-click into the directory location and select the ‘Paste’ command.
21. Launch the C2C RI application.
22. Select the **File** menu and select: **New** command from the C2C RI MENU BAR.
23. Click the down arrow button for the Application Layer or Information Layer Standard Test Suite pull-down list (depending on which test suite was added to the system) and check the names shown for the different test suites. Verify that the system shows the new custom defined test suite in the list which matches the new name of the test suite entered earlier.
24. Select the new test suite from the list and check that it shows the correct name and description that was entered in the ‘SuiteSpec.properties’ file. Once the test suite is selected a black labeled text is displayed below the drop down list box that reads ‘User defined Test Suite’ indicating that the RI allowed the user to add a new custom test suite to the list.

9.4 Creating User Defined Test Suites

The C2C RI application is capable of supporting user defined test suites. With this available functionality users are able to create their own test suites which will be displayed in the C2C RI application even though they are not authorized. When a new application or information layer test suite is selected a red labeled text is displayed below the drop down list box that reads ‘User Defined Test Suite’ indicating that the RI allowed the user to add a new user defined test suite to the list.

A user-defined information layer test suite can be created using similar steps explained here when creating one for the application layer. The following steps detail the process of adding a new user defined application layer test suite to the system:

1. Close the C2C RI application if it is currently running.
2. Move to the ‘C:\C2CRI’ directory and double-click the folder named ‘CustomTestSuites’.
3. Right-click inside the ‘C:\C2CRI\CustomTestSuites’ directory and select ‘New->Folder’ command and then enter a new name for the folder.
4. Create a sub-folder inside that parent folder with the name ‘AppLayer’ or ‘InfoLayer’ depending on the test suite that is needed.
5. Create a ‘SuiteSpec.properties’ file also inside the parent folder/directory.
6. Double-click the folder named ‘AppLayer’ or ‘InfoLayer’ and create the following files and folders: a folder name ‘Data’ to hold test case data file, a folder name ‘Scripts’ to hold test procedure XML scripts, ‘NRTM.csv’ file, ‘Predicates.csv’ file, ‘TestCaseMatrix.csv’ file and ‘TestCases.csv’.
7. Double-click to open the file named ‘SuiteSpec.properties’.
8. Modify the: SuiteName, InfoLayerStandard, AppLayerStandard, SuiteDescription, BaselineTestSuiteName and TestSuiteType. Note that the value used for the ‘TestSuiteType’ should

be ‘Application’ if an application layer test suite is being created or ‘Information’ if it for an information layer test suite.

9. Save the changes to the file named ‘SuiteSpec.properties’ and then close it.
10. Launch the C2C RI application.
11. Select the **File** menu and select: **New** command from the C2C RI MENU BAR.
12. Click the down arrow button for the Application Layer Standard Test Suite pull-down list. Verify that the system shows the new user-defined test suite that was created earlier.
13. Select the new test suite from the list and check that it shows the correct name and description that was entered in the ‘SuiteSpec.properties’ file. Verify that once the test suite is selected that a red labeled text is displayed below the drop down list box that reads ‘User Defined Test Suite’ indicating that the RI allowed the user to add a new user defined test suite to the list.

9.5 Modifying Test Case Data

Each test case within a test suite has an associated test case data file that defines the parameters and values necessary for that test case. Some parameters are fixed and may not be changed while others are editable. In certain test environments one or more of the editable parameters may need to be changed to successfully perform a test. To accomplish this, a user may define a test case file that overrides the necessary editable parameters for the test case. When the test case is executed by the RI the parameters values that are defined in the user test case data file replace the identical values that are pre-defined within the test suite test case data file. There are two methods for modifying test case data, both of which are described below:

1. Using the built-in test case data editor
2. Using a text editor (Section 9.5.2)

9.5.1 Using the Built-in Test Case Data Editor

The user can edit test case data for information and application layer test cases using the build-in editor, which is accessible from the test cases tabs in the Configuration File Window shown in [Figure 6-5](#)[Figure 6-5](#) and [Figure 6-6](#)[Figure 6-6](#). Click the “Edit” pushbutton for the test case to be edited, and an Open dialog similar to the one shown in Figure 9-3 will open, from which you can choose (or create) the relevant test case data file on which to base your edits.

The most common use of the built-in editor is to modify attribute values on each of the group tabs (General, Values, Message). The Test Case Parameters Editor Window (Figure 9-4) is initially populated with the default test case data. Attribute values which are editable are shown in light gray, and values which cannot be edited are dark gray and are noted as “Not Editable” in the bottom pane of the window.

The built-in editor has other functionality, described below, to modify iteration and group names and to add/remove/move parameters in the data files.

9. Performing Support Functions

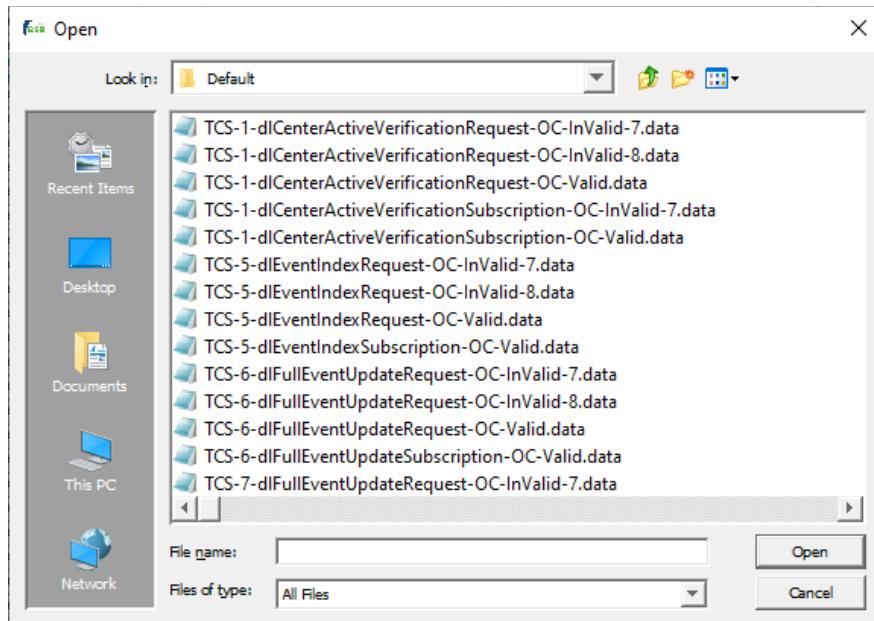


Figure 9-3. Test Case Data Editor Open Dialog
(Source: TransCore ITS, LLC, June 2021.)

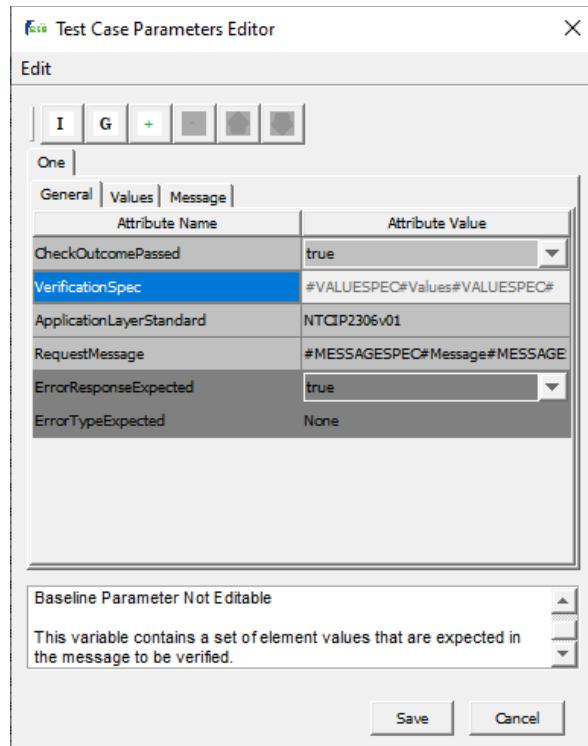


Figure 9-4. Test Case Parameters Editor
(Source: TransCore ITS, LLC, June 2021.)

Test Case Parameters Editor Layout Screen

The Test Case Parameters Editor dialog has six main sections:

1. The Menu Bar
2. Main Tool Bar
3. Test Case Iteration Bar
4. Test Case Group Bar
5. Attribute Name and Attribute Value Display Window
6. Variable Description Window

Test Case Parameters Editor Menu Bar

The Test Case Parameters Editor dialog has only one menu bar item: Edit. This menu has six sub-menu options, as shown in Figure 9-5. These sub-menu options duplicate the behavior provided by the tool bar icons.

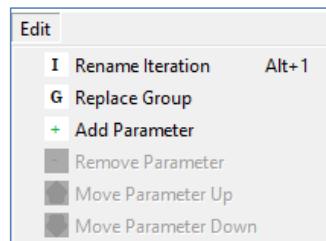


Figure 9-5: Test Case Parameters Editor Menu Bar

(Source: TransCore ITS, LLC, June 2021.)

Test Case Parameters Editor Main Tool Bar

The Test Case Parameters Editor dialog contains a tool bar with icons, from left right for Rename Iteration, Replace Group, Add Parameter, Remove Parameter, Move Parameter Up, and Move Parameter Down.



Test Case Parameters Editor Iteration Tab

The Test Case Parameters Editor dialog contains an Iteration tab which shows the name of the iteration(s) defined in the test case. This tab is between the Main Tool Bar and the Groups tabs. Figure 9-6 shows the name of the same iteration, specified in the test casa data file, in Figure 9-7.

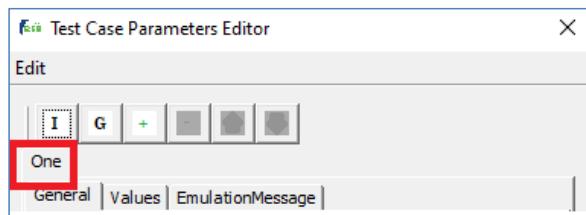


Figure 9-6: Test Case Parameter Editor Iteration Bar

(Source: TransCore ITS, LLC, June 2021.)

9. Performing Support Functions

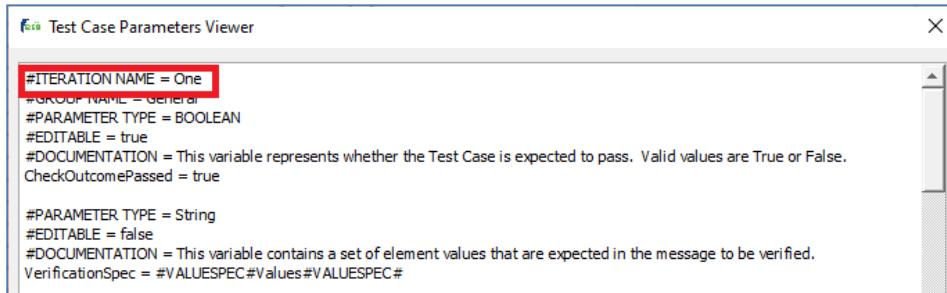


Figure 9-7: Test Case Viewer Showing Iteration Name
(Source: TransCore ITS, LLC, June 2021.)

Test Case Parameters Editor Group Bar

The Test Case Parameters Editor dialog also contains a Groups tab section which shows the name of the groups defined in the test case. This bar is between the Iteration tab and the Attribute Name and Attribute Value Display window as shown in Figure 9-8. This figure shows the names of the groups as specified in the test case data file shown in Figure 9-9

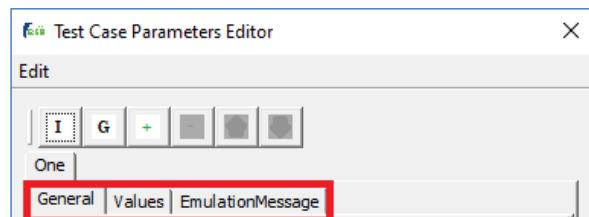
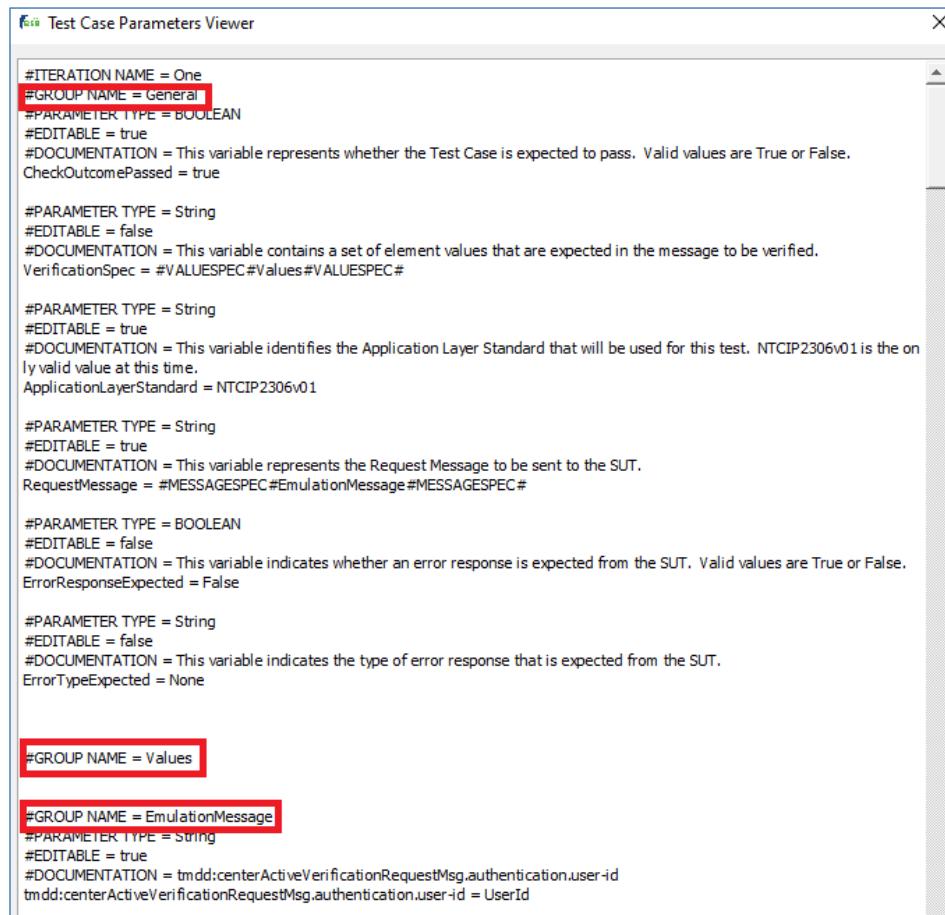


Figure 9-8: Test Case Parameters Editor Groups Tab Bar
(Source: TransCore ITS, LLC, June 2021.)

9. Performing Support Functions



The screenshot shows the 'Test Case Parameters Viewer' window. The window title is 'Test Case Parameters Viewer'. Inside, there is a large block of text representing XML-like configuration for a test case. The text includes several sections starting with '#ITERATION NAME = One', '#GROUP NAME = General', and '#PARAMETER TYPE = BOOLEAN'. It also contains sections for documentation, verification specifications, application layer standards, request messages, error response expectations, and specific parameters like 'ErrorResponseExpected' and 'ErrorTypeExpected'. Two specific sections are highlighted with red boxes: '#GROUP NAME = Values' and '#GROUP NAME = EmulationMessage'. The 'Values' section is associated with a 'String' type parameter and a documentation entry involving 'tmdd:centerActiveVerificationRequestMsg.authentication.user-id'. The 'EmulationMessage' section is associated with a 'String' type parameter and a documentation entry involving 'tmdd:centerActiveVerificationRequestMsg.authentication.user-id = UserId'.

```
#ITERATION NAME = One
#GROUP NAME = General
#PARAMETER TYPE = BOOLEAN
#EDITABLE = true
#DOCUMENTATION = This variable represents whether the Test Case is expected to pass. Valid values are True or False.
CheckOutcomePassed = true

#PARAMETER TYPE = String
#EDITABLE = false
#DOCUMENTATION = This variable contains a set of element values that are expected in the message to be verified.
VerificationSpec = #VALUESPEC#Values#VALUESPEC#

#PARAMETER TYPE = String
#EDITABLE = true
#DOCUMENTATION = This variable identifies the Application Layer Standard that will be used for this test. NTCIP2306v01 is the only valid value at this time.
ApplicationLayerStandard = NTCIP2306v01

#PARAMETER TYPE = String
#EDITABLE = true
#DOCUMENTATION = This variable represents the Request Message to be sent to the SUT.
RequestMessage = #MESSAGESPEC#EmulationMessage#MESSAGESPEC#

#PARAMETER TYPE = BOOLEAN
#EDITABLE = false
#DOCUMENTATION = This variable indicates whether an error response is expected from the SUT. Valid values are True or False.
ErrorResponseExpected = False

#PARAMETER TYPE = String
#EDITABLE = false
#DOCUMENTATION = This variable indicates the type of error response that is expected from the SUT.
ErrorTypeExpected = None

#GROUP NAME = Values

#GROUP NAME = EmulationMessage
#PARAMETER TYPE = String
#EDITABLE = true
#DOCUMENTATION = tmdd:centerActiveVerificationRequestMsg.authentication.user-id
tmdd:centerActiveVerificationRequestMsg.authentication.user-id = UserId
```

Figure 9-9: Test Case Viewer Showing Group Names
(Source: TransCore ITS, LLC, June 2021.)

Test Case Parameters Editor Attribute Name and Attribute Value Display Window

The Test Case Parameters Editor dialog contains an Attribute Name and Attribute Value Display window, between the Groups tabs and Variable Description Window. The Test Case Parameters Editor dialog (Figure 9-10), shows each attribute name and its associated attribute value for the selected group (General, in this example). The Test Case Parameters Viewer (Figure 9-11) shows the attribute names and values that are associated with the group.

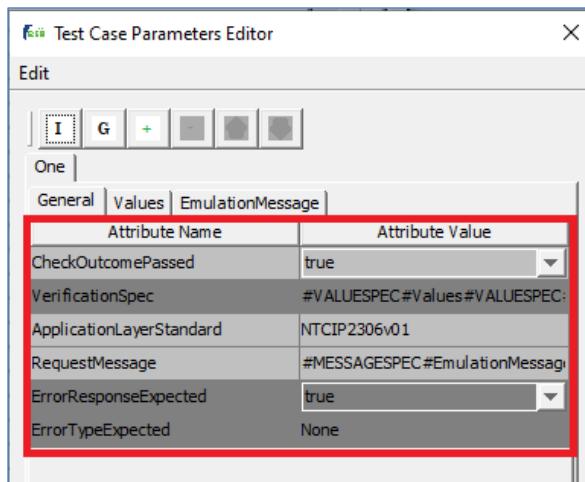


Figure 9-10: Test Case Parameter Editor Showing Attribute Name and Attribute Value
(Source: TransCore ITS, LLC, June 2021.)

9. Performing Support Functions

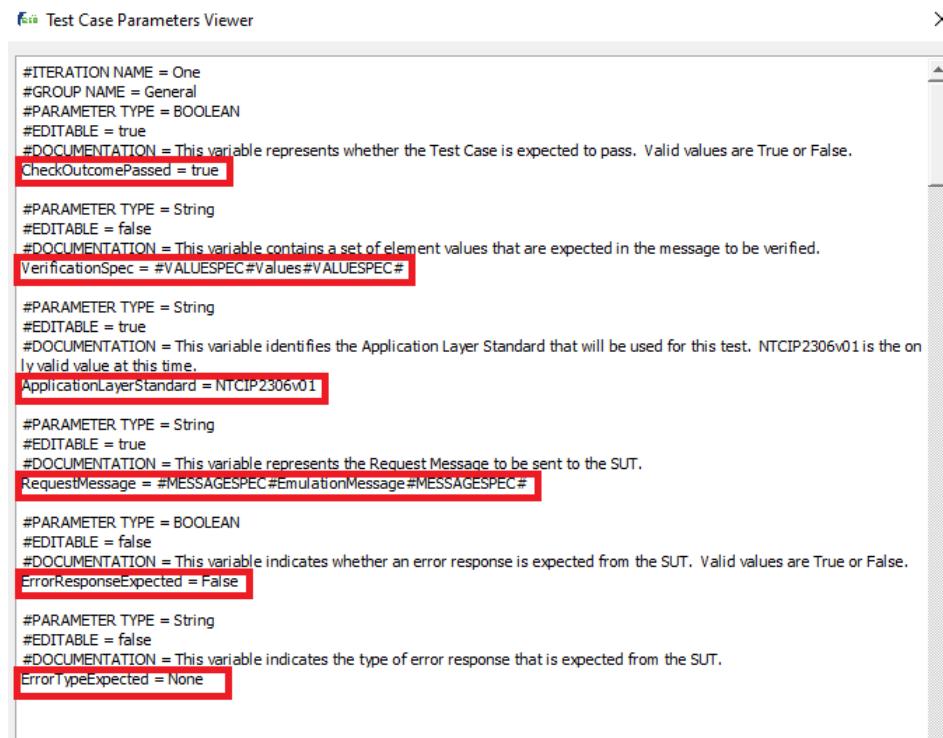


Figure 9-11: Test Case Parameters Viewer Showing Attributes Names and Attribute Values
(Source: TransCore ITS, LLC, June 2021.)

Clicking on any groups tab display all attribute names and their associated attribute values that belong to that group.

Test Case Parameters Editor Variable Description Window

The Test Case Parameters Editor dialog contains a Variable Description Window as shown in Figure 9-12. Clicking on a row inside the Attribute Name and Attribute Value window will give a description for the baseline parameter variable in the Variable Description Window.

Click on the up and down arrow buttons or the scrollbar to scroll through this description window.

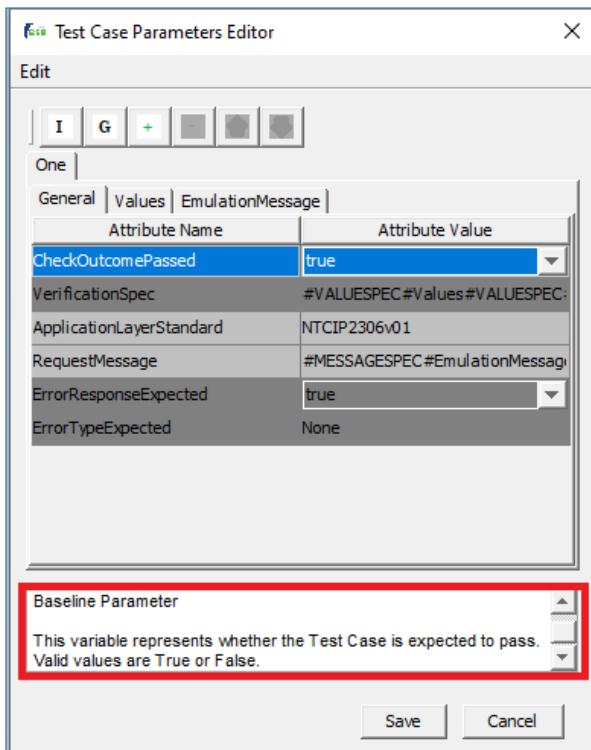


Figure 9-12: Test Case Parameters Editor showing Attribute Baseline Parameter
(Source: TransCore ITS, LLC, June 2021.)

Renaming an Iteration in a Test Case Data File

This feature lets users rename the iteration used in the test case data file by performing the following steps:

1. Open the Test Case Parameters Editor as shown in [Figure 9-4](#)[Figure 9-4](#). The iteration name is shown as a tab inside the Test Case Editor as shown in [Figure 9-13](#)[Figure 9-13](#).

Formatted: Font: Not Bold

Formatted: Font: Not Bold

Formatted: Font: Not Bold

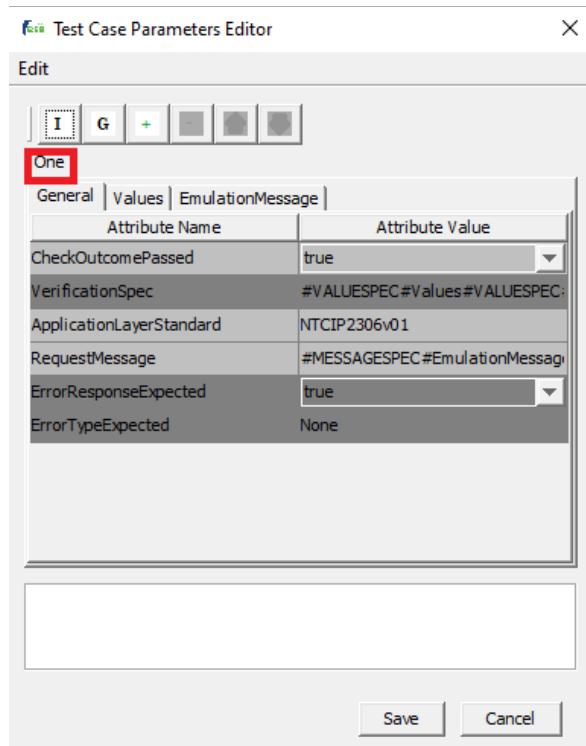


Figure 9-13: Test Case Editor Showing Selected Iteration
(Source: TransCore ITS, LLC, June 2021.)

2. Click on the iteration that will be updated.
3. Click on the “Rename Iteration” tool bar button, and the dialog similar to the one shown in [Figure 9-14](#) [Figure 9-14](#) will be displayed:

Formatted: Font: (Default) Arial, 10 pt

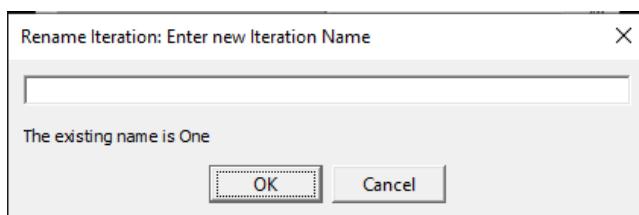


Figure 9-14: Rename Iteration Dialog
(Source: TransCore ITS, LLC, June 2021.)

9. Performing Support Functions

4. Enter a value in the input box to replace the existing iteration name. The system will then display a message showing the existing and new iteration names, as shown in [Figure 9-15](#)[Figure 9-16](#).

Formatted: Font: (Default) Arial, 10 pt

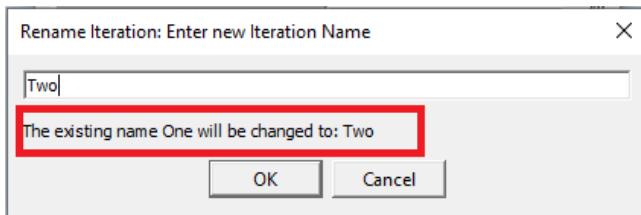
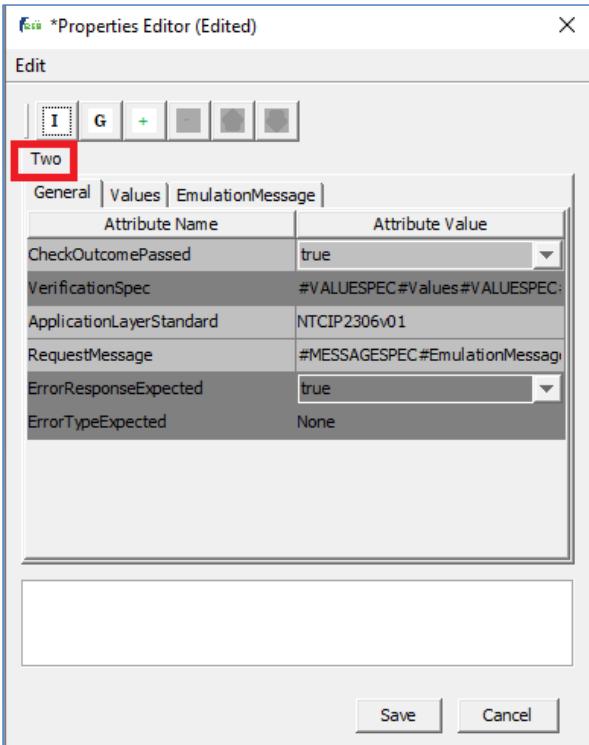


Figure 9-15: Renaming Iteration dialog showing old and new Iteration Name
(Source: TransCore ITS, LLC, June 2021.)

5. Click the OK button. The new iteration name will now replace the original iteration, as shown in [Figure 9-16](#)[Figure 9-16](#). Click the CANCEL button to abort the operation.

Formatted: Font: (Default) Arial, 10 pt

**Figure 9-16: Iteration Showing New Name**

(Source: TransCore ITS, LLC, June 2021.)

6. Click the SAVE button on the Properties Editor (Edited) dialog to save the change to the test case data file. Click the CANCEL button to abort the operation.

Replace Group in a Test Case Data File

This feature lets users replace a group used in the test case data file by performing the following steps:

1. Open the Test Case Parameters Editor as shown in [Figure 9-4](#)[Figure 9-4](#). Each group name is shown as a tab inside the Test Case Parameters Editor dialog as shown in [Figure 9-8](#)[Figure 9-8](#).
2. Click on the tab which represents the group that will be replaced.
3. Click on the “Replace Group” button on the tool bar, and a dialog similar to the one shown in [Figure 9-17](#)[Figure 9-17](#) will be displayed.

Formatted: Font: (Default) Arial, 10 pt

Formatted: Font: (Default) Arial, 10 pt

Formatted: Font: (Default) Arial, 10 pt

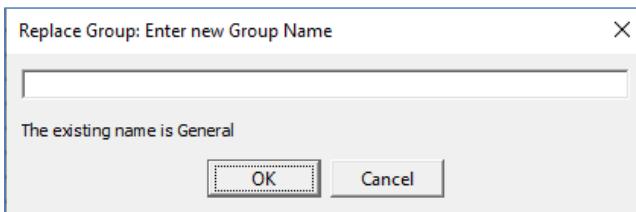


Figure 9-17: Replace Group Dialog
(Source: TransCore ITS, LLC, June 2021.)

4. Enter a value in the input box to replace the existing group name. The system will display a message showing the existing and new group names, as shown in [Figure 9-18](#)[Figure 9-18](#).

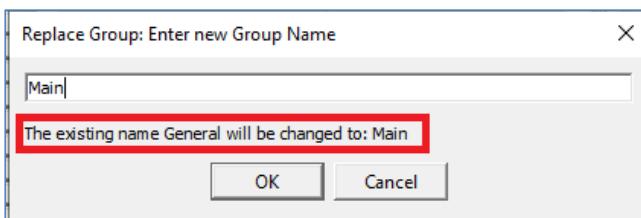


Figure 9-18: Replace Group Dialog Showing Old and New Group Name
(Source: TransCore ITS, LLC, June 2021.)

5. Click the OK button on the Replace Group dialog box, and a Group Parameters Action dialog box similar to the one in [Figure 9-19](#)[Figure 9-19](#) will be displayed.

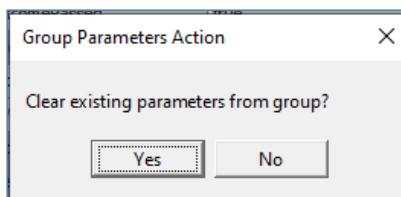


Figure 9-19: Group Parameters Action Dialog
(Source: TransCore ITS, LLC, June 2021.)

6. Confirm the user operation to clear existing parameters from the group by clicking the YES or NO button.

- a) Click the YES button to confirm the operation to clear existing parameters from the group.

The Properties Editor (Edited) dialog will be shown with the new group name along with no attribute names and values as displayed in [Figure 9-20](#)[Figure 9-20](#).

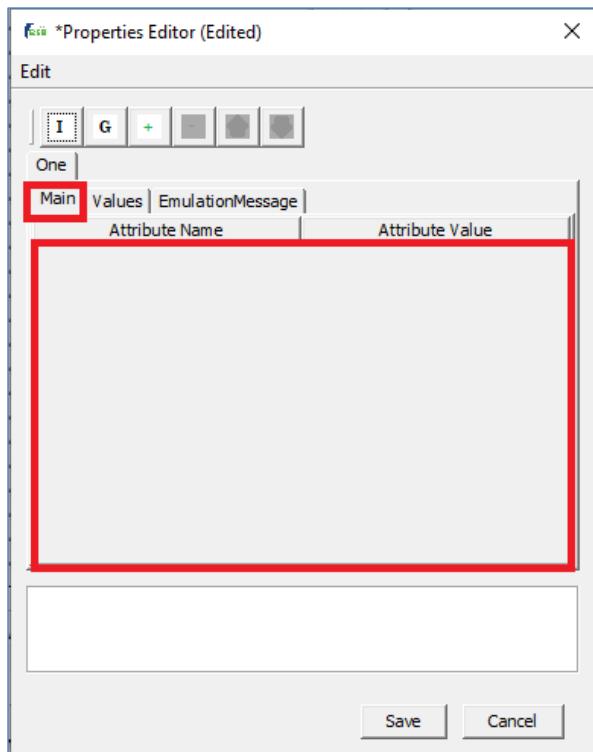


Figure 9-20: Replaced Group Showing Cleared Parameters
(Source: TransCore ITS, LLC, June 2021.)

- b) Click the NO button to keep the existing parameters for the group.

The Properties Editor (Edited) dialog will be shown with the new group name along with any transferred attribute names and values as displayed in [Figure 9-21](#)[Figure 9-21](#).

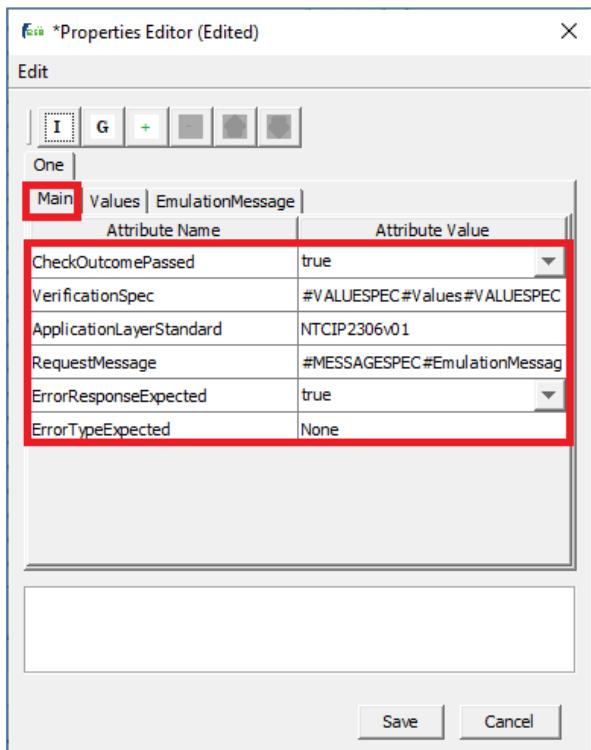


Figure 9-21: Replaced Group Showing Parameters
(Source: TransCore ITS, LLC, June 2021.)

- Click the SAVE button on the Properties Editor (Edited) dialog to save the change to the test case data file. Click the CANCEL button to abort the operation.

Add Parameter to Test Case Data File

This feature gives users the ability to add a new parameter to a group in a test case data file. The steps to be performed are as follows:

- Open the Test Case Parameters Editor as shown in [Figure 9-4](#). Each group name is shown as a tab inside the Test Case Parameters Editor dialog as shown in [Figure 9-8](#).
- Click on the tab which represents the group that will be updated with the new parameter.
- Click on the "Add Parameter" button on the tool bar, and a dialog similar to the one shown in [Figure 9-22](#) will be displayed.

Formatted: Font: (Default) Arial, 10 pt

Formatted: Font: (Default) Arial, 10 pt

Formatted: Font: (Default) Arial, 10 pt

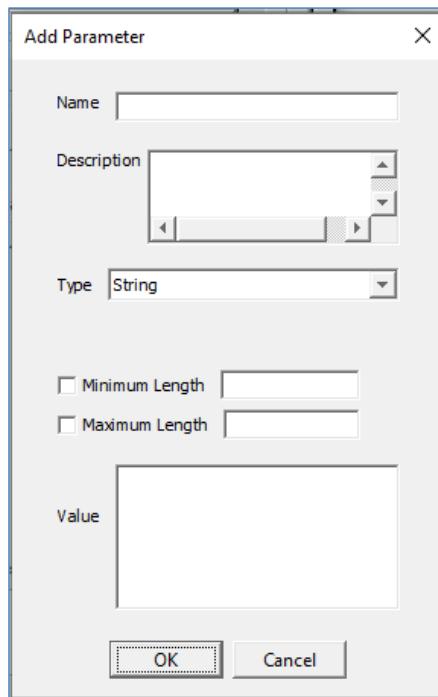


Figure 9-22: Add Parameter Dialog
(Source: TransCore ITS, LLC, June 2021.)

4. Enter a value in the “Name” input field.
5. Enter a value in the “Description” field.
6. Select a value from the “Type” drop down list.
7. Check the box next to “Minimum Length” box and enter a value.
8. Check the box next to “Maximum Length” box and enter a value.
9. Enter a value in the “Value” input field. The completed form should look similar to the one shown in [Figure 9-23](#).

Formatted: Font: (Default) Arial, 10 pt, Condensed by 0.2 pt, Kern at 10 pt

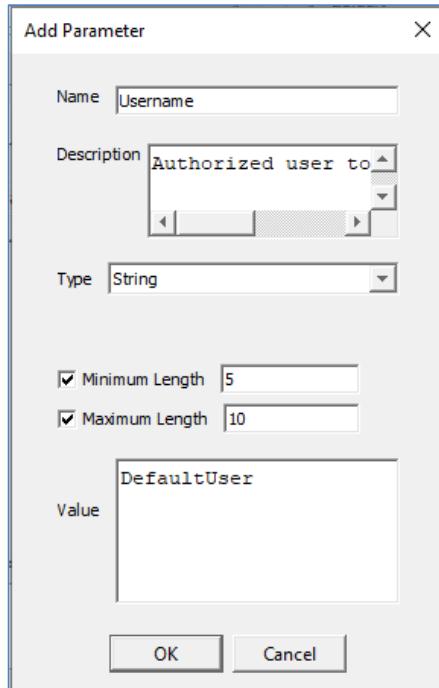


Figure 9-23: Add Parameter Form Completed with Data
(Source: TransCore ITS, LLC, June 2021.)

10. Click the OK button on the Add Parameter dialog, and the Properties Editor (Edited) dialog, which looks similar to [Figure 9-24](#)[Figure 9-24](#), will be displayed. Click the CANCEL button to abort the operation.

Formatted: Font: (Default) Arial, 10 pt

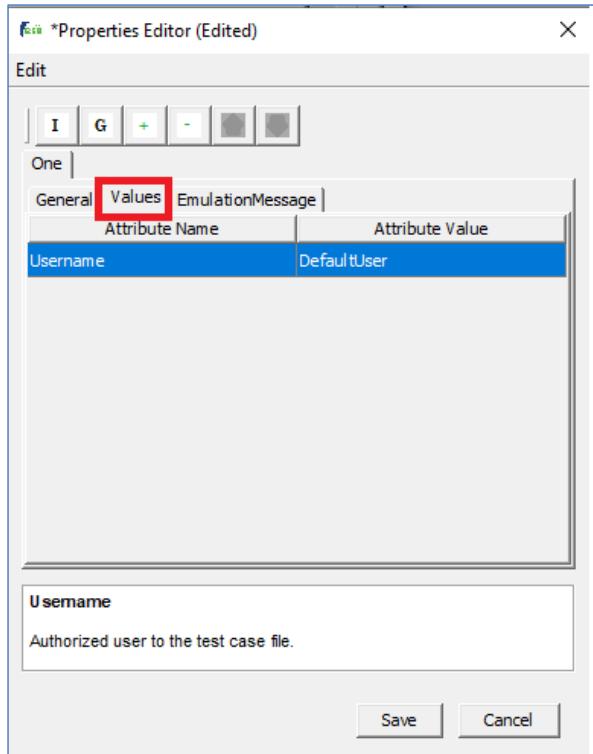


Figure 9-24: New Parameter Added to Group
(Source: TransCore ITS, LLC, June 2021.)

11. Click the SAVE button on the Properties Editor (Edited) dialog to save the change to the test case data file. Click the CANCEL button to abort the operation.
12. Click on the same test case that was updated from the Test Configuration Editing screen as shown in [Figure 9-25](#)[Figure 9-25](#).

Formatted: Font: (Default) Arial, 10 pt

9. Performing Support Functions

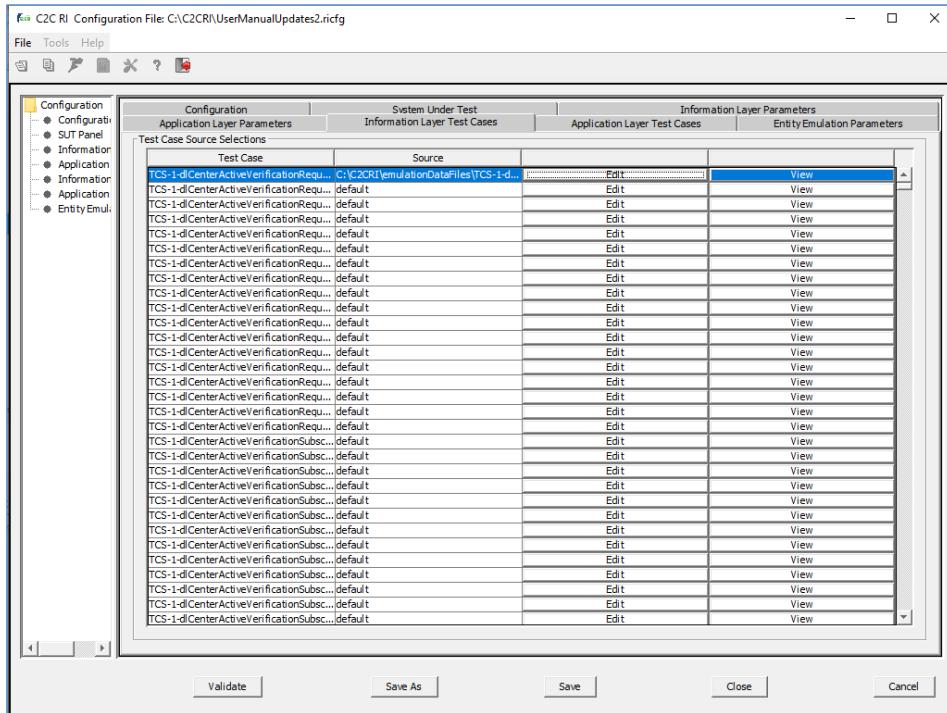


Figure 9-25: Test Configuration Editing Screen Showing Selected Test Case
(Source: TransCore ITS, LLC, June 2021.)

13. Then click on the VIEW button, and the Test Case Viewer will show the details for the new parameter that was added to the group, as shown in [Figure 9-26](#)[Figure 9-26](#).

Formatted: Font: (Default) Arial, 10 pt

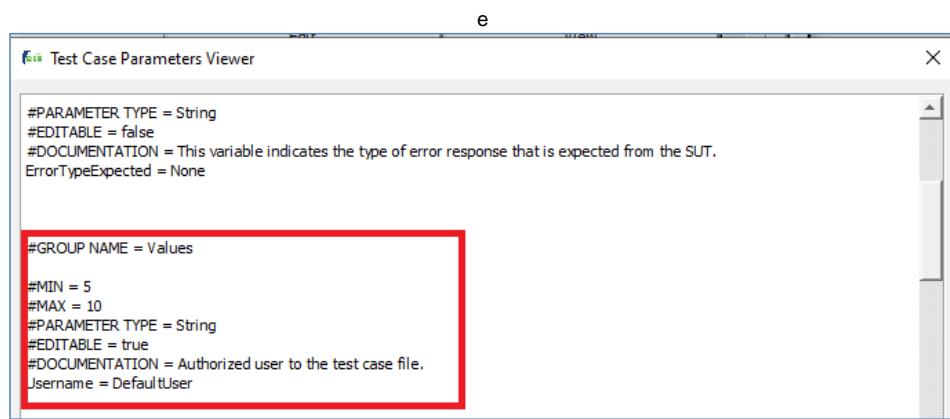


Figure 9-26: Test Case Viewer Showing Newly Added Parameter
(Source: TransCore ITS, LLC, June 2021.)

14. Click the CLOSE button from the Test Case Viewer dialog, and the window will close.

Remove Parameter from Test Case Data File

This feature allows users to remove a parameter from a group defined in a test case data file by following these steps:

1. Open the Test Case Parameters Editor as shown in Figure 9-4. Each group name is shown as a tab inside the Test Case Parameters Editor dialog as shown in [Figure 9-8](#)[Figure 9-8](#).
2. Click on the tab which represents the group from which the parameter will be removed.
3. Click on the attribute to delete, and the attribute will be highlighted in the list.
4. Click on the “Delete Parameter” button in the tool bar, and a dialog similar to the one shown in Figure 9-27 will be displayed:

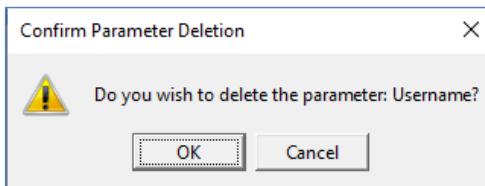


Figure 9-27: Confirm Parameter Deletion
(Source: TransCore ITS, LLC, June 2021.)

5. Click the OK button and the selected parameter will be deleted from the group. Click the CANCEL button to abort the operation.
6. Click the SAVE button on the Properties Editor (Edited) dialog to save the change to the test case data file. Click the CANCEL button to abort the operation.

Move Parameter Up/Down in Test Case Data File

This feature allows users to move a parameter up or down in the attribute list of a group that is defined in a test case data file. The steps to be performed are as follows:

1. Open the Test Case Parameters Editor as shown in [Figure 9-4](#)[Figure 9-4](#). Each group name is shown as a tab inside the Test Case Parameters Editor dialog as shown in [Figure 9-8](#)[Figure 9-8](#).
2. Click on the tab which represents the group from which the parameter will be rearranged.
3. Click on the parameter in the list and the Move Parameter Up and/or the Move Parameter Down icons will be enabled as shown in [Figure 9-28](#)[Figure 9-28](#).

Formatted: Font: (Default) Arial, 10 pt

Formatted: Font: (Default) Arial, 10 pt

Formatted: Font: (Default) Arial, 10 pt

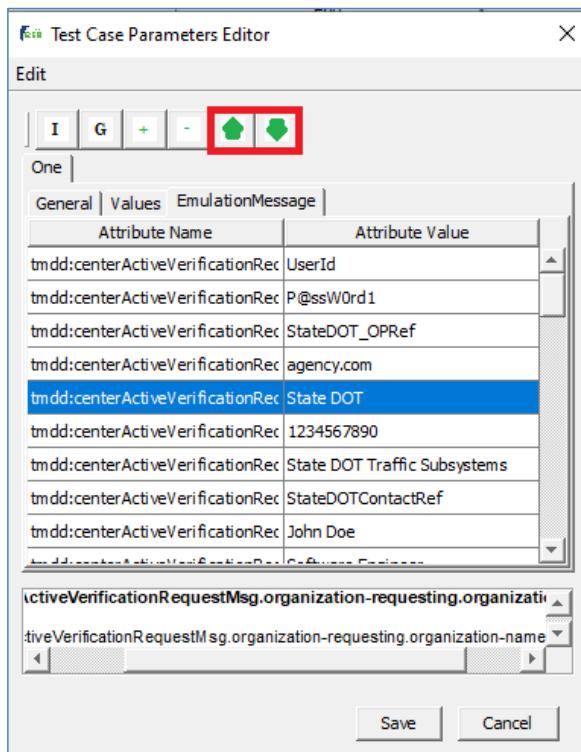


Figure 9-28: Selected Parameter with Move Parameter Up and Move Parameter Down Buttons enabled
(Source: TransCore ITS, LLC, June 2021.)

4. Click on the up or down button, and the parameter will be moved one step from its current position in the grid view as shown in [Figure 9-29](#)[Figure 9-29](#) (parameter moved up).

Formatted: Font: (Default) Arial, 10 pt

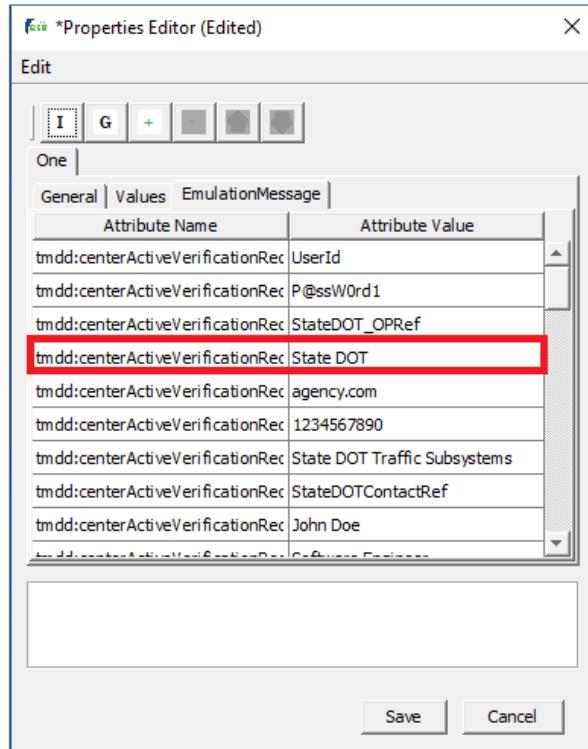


Figure 9-29: Parameter Value in List
(Source: TransCore ITS, LLC, June 2021.)

- Click the SAVE button on the Properties Editor (Edited) dialog to save the change to the test case data file. Click the CANCEL button to abort the operation.

9.5.2 Create a Test Case Data File

The test case data file is a text file that may be created using any text editor. Property files are frequently used in software systems in order to make configuration changes available to the user. The steps to be performed are as follows:

- Create a text file to be used as the user test case data file.
- Enter parameters for each test case data value that needs to be replaced. Information on defining the parameters is described in the Test Case Parameters section below.
- Save the file.

9.5.2.1 Test Case Parameters

Each test case parameter entry is defined with the following format:

PARAMETERNAME = PARAMETERVALUE

PARAMETERNAME represents that name that has been assigned to the parameter and PARAMETERVALUE is the value that will be assigned.

Test case data files are similar to other properties file used to configure software applications; however, they contain additional metadata. The basic meta-attributes available for each property are PARAMETER TYPE, EDITABLE, and DOCUMENTATION. The Parameter Type represents the type of a property. The meta-attribute EDITABLE indicates whether the user can alter the value of a property using a test case data file. The DOCUMENTATION meta-attribute specifies the text that provides assistance to the end user regarding the purpose for the parameter.

Apart from the basic meta-attributes, there are three special meta-attributes. The first one, GROUP NAME, is used to group related properties together. This is often used to group the elements of a message – each defined as parameters – together. Note that this parameter is not specified for each property separately. Once a GROUP NAME tag appears, all the parameters listed before the next GROUP NAME tag are grouped together. Another special meta-attribute is ALLOWED VALUES, which is used in conjunction with the SYMBOL parameter type. This can be used to specify the list of valid choices for a SYMBOL property. These values should be separated by commas.

The final meta-attribute is ITERATION. This attribute groups a set of parameters together including the GROUP attribute – and are used to run test procedures multiple time with different parameters.

Figure 9-30 shows an example of a test case data file. Note that the parameter within the user defined test case file must reside within the same ITERATION and/or GROUP NAME meta-data definitions as the test suite file.

```
#ITERATION NAME = One
#GROUP NAME = Messages
#PARAMETER TYPE = String
#EDITABLE = true
#DOCUMENTATION = This variable represents the WSDL File to be verified.
WSSDLFile = file:///c:/c2cri/testfiles/WSDLFiles/2306TstData1a1.wsdl

#PARAMETER TYPE = boolean
#EDITABLE = true
#DOCUMENTATION = This variable represents whether the Test Case is expected to fail.
CheckOutcomeFailed = true

#PARAMETER TYPE = String
#EDITABLE = true
#DOCUMENTATION = This variable represents the text of the test step where a failure is expected.
FailedTestStepText = Step 11 VERIFY that the WSDL file consists of 5 child sections: types/schema, message, portType, binding, service.

#ITERATION NAME = Two
#GROUP NAME = Messages
#PARAMETER TYPE = String
#EDITABLE = true
#DOCUMENTATION = This variable represents the WSDL File to be verified.
WSSDLFile = file:///c:/c2cri/testfiles/WSDLFiles/2306TstData1a2.wsdl

#PARAMETER TYPE = boolean
#EDITABLE = true
#DOCUMENTATION = This variable represents whether the Test Case is expected to fail.
CheckOutcomeFailed = true
```

Figure 9-30. Test Case Data File Example

(Source: TransCore ITS, LLC, February 2014.)

9.5.2.2 Special Parameter Values and Usage

There are a number of special values that can be used as parameter values. When a test is run, these special values are replaced as specified below.

1. USERDEFINED – When this special parameter is used as a parameter value, the user will be prompted for input when the test case is executed. An example of the usage for this parameter is as follows:

ResponseMessage = #USERDEFINED#

2. CONTEXTVARIABLE – When this special parameter is used as part of the parameter value, the enclosed context variable reference from the Test Configuration file is used. An example of the usage for this parameter is as follows:

WebServiceURL = #CONTEXTVARIABLE#RI_WEBSERVICEURL#CONTEXTVARIABLE#

The context variables that are defined are listed below.

- a. RI_HOSTNAME – The value entered in the host name field of the test configuration file.
- b. RI_IPADDRESS – The value entered in the IP Address field of the test configuration file.
- c. RI_PORT – The value entered in the Port field of the test configuration file.
- d. RI_USERNAME – The value entered in the user name field of the test configuration file.
- e. RI_PASSWORD – The value entered in the password field of the test configuration file.
- f. RI_WEBSERVICEURL – The value entered in the web service URL field of the test configuration file.

3. VALUESPEC – When this special parameter is used as part of the parameter value, the parameters within the GROUP name that it encloses are used as value specifications. A value specification defines a value or set of values that must be contained within a message. An example of the usage for this parameter is as follows:

VerificationSpec = #VALUESPEC#Values#VALUESPEC#

4. MESSAGESPEC – When this special parameter is used as part of the parameter value, the parameters within the GROUP name that it encloses are used as message values. Each parameter name and value referenced within the group define an element within a message. The order in which the parameters are consistent with the order as they are to be found within the actual message representation. An example of the usage for this parameter is as follows:

ResponseMessage = #MESSAGESPEC#DeviceInventoryMsg#MESSAGESPEC#

5. FILE – When this special parameter is used as part of the parameter value, the contents of the specified file are referenced. The contents of the file are expected to be ASCII. If a file is not specified, a file chooser dialog will open, allowing the user to choose a file. An example of the usage for this parameter is as follows:

RequestMessage = #FILE#TestMessage25.txt#FILE#

9.5.3 Edit the Test Configuration File Test Case Source

The test configuration file must be edited to reference the user developed test case data file. The steps to accomplish this are as follows:

1. Open a test configuration file.
2. Ensure that the user test case data file has been created.
3. Select either the Application Layer Test Cases or Information Layer Test Cases tab depending on which standard type the test case belongs to.
4. Find the test case in the table and select the associated cell in the Source column. When the test suite provided test case data will be used unchanged the text within this field should be set to “default.”
5. Enter the path and file name for the user created test case file in the cell.
6. Save the test configuration file.

9.5.4 Edit the Entity Emulation Data Message Files

The RI is installed with a full suite of default entity emulation data. This data may be modified to include accurate information for the OC. Example emulation data files are provided in the folder C2CRI/emulationDataFiles/OriginalEntityEmulation. Relevant files should be copied to the folder C2CRI/emulationDataFiles/CustomEntityEmulation and edited using a text editor to contain accurate information for the OC. The content of the file must represent a valid TMDD message for the entity data type. The user must take care to maintain the order of lines (representing message elements) in the file, specify valid message values, and ensure all mandatory elements are specified. The entity emulation sources can then be updated as described in Section 6.0: Define a Test Configuration.

9.6 Test Suite Creation

The C2C RI application includes pre-defined test suites for TMDD v3.03c, TMDD v3.03d, TMDD v.3.1, and NTCIP 2306. Any additional test suites created must follow the format specified below.

The following represents the hierarchy of the files required for a test suite. At the top of the hierarchy is the “BaseDirectory” which contains all of the test suite files and subfolders. The subfolder within the “BaseDirectory” folder will be AppLayer if the test suite is for an Application Layer Standard (such as NTCIP 2306). For an Information Layer Standard such as TMDD v3.03c the subfolder must be named InfoLayer. Within the AppLayer or InfoLayer subfolder, a Data subfolder exists to hold test case data files and a scripts subfolder exists to hold test procedure script files.

```

BaseDirectory\
  SuiteSpec.properties
  {AppLayer or InfoLayer}\
    NRTM.csv
    Predicates.csv
    TestCases.csv
    TestCaseMatrix.csv
    TestCaseDescriptions.xml
    TestProcedureDescriptions.xml
    TestProcedureSteps.xml
    Data\
      {test case name}.dat
  EmulationData\

```

{Entity Data Type}
Scripts\
{test case name}.xml

The following sections provide details on the internal files shown in the layout above.

9.6.1 SuiteSpec.properties

This file consists of a list of properties which identify the test suite and any applicable baseline test suite. Each line of the file contains a property followed by an “=” character, followed by the value of the property. Valid properties and values are specified below.

- SuiteName {String} – Unique name associated with this test suite. It is presented in the C2C RI GUI and is used as a reference for any custom test suites.
- SuiteVersion {String} – Unique version number associated with this test suite.
- InfoLayerStandard {String} – Defines the information Layer Standard associated with this test suite.
- AppLayerStandard {String} – Defines the application Layer Standard associated with this test suite.
- SuiteDescription {String} – Provides a description of the test suite.
- BaselineTestSuiteName {String} – The SuiteName of the test suite that serves as a baseline test suite for this test suite specification. If this test suite is a Pre-defined test suite, the Baseline Test Suite Name is the same as the SuiteName.
- BaselineTestSuiteVersion {String} – The SuiteVersion of the test suite that serves as a baseline test suite for this test suite specification. If this test suite is a Pre-defined test suite, the Baseline Test Suite version is the same as the SuiteVersion.
- TestSuiteType {Information or Application} – Indicates whether this test suite is associated with an Information or Application Level standard.

9.6.2 {AppLayer or InfoLayer}\NRTM.csv

Contains the data that relates the applicable standard (or project) Needs to the Requirements. The first row defines the header for this CSV file and must contain the following text:

“ID,UN ID,User Need,NeedConformance,NeedSupport,Requirement ID,Requirement,Conformance,Support,Other Requirements,ReqFlag,ReqFlagValue,OtherReqParameter,OtherRequirementValues,NeedFlag,NeedFlagValue”

All other records in the File must include entries for each of the fields identified in the header, where:

- ID {Integer} – A unique ID for this row within the CSV file.
- UN ID {String} – An identifier assigned to a particular need.
- User Need {String} – Text describing the need.
- NeedConformance {Mandatory or Optional} – Indicates whether the need is mandatory or optional.

- NeedSupport {String} – Provides information regarding whether the user may choose to select the need.
- Requirement ID {String} – A unique ID assigned to a particular requirement.
- Requirement {String} – Text describing the requirement.
- Conformance {String} – Text describing whether this requirement is necessary for conformance. If the field is set to Mandatory or M, then the requirement is necessary for conformance. Otherwise, the requirement is considered optional.
- Support {String} – Provides information regarding whether the user may choose to select the requirement.
- Other Requirements {String} – Provides text describing any additional requirements associated with the Requirement ID.
- ReqFlag {String} – A unique name provided for this Need ID/Requirement ID combination. This name may be referenced by test scripts to determine whether this requirement has been selected by the user.
- ReqFlagValue {true or false} – Indicates the default value of the ReqFlag defined parameter.
- OtherReqParameter {String} – A unique name provided for the Need ID/Requirement ID combination that represents a parameter defined for the Other Requirements description.
- OtherRequirementValues {String} – Indicates the default value of the OtherReqParameter defined parameter.
- NeedFlag {String} – A unique name provided for the Need that can be referenced by test scripts.
- NeedFlag Value {true or false} – Indicates the default value for the NeedFlag.

The text below is an example of valid NRTM.csv content. Note that an additional field – RequirementType – is included in this sample. Additional fields may be included within the CSV file, however they will not be used by the software.

```
ID,UN ID,User Need,NeedConformance,NeedSupport,Requirement
ID,Requirement,Requirement Type,Conformance,Support,Other
Requirements,ReqFlag,ReqFlagValue,OtherReqParameter,OtherRequirementValues,NeedFlag,Need
FlagValue
```

1,2,3,1,1,Verify Connection Active,Mandatory,Yes,3.3.1.1.1,Send Center Active Verification Upon Request,Dialogs,M,Yes,The owner center shall respond within ____ (100 ms – 1 hour; Default = 1 minute) after receiving the request. See Section
3.4.2.,TMDD_N1R1_Send_Center_Active_Verification_Upon_Request_Supported,True,TMDD_N1R1_Send_Center_Active_Verification_Upon_Request_Parameter,60000,TMDD_N1_Flag,True

2,2,3,1,1,Verify Connection Active,Mandatory,Yes,3.3.1.1.2,Publish Center Active Verification Information,Dialogs,Subscription:O,Yes / No / NA,The owner center shall begin sending the updated response message within ____ (100 ms – 24 hours; Default = 100 ms) after the information is updated in the owner center. See Section
3.4.1.,TMDD_N1R2_Publish_Center_Active_Verification_Information_Supported,False,TMDD_N1R2_Publish_Center_Active_Verification_Information_Parameter,900000,TMDD_N1_Flag,True

3,2,3.1.1,Verify Connection Active,Mandatory,Yes,3,3.1.1.3,Subscribe to Center Active Verification Information,Dialogs,Subscription:O,Yes / No / NA,
TMDD_N1R3_Subscribe_to_Center_Active_Verification_Information_Supported,False,TMDD_N1R3
_Subscribe_to_Center_Active_Verification_Information_Parameter,,TMDD_N1_Flag,True

9.6.3 {AppLayer or InfoLayer}\Predicates.csv

Contains the data that specifies the Predicates defined within this standard or extension of a standard.

The first row defines the header for this CSV file and must contain the following text:

"ID,Predicate,Section,ParentNeed"

All other records in the File must include entries for each of the fields identified in the header, where:

- ID {Integer} – A unique ID assigned to this predicate.
- Predicate {String} – The name assigned for this predicate.
- Section {String} – A reference to the Requirement ID that this predicate represents.
- ParentNeed {String} – If this predicate is specified within one need, but is intended to be used throughout all needs, then this field contains the parent Need ID. If not, then the value may be left empty or set to NULL.

The text below is an example of valid Predicates.csv content.

```
ID,Predicate,Section,ParentNeed  
1,Subscription*,3.3.1.3.1 OR 3.3.1.3.2,2,3.1.3  
2,AuthOrg,3.3.2.4.2.1,NULL  
3,ContactInformation,3.3.2.5.2.4,NULL  
4,CenterInformation,3.3.2.5.2.5,NULL
```

9.6.4 {AppLayer or InfoLayer}\TestCases.csv

Contains the list of test cases defined for this standard, or extension of a standard.

The first row defines the header for this CSV file and must contain the following text:

"Name,ScriptFile,DataFile,Description,Type"

All other records in the File must include entries for each of the fields identified in the header, where:

- Name {String} – The name of the test case.
- ScriptFile {String} – The name of the script file that executes the procedure related to this test case.
- DataFile {String} – The name of the test case data file that is associated with this test case.

- Description {String} – A description of the purpose of this test case.
- Type {OC or EC} – Indicates whether this test case should be made available when the RI is acting as an External Center (EC) or Owner Center (OC).

The text below is an example of valid TestCases.csv content.

Name,ScriptFile,DataFile,Description,Type

TCS-1-dlCenterActiveVerificationRequest-OC-Valid,TPS-1-dlCenterActiveVerificationRequest-OC.xml,TCS-1-dlCenterActiveVerificationRequest-OC-Valid.data,This test case is used to verify the SUTs support of the dlCenterActiveVerificationRequest dialog as an OC using the variable values specified by the Test Plan. This test case supports verification of requirements related to user need 2.3.1.1 [Verify Connection Active]. This Test Case tests for a valid response result.,EC

TCS-1-dlCenterActiveVerificationRequest-OC-InValid-1,TPS-1-dlCenterActiveVerificationRequest-OC.xml,TCS-1-dlCenterActiveVerificationRequest-OC-InValid-1.data,This test case is used to verify the SUTs support of the dlCenterActiveVerificationRequest dialog as an OC using the variable values specified by the Test Plan. This test case supports verification of requirements related to user need 2.3.1.1 [Verify Connection Active]. This Test Case tests for an invalid response result. An error response message is expected be returned with an error-code set to 1 (unknown processing error).,EC

TCS-1-dlCenterActiveVerificationRequest-OC-InValid-2,TPS-1-dlCenterActiveVerificationRequest-OC.xml,TCS-1-dlCenterActiveVerificationRequest-OC-InValid-2.data,This test case is used to verify the SUTs support of the dlCenterActiveVerificationRequest dialog as an OC using the variable values specified by the Test Plan. This test case supports verification of requirements related to user need 2.3.1.1 [Verify Connection Active]. This Test Case tests for an invalid response result. An error response message is expected be returned with an error-code set to 2 (center does not support this type message).,EC

9.6.5 {AppLayer or InfoLayer}\TestCaseMatrix.csv

Contains the data that specifies the Predicates defined within this standard or extension of a standard.

The first row defines the header for this CSV file and must contain the following text:

"ID,NeedID,RequirementID,TCID,ItemType,Precondition"

All other records in the File must include entries for each of the fields identified in the header, where:

- ID {Integer} – A unique integer representing this row of the file.
- NeedID {String} – The unique identifier assigned to a need that will be fully/partially satisfied by the referenced test case.
- RequirementID {String} – The unique identifier assigned to a requirement that will be full/partially satisfied by the referenced test case.
- TCID {String} – The name of the test case.

- ItemType {RI, SUT/RI, RI;SUT/RI} – This value indicates whether this entry is valid only for internal RI testing (RI), both SUT and RI testing (SUT/RI), or both internal and SUT testing (RI;SUT/RI).
- Precondition {String} – This value references any precondition predicate that should be selected before including this test case in the list of available tests. If no predicates apply, then this field should be left empty.

The text below is an example of valid TestCaseMatrix.csv content.

```
ID,NeedID,RequirementID,TCID,ItemType,Precondition
876152,2.3.1.1,3.3.1.1.1,TCS-1-dlCenterActiveVerificationRequest-EC-InValid-6,RI,
876153,2.3.1.1,3.3.1.1.1,TCS-1-dlCenterActiveVerificationRequest-EC-InValid-7,RI,
876154,2.3.1.1,3.3.1.1.1,TCS-1-dlCenterActiveVerificationRequest-EC-InValid-8,RI,
876155,2.3.1.1,3.3.1.1.3,TCS-1-dlCenterActiveVerificationSubscription-OC-Valid,SUT/RI,Subscription
876156,2.3.1.1,3.3.1.1.2,TCS-1-dlCenterActiveVerificationSubscription-OC-Valid,SUT/RI,Subscription
876157,2.3.1.1,3.3.1.1.5,TCS-1-dlCenterActiveVerificationSubscription-OC-Valid,SUT/RI,Subscription
```

9.6.6 {AppLayer or InfoLayer}TestCaseDescriptions.xml

Contains descriptions of the test cases, the requirements verification they support, and the expected response. The file is used to generate test case description reports without running tests.

The text below is an example of valid TestCaseDescriptions.xml content

```
<TestCaseDescriptions>
  <TestCaseDescription>
    <testCaseId>TCS-1-dlCenterActiveVerificationRequest-OC-Valid</testCaseId>
    <testCaseDescription>This test case is used to verify the SUTs support of the dlCenterActiveVerificationRequest dialog as an OC using the variable values specified by the Test Plan. This test case supports verification of requirements related to user need 2.3.1.1 [Verify Connection Active]. This Test Case tests for a valid response result.</testCaseDescription>
    <testCaseItems> 3.3.1.1.1 3.3.1.1.5.1 3.3.1.1.5.2.1 </testCaseItems>
    <testCaseInputs>Refer to the Test Case Data Variable Table in the appendix for the test case input parameters. </testCaseInputs>
    <testCaseInputProcedures>TPS-1-dlCenterActiveVerificationRequest-OC</testCaseInputProcedures>
    <testCaseOutputs>Each input execution shall generate an RI Test Result Status of Passed or Failed associated with the matching expected result shown in the Test Case Data Variable Table in the appendix. </testCaseOutputs>
    <testCaseOutputProcedures> </testCaseOutputProcedures>
    <testCaseHardware> See C2C RI SRS </testCaseHardware>
    <testCaseSoftware> See C2C RI SRS </testCaseSoftware>
```

```

<testCaseOther> None </testCaseOther>
<testCaseSpecial> None </testCaseSpecial>
<testCaseIntercase>None</testCaseIntercase>
</TestCaseDescription>
-<TestCaseDescription>
  <testCaseId>TCS-1-dlCenterActiveVerificationRequest-OC-Invalid-1</testCaseId>
  <testCaseDescription>This test case is used to verify the SUTs support of the
    dlCenterActiveVerificationRequest dialog as an OC using the variable values specified by
    the Test Plan. This test case supports verification of requirements related to user need
    2.3.1.1 [Verify Connection Active]. This Test Case tests for an invalid response result. An
    error response message is expected be returned with an error-code set to 1 (unknown
    processing error). </testCaseDescription>
  <testCaseItems> 3.3.1.1.1 3.3.1.4.1 3.3.1.4.1.1 3.3.1.4.1.2.1 </testCaseItems>
  <testCaseInputs>Refer to the Test Case Data Variable Table in the appendix for the test case
    input paramaters. </testCaseInputs>
  <testCaseInputProcedures>TPS-1-dlCenterActiveVerificationRequest-OC
    </testCaseInputProcedures>
  <testCaseOutputs>Each input execution shall generate an RI Test Result Status of Passed or
    Failed associated with the matching expected result shown in the Test Case Data
    Variable Table in the appendix. </testCaseOutputs>
  <testCaseOutputProcedures> </testCaseOutputProcedures>
  <testCaseHardware> See C2C RI SRS </testCaseHardware>
  <testCaseSoftware> See C2C RI SRS </testCaseSoftware>
  <testCaseOther> None </testCaseOther>
  <testCaseSpecial> None </testCaseSpecial>
  <testCaseIntercase>None</testCaseIntercase>
</TestCaseDescription>
</TestCaseDescriptions>

```

9.6.7 {AppLayer or InfoLayer}TestProcedureDescriptions.xml

Contains descriptions of the test procedures, the requirements verification they support, the test cases for which they are used (valid, invalid, or both), and the pass/fail criteria. The file is used to generate test procedure description reports without running tests.

The text below is an example of valid TestProcedureDescriptions.xml content.

```

-<TestProcedureDescriptions>
  -<TestProcedureDescription>
    <testProcedureName>TPS-1-dlCenterActiveVerificationRequest-OC </testProcedureName>
    <testProcedureId>Need 1 dlCenterActiveVerificationRequest for a SUT in OC Mode
      </testProcedureId>
    <testProcedureDescription>This test procedure is called by a test case and is used to verify
      the SUTs support of the dlCenterActiveVerificationRequest dialog as an OC using
      variables provided by the calling test case. This procedure supports verification of
      requirements related to user need 2.3.1.1 [Verify Connection Active] and is used for both
      valid and invalid test cases. </testProcedureDescription>
    <Requirements>3.3.1.1.1 3.3.1.1.5 3.3.1.1.5.1 3.3.1.1.5.2.1 3.3.1.4.1 3.3.1.4.1.1 3.3.1.4.1.2.1
      </Requirements>
    <Variables>ApplicationLayerStandard [String]
      TMDD_N1R1_Send_Center_Active_Verification_Upon_Request_Parameter [Integer]

```

```

TMDD_N1R8_Restrictions__Center_Active_Supported [Boolean]
TMDD_N1R14_Restrictions__Error_Report_Supported [Boolean] RequestMessage
[String] ErrorResponseExpected [Boolean] ErrorTypeExpected [String]
CONTENTVERIFIED [Boolean] </Variables>
<PassFailCriteria>The SUT shall pass every verification step included within the Test
Procedure in order to pass the Test Procedure. If a verification step fails, then test
execution will proceed at the next subsequent Post Condition step, if present. Otherwise,
test execution will proceed to the final Exit step. </PassFailCriteria>
</TestProcedureDescription>
-<TestProcedureDescription>
<testProcedureName>TPS-1-dlCenterActiveVerificationRequest-EC </testProcedureName>
<testProcedureId>Need 1 dlCenterActiveVerificationRequest for a SUT in EC Mode
</testProcedureId>
<testProcedureDescription>This test procedure is called by a test case and is used to verify
the SUTs support of the dlCenterActiveVerificationRequest dialog as an EC using
variables provided by the calling test case. This procedure supports verification of
requirements related to user need 2.3.1.1 [Verify Connection Active] and is used for both
valid and invalid test cases. </testProcedureDescription>
<Requirements>3.3.1.1.3 3.3.1.1.4 3.3.1.1.4.1 </Requirements>
<Variables>ApplicationLayerStandard [String]
TMDD_N1R1_Send_Center_Active_Verification_Upon_Request_Parameter [Integer]
ResponseMessage [String] AuthenticationCheck [Boolean] Username [String] Password
[String] OperatorID [String] AllowAccess [Boolean] ErrorResponseExpected [Boolean]
ErrorTypeExpected [String] DataValid [Boolean] CONTENTVERIFIED [Boolean]
</Variables>
<PassFailCriteria>The SUT shall pass every verification step included within the Test
Procedure in order to pass the Test Procedure. If a verification step fails, then test
execution will proceed at the next subsequent Post Condition step, if present. Otherwise,
test execution will proceed to the final Exit step. </PassFailCriteria>
</TestProcedureDescription>
</TestProcedureDescriptions>

```

9.6.8 {AppLayer or InfoLayer}TestProcedureSteps.xml

Contains the test procedure count (incrementing), test step numbers, procedure details, and procedure results. The file is used to generate test procedure description reports without running tests.

The text below is an example of valid TestProcedureSteps.xml content.

```

-<TestProcedureSteps>
-<TestProcedure>
<procedureNumber>1</procedureNumber>
<testProcedureName>TPS-1-dlCenterActiveVerificationRequest-OC</testProcedureName>
-<step>
<stepCount>1</stepCount>
<testStepNumber>1 </testStepNumber>
<testProcedure>CONFIGURE: Determine the Application Layer Standard that will be
used for this test. RECORD this information as: ApplicationLayerStandard
</testProcedure>
<TestProcedureResults>NA </TestProcedureResults>

```

```
</step>
-<step>
  <stepCount>2</stepCount>
  <testStepNumber>2 </testStepNumber>
  <testProcedure>CONFIGURE: Determine the dialog performance requirement for Send
    Center Active Verification Upon Request (NRTM 3.3.1.1.1). RECORD this value as:
    TMDD_N1R1_Send_Center_Active_Verification_Upon_Request_Parameter
  </testProcedure>
  <TestProcedureResults>NA </TestProcedureResults>
-<step>
  <stepCount>3</stepCount>
  <testStepNumber>3 </testStepNumber>
  <testProcedure>CONFIGURE: Determine whether Restrictions - Center Active is
    required by the specification. (NRTM 3.3.1.1.5.2.1). RECORD this information as:
    TMDD_N1R8_Restrictions__Center_Active_Supported </testProcedure>
  <TestProcedureResults>NA </TestProcedureResults>
-<step>
...
</TestProcedure>
</TestProcedureSteps>
```

9.6.9 {AppLayer or InfoLayer}\Data

This subfolder contains each test case data file for the test suite. The format of these files is described in Section 9.5.1.

9.6.10 {InfoLayer}\EmulationData

This subfolder contains each Entity Data File for the test suite. The Entity Data File contains the initial set of entity information that will be used in testing. Typically, an entity file will exist for each type of information message that is defined for an entity. For example, in TMDD CCTV entity information is provided through the cCTVInventoryMsg and cCTVStatusMsg messages. Emulation Data Files would be created as CCTVINVENTORY and CCTVSTATUS.

Each entity file contains a set of parameter entries that represent each value that will appear in the response message. The parameters must be defined in the order as they appear within the message.

Each test case parameter entry is defined with the following format:

ELEMENTNAME = ELEMENTVALUE

ELEMENTNAME represents that name that has been assigned to the parameter and ELEMENTVALUE is the value that will be assigned.

9.6.11 {AppLayer or InfoLayer}\Scripts

This subfolder contains each Test Script file associated with the test suite.

10.0 Error Messages and Problem Resolution

The RI provides standard messages when performing file management activities such as opening, saving, and closing the configurations and creating test logs and reports.

10.1 Overwriting a Test Log File

The system prompts the user with the dialog similar to the one shown in [Figure 10-1](#)[Figure 10-4](#) when the user chooses to overwrite the name for a test report file. Click the **Yes** button to confirm the overwrite operation. Click the **No** button to cancel the operation.

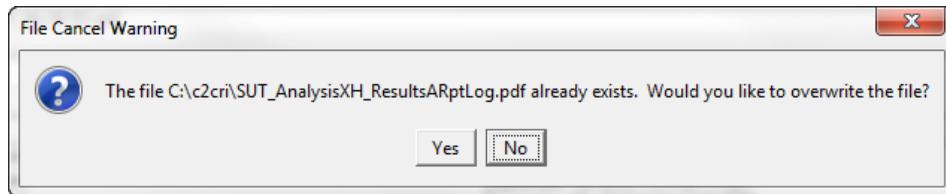


Figure 10-1. Overwriting a Test Log File Confirmation Dialog
(Source: TransCore ITS, LLC, February 2014.)

10.2 Cancelling Changes to Test Configuration File

The system prompts the user with the dialog similar to the one shown in Figure 10-2 when the user chooses to cancel the Test Configuration Editing Window. Click the **Yes** button to close the Test Configuration Editing Window. Click the **No** button and the Test Configuration Editing Window will remain open.

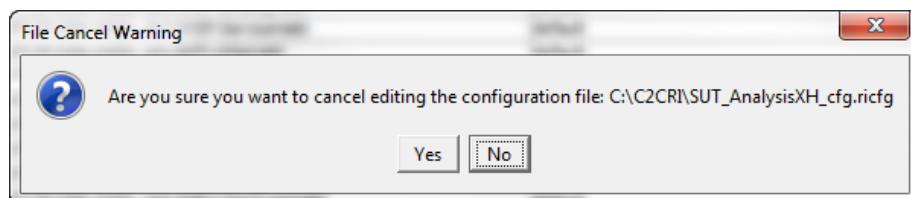


Figure 10-2. Cancelling File Changes Confirmation Dialog
(Source: TransCore ITS, LLC, February 2014.)

10.3 Closing a Test Configuration File

The system prompts the user with the dialog similar to the one shown in Figure 10-3 when the user chooses to close the Test Configuration Editing Window. Clicking the **Yes** button will save user changes to the Test Configuration file and then close the Test Configuration Editing Window. Click the **No** button to abort the operation to close the Test Configuration Editing Window.

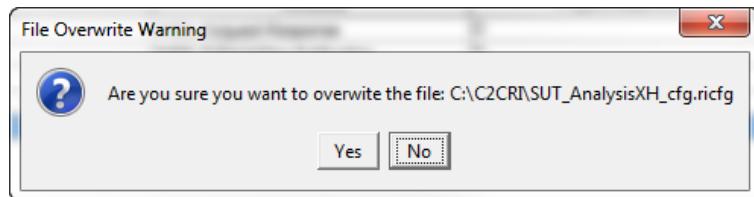


Figure 10-3. Closing a File Confirmation Dialog
(Source: TransCore ITS, LLC, February 2014.)

10.4 Error Message Displayed When Insufficient Need or Requirement Selected

When a Test Configuration File is not configured with any application layer need or requirements then the dialog similar to the one shown in Figure 10-4 is displayed when the user chooses to save the file.

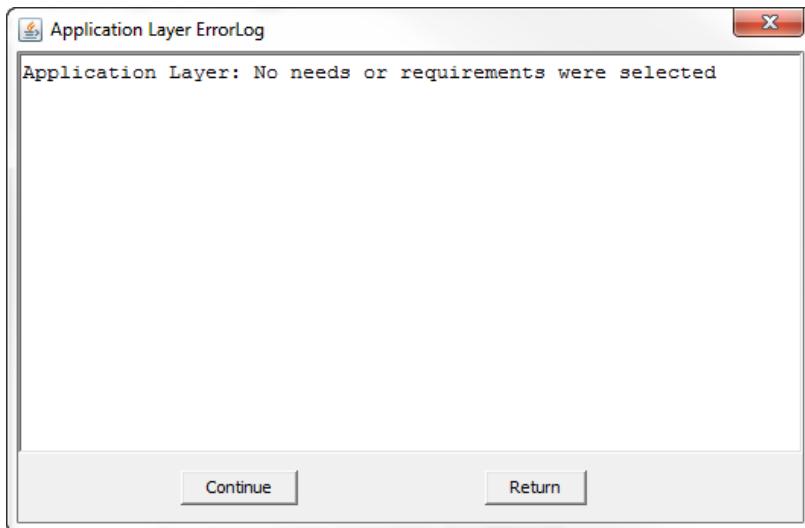


Figure 10-4. Insufficient Need and Requirement Error Log

(Source: TransCore ITS, LLC, February 2014.)

Click the **Continue** button and a dialog similar to the one shown Figure 10-2 will be displayed. Click the **Return** button and a dialog similar to the one shown in Figure 10-5 will be displayed. When the **OK** button is clicked the Test Configuration Editing Window will remain open.

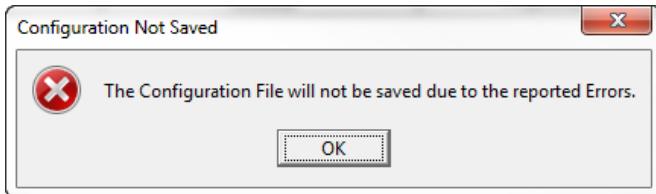


Figure 10-5. Configuration Not Saved Window

(Source: TransCore ITS, LLC, February 2014.)

10.5 Saving changes to a Test Configuration File

The system prompts the user with the dialog similar to the one shown in Figure 10-3 when the user chooses to save changes that were made to the Test Configuration file. Clicking the **Yes** button will confirm the operation to save any changes that the user made to the Test Configuration File. to the file. Click the **No** button to abort the operation and no changes will be saved to the Test Configuration File.

10.6 Using the 32-bit Version of the C2C RI Application

The recommended version of the C2C RI Application is the 64-bit version; however, a 32-bit version of older releases is also available. The 32-bit version has a known limitation on the size of log files which can be managed to create reports. The C2C RI will present an error message as shown in Figure 10-6 if the test log file exceeds the manageable size. The test log file can be opened in the 64-bit version to create reports.

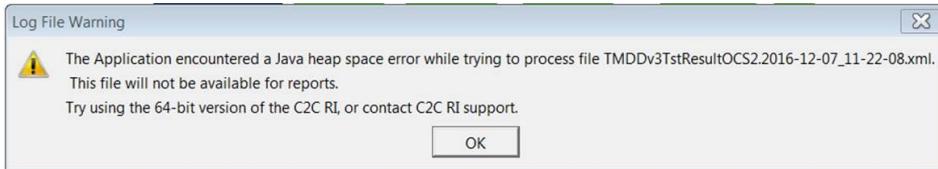


Figure 10-6: 32-bit File Size Warning Window

(Source: TransCore ITS, LLC, February 2017.)

10.7 Deleting Files from the C2C RI Application

10.7.1 How to delete a Test Configuration File

Users are able to use the C2C RI application to remove Test Configuration files from the system that are no longer needed. These can be any Test Configuration file stored in the C2C RI installation directory or any other directory location defined by the user.

To remove a Test Configuration File from the system:

1. Select the **Tools** menu and select: **Maintenance** and the dialog shown in [Figure 10-7](#) will open.
2. Select the: **Configuration Files** radio button.
3. Select a Test Configuration File from the list.
4. Click the **Delete** button and a dialog similar to the one shown in [Figure 10-7](#) will be displayed.

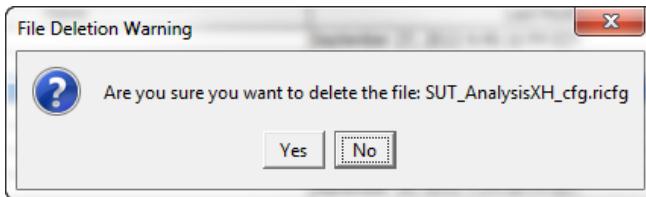


Figure 10-7. Test Configuration File Deletion Confirmation Dialog
(Source: TransCore ITS, LLC, February 2014.)

5. Click the **Yes** button to confirm the delete operation and the Test Configuration File will be deleted from the system. Clicking the **No** button will abort the delete operation.

10.7.2 How to delete a Test Log File

Users are able to use the C2C RI application to remove Test Log files from the system that are no longer needed. These can be any Test Log file stored in the C2C RI installation directory or any other directory location defined by the user.

To remove a Test Log File from the system:

1. Select the **Tools** menu and select: **Maintenance** and the dialog shown in Figure will open.
2. Select the: **Log Files** radio button.
3. Select a Test Log File from the list.
4. Click the **Delete** button and a dialog similar to the one shown in [Figure 10-8](#) will be displayed.

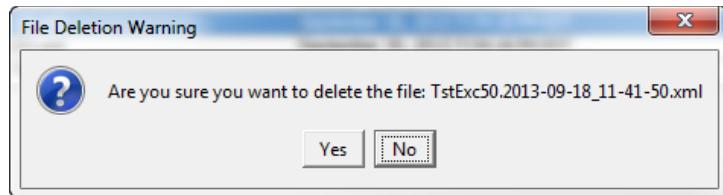


Figure 10-8. Test Log File Deletion Confirmation Dialog

(Source: TransCore ITS, LLC, February 2014.)

5. Click the **Yes** button to confirm the delete operation and the Test Log File will be deleted from the system. Clicking the **No** button will abort the delete operation.

11.0 Glossary

Term	Meaning
IEEE 1512	A series of standards for incident management and traffic incident related messages relating to public safety and emergency management centers and services.
AASHTO	American Association of State Highway and Transportation Officials
Application Layer Standards	ITS C2C standards which define how conversations between centers are managed, how data transferred is encoded, and the syntax and grammar rules for the conversation.
ATIS	Advance traveler information system
C2C	Center-to-center
CCTV	Closed circuit television
Conformance Testing	The execution of a set of tests to verify that an implementation meets the requirements of an accepted standard.
ConOps	Concept of operations
Compliance Testing	The execution of a set of tests to verify that an implementation satisfies the requirements for a specific project procurement.
COTS	Commercial off the shelf
CSV	Comma separated value file format
DATEX	A general purpose C2C data exchange protocol stack.
DOT	Department of Transportation
Entity	A device or other item controllable by an owner center
External Center (EC)	A traffic management center that requests data and sends control to an owner center of traffic devices.
FHWA	Federal Highway Administration
Html	Hypertext markup language
IEEE	The Institute of Electrical and Electronics Engineers
Information Layer Standards	ITS standards for data elements, objects, and messages to be transmitted between ITS entities.
ITE	Institute of Transportation Engineers
ITS	Intelligent transportation systems
JAR	Java archive
LRMS	Location reference message specification
MS/ETMCC	Message sets and external traffic management center communication
NEMA	National Electrical Manufacturers Association
NTCIP	National transportation communications for ITS protocol
Owner Center (OC)	A traffic management center that has direct control of the traffic devices and provides an external interface to the devices.
PDF	Portable document format
pre-defined tests	Standards conformance tests included with the RI.

11. Glossary

Term	Meaning
Requesting center	A center that uses the center-to-center services provided by an owner center.
RI	C2C Reference Implementation
RI-SUT Communications interface	The communication interface that supports communications between the RI and the SUT.
SAE	Society of Automotive Engineers
SOAP	Simple object access protocol
Section 1201	Section 1201 of Safe, Accountable, Flexible, Efficient Transportation Equity Act: A Legacy for Users (SAFETEA-LU)
SRS	Software requirements specification
SUT	System under test
TCP/IP	Transmission control protocol / internet protocol
Test Configuration	The set of parameters, scripts and settings which are needed to establish a conformance test.
TMDD	Traffic management data dictionary
UDP	User datagram protocol
User-Defined Tests	Standards conformance tests which are created by the RI users
URI	Universal resource identifier
URL	Universal resource locator
WSDL	Web services description language
W3C	World Wide Web Consortium
XML	Extensible markup language

U.S. Department of Transportation
ITS Joint Program Office-HOIT
1200 New Jersey Avenue, SE
Washington, DC 20590

itshelp@volpe.dot.gov
www.its.dot.gov

FHWA JPO-XX-XXX



U.S. Department of Transportation