

Load Dynamix

Release 5.6 TDE Help

Table of Contents

Load Dynamix Help

Help	3
TDE and Appliance Introduction.....	4
Technology References and Terminology	13
Product Installation	16
Test Development Environment and GUI.....	39
Test Creation.....	81
Executing Tests and Assessing Results.....	123
Advanced Concepts: User Parameters	158
Advanced Concepts: Data File Systems & Data Verification.....	187
Appendix: Jumbo Frames and Delayed ACK.....	199
Advanced Concepts: Response Handling	204
Advanced Concepts: Variables and Aliases.....	210
Advanced Concepts: Test Execution Rules.....	216
Advanced Concepts: Chained Commands.....	235
Advanced Concepts: Threads and Async Operations.....	239
Troubleshooting Projects	245
Tips and FAQ	253
Appendix: Office.....	262
Appendix: Test Automation and LDX-E Integration.....	264
Appendix: Client Side DFS Support.....	287
Appendix: Scenario Control Actions	289
Appendix: NFS v3, v4 and v4.1 Notes.....	303
Appendix: Sample Projects.....	321
Appendix: TCP/UDP Echo and Discard Protocols	324
Appendix: Functions and Formula.....	326
Appendix: Max TCP Open Connections and Open Rate	338
Appendix: IPv6.....	346
Appendix: DNS and UDP Protocols	348
Appendix: DCB/DCBx	360
Appendix: NTLM Flags	365
Appendix: Change Notify and Change Notify Cancel Actions.....	368
Appendix: Virtual Appliance Constraints & Licensing	379
Reference: CIFS/SMB Commands and Behaviors.....	389
Reference: SMB2 Commands and Behaviors	394
Reference: SMB 3.0 Commands and Behaviors.....	415
Reference: NFSv2 Command List	425
Reference: NFSv3 Commands and Behaviors.....	427
Reference: NFSv4, v4.1 Command List.....	432
Reference: Kerberos v5 Command List.....	439
Reference: Load Dynamix HTTP/HTTPS Commands and Behaviors	440
Reference: iSCSI Commands and Behaviors.....	472
Reference: FC/iSCSI/SCSI Commands and Behaviors.....	487
Reference: TCP/UDP Echo and Discard Protocols Command List.....	574
Reference: Action Input Shorthand.....	575

Reference: HTTP Storage Commands and Behaviors	577
Reference: End of Life Statement	591

Test Development Environment Online Help



Release 5.6
Build 0.56.4xxxx

Copyright © 2008-2017 Load DynamiX Inc.

[Load DynamiX Support](#)

TDE and Appliance Introduction

TDE and Appliance Introduction

The Load DynamiX software and Workload Generation Appliances (from here on out just Appliances) are the leading storage industry test tools used to evaluate the performance and scalability of networked and object storage and storage-aware devices/services. The Load DynamiX Client simulates large numbers of clients accessing networked storage using the Fibre Channel, HTTP, HTTPS, CIFS-SMB, SMB2, SMB3, iSCSI or NFSv2/v3/v4/v4.1 protocols or large numbers of clients accessing HTTP storage using the HTTP/HTTPS, Amazon S3, CDMI, OpenStack Swift or OpenStack Cinder protocols. The Load DynamiX Server simulates large HTTP, HTTPS, CIFS-SMB, SMB2, iSCSI and NFSv3 file server installations. Both Client and Server run on the Load DynamiX Appliance under the control of the Load DynamiX Test Development Environment, Load DynamiX Enterprise (LDX-E), Load DynamiX Automation or Load DynamiX API.

The Load DynamiX Test Development Environment (TDE) is a graphical software application that allows users to configure and run tests (Projects or Workloads) on the Load DynamiX Appliance, analyze test results and create tests that can be executed under the control Load DynamiX Automation or LDX-E or Load DynamiX API. The Load DynamiX Appliance executes scripts created by the TDE or by the API or by LDX-E, all of which can be initiated by the TDE, LDX-E, LdxCmd or the Load DynamiX API.

Windows .NET framework version 4.x must be installed - see [Product Installation section for details.](#)

Purpose

This guide provides information on how to use the Load DynamiX Appliances to test IP and Fibre Channel networked storage/object devices and services. See the [Troubleshooting](#) and [Tips and FAQ](#) chapters for project design and debugging guidance.

Scope and Audience

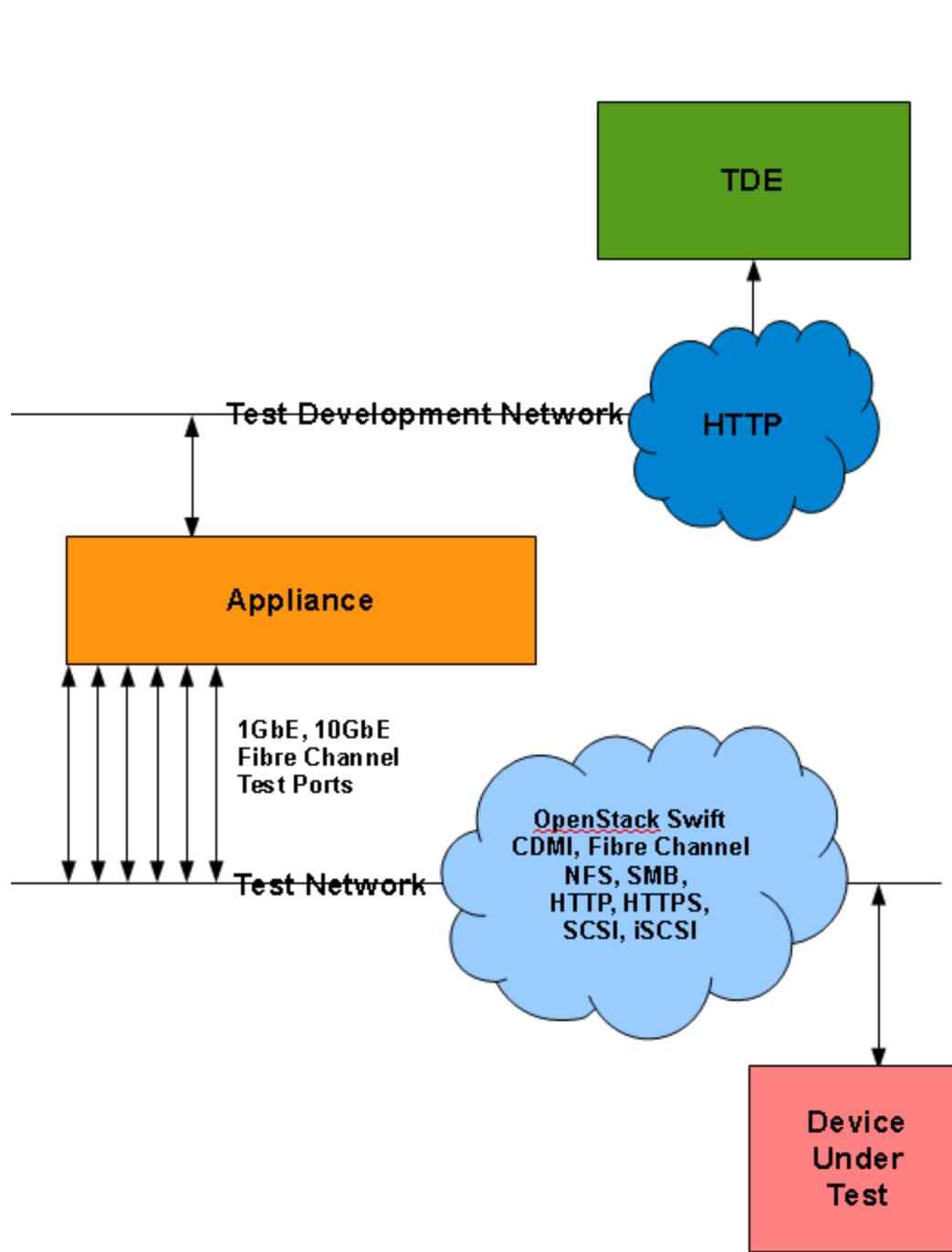
This guide is intended to be used by development and test engineers who have a need to verify the storage/object capabilities of various IP network devices and services. It is assumed that these individuals are familiar with:

- Ethernet and Fibre Channel networks
- Networked storage and object storage test methodologies
- CIFS-SMB, SMB2, SMB3 and NFSv2,v3,v4,v4.1 file-oriented protocols and file servers
- iSCSI block-oriented protocol and devices
- Fibre Channel and Fibre Channel over Ethernet block-oriented protocol and devices
- SCSI block protocol
- Kerberos, NTLM, CHAP and other network authentication protocols
- HTTP and HTTPS protocols
- HTTP Storage: Amazon S3, Keystone Identity Service v2/v3, CDMI, Cinder and OpenStack Swift protocols

This guide contains information on the capabilities of the Load DynamiX Test Development Environment and Appliances for testing networked storage/object devices and services, and focuses on the unique features of the products that support test development and execution. The Load DynamiX Appliances adhere to multiple standards such as IPv4, IPv6, iSCSI, CIFS-SMB, SMB2, SMB3, Kerberos, Fibre Channel, HTTP/S, NFSv2/v3/v4.1, Amazon S3, OpenStack, CDML, etc. but this guide does not describe the complete details of any of these protocols. Links to protocol reference materials are provided in the [References and Terminology section](#).

Load DynamiX Deployment Model

The following diagram shows the typical deployment model for the Load DynamiX development software (the TDE), the Load DynamiX execution environment (the Appliance) and the device or devices that are being tested.



The Load DynamiX TDE is used to develop test Projects that will later be executed on the Load DynamiX Appliance to test the Device Under Test. A test is created to exercise the NFS, HTTP/HTTPS, SMB1/2/3, CDMI, OpenStack Swift, Amazon S3, OpenStack Cinder, Fibre Channel or iSCSI capabilities of the Device Under Test and then is delivered to the Load DynamiX Appliance from the TDE using the HTTP protocol. The test Project consists of one or more Logical Ports that contain Fibre Channel or IPv4/IPv6 networks (defined by Network Profiles) and test content (Scenarios that contain Actions) that the Load DynamiX Appliance executes against the Device Under Test. Logical Ports can be configured to appear to be 1 to very large numbers of individual Clients. Each Logical Port in a Project uses one of the Physical Ports on the Load DynamiX Appliance. A single Physical Port on the Load DynamiX Appliance can simulate thousands and thousands of end users accessing the Device Under Test.

This document explains the basic concepts necessary to develop and execute Projects, interpret Project results and trouble shoot any issues encountered in creating or executing Projects but does not attempt to document all network, Protocol or device behaviors. When there is a Load DynamiX extension to a documented protocol, this document will attempt to point out the extension and how to use it. In the text of this document, Actions (those elements of the Toolbox that are used to create Scenarios) are highlighted in **BOLD**. Protocol level concepts and terminology, where relevant to the discussion, are highlighted in "...".

How To Pointers

- Create a test Project: [Test Development Environment and GUI](#) and [Test Creation](#).
- Execute a Project and assess the results: [Executing Tests and Assessing Results](#).
- Troubleshooting Project failures and results: [Troubleshooting Projects](#) and [Tips and FAQ](#).
- Delivering input to Actions dynamically in a Project: [User Parameters](#).
- Verifying Read and Write Action contents in a Project: [Data File Systems and Data Verification](#).
- Automate Project execution in a Windows or Linux environment: [Test Automation](#).
- Delivering Action input using Functions: [Functions and Formula](#).
- Action Optimizations: [Chained Commands](#) (CIFS-SMB/SMB2), Number of Outstanding Requests ([NFS](#) and [SCSI](#)).
- Controlling Scenario behavior: [Threads and Async Operations](#), [Scenario Control Actions](#), [Test Execution Rules](#) and [Response Handling](#).
- Creating and using reusable Input values in a Project: [Variable and Aliases](#).
- Finding the maximum open TCP connections and connection open rate for a DUT: [Max TCP Open Connections and Open Rate](#).
- Scenario synchronization/concurrency control: [Scenario Control Actions](#).
- For a list of the kinds of information typically required to design a Load DynamiX Project, see the Information Typically Required for Project Design section below.

Feature Highlights

The Load DynamiX Appliances products are a combination of a Windows software application for test development, execution and analysis, and an Appliance that is the execution engine. The Test Development Environment (Load DynamiX TDE) allows development and test engineers to create tests for networked storage/object devices or services that they develop or need to evaluate. Tests are built from a set of common Actions such as communications channel creation, establishing credentials, file operations (open/read/write/close), loop control, test parameter control, [IPv4](#) or [IPv6](#) or [Fibre Channel](#) networks, etc.

The order and structure of tests created in the Load DynamiX TDE are a function of the target protocol

(e.g. CDMI, OpenStack Swift, HTTP, iSCSI, CIFS-SMB, SMB2 and NFS tests require completely different command sets and command order), and the intended purpose of the test (e.g. a test to evaluate file open/close scalability is very different from a test to evaluate read performance in a WAN acceleration environment). The TDE provides the framework (Timeline and Ports, Network Profiles and Scenarios), low level components (protocol and Scenario Control Actions) and output (graphical and text results files) necessary to create, execute and evaluate tests in networked storage/object services environments. Test execution can be controlled using Test Execution Rules (see [Test Creation](#) and [Test Execution Rules](#)) as well as the [Response Handling](#) (for NFS, CIFS-SMB, SMB2, HTTP, HTTP Storage, SCSI Actions). Input to tests can be delivered at execution time using [User Parameter](#) files or [Functions and Formula](#) or manual input to Actions or from string or numeric [Variables and Aliases](#) created in Scenarios.

The Load DynamiX TDE is a very powerful tool. It is used to create tests for a variety of network storage environments, execute these tests and evaluate the results. Load DynamiX Automation can also be used to execute tests. Testers can have multiple instances of the TDE GUI running at the same time on the same Windows platform, interacting with the same or different instances of the Appliance test platform thus being able to control multiple tests from the same desktop or laptop. It is possible to have multiple different versions of the TDE installed on the same desktop/laptop and interact with different Appliances running different versions of the Load DynamiX Appliance Firmware.

Results from test execution are captured in time-stamped results folder for review and analysis or visual comparison with results from past runs. Results are captured in the same My Projects folder where tests are stored. Packet flow can be captured (from the beginning of the Project or in a circular buffer format) in PCAP format and reviewed with Wireshark or other PCAP compatible tools.

Once tests have been developed and debugged, they can be executed via the command line or various scripting languages such as Perl or TCL. See [Test Automation](#) and for a more detailed discussion of how to automate the execution of Load DynamiX TDE developed tests.

Significant Enhancements and New Features in Release 5.6

- Templates for CDB Actions added to Custom Action Toolbox. See [Reference: FC/SCSI/iSCSI Commands and Behaviors](#) for details.
- Check Condition return value Response Handling processing can be controlled by the Custom extension feature. See [Reference: FC/SCSI/iSCSI Commands and Behaviors](#) for details.
-

Information Typically Required to Design a Load DynamiX Project

Some pieces of information that will be helpful to know before beginning to design a Project using the Load DynamiX TDE

Device Under Test IP Address(es): Load DynamiX Projects require a device under test to connect to and that device is specified by its IP Address. Is just a single IP address required (for example, NFSv3 tests require three separate connections, are these connections to the same IP Address)?

Device Under Test WWPN: Target device identifier in a Fibre Channel environment. Are they provisioned in the SAN switch to be accessed by the Load DynamiX Appliance?

Protocol: Which Protocol (CIFS-SMB, SMB2, NFSv2, NFSv3, NFSv4, NFSv4.1, iSCSI, HTTP, HTTPS, Fibre Channel, FCoE, OpenStack Swift, CDMI) is to be tested?

TCP Ports: Which TCP Ports will be used (CIFS-SMB:445, SMB2:445, NFSv2:2049,

NFSv3:111/627/2049, NFSv4:2049, iSCSI:3260, HTTP:80 or others)?

Domain/Machine Name: CIFS-SMB and SMB2 have fields for Domain and Machine Name in their authentication commands. What values are to be used?

Authentication Method: All Load DynamiX supported protocols require some form of authentication. Be sure to know what authentication method the device under test uses to authenticate access.

Users: How many and what are the User names and passwords that will be used during the test.

Filesystems and Files: Most Load DynamiX protocols are file-oriented (CIFS-SMB, SMB2, NFSv2, NFSv3, NFSv4, HTTP). What filesystem (volume or share name) and files are going to be used?

Are the files going to be created or must they exist in advance? What type of files are required (regular, device, link, pipe, stream, etc)?

URI: The HTTP protocol requires URI information to access files.

IQN: The iSCSI Protocol requires iSCSI Qualified Names for Clients to log in to iSCSI Servers.

LUN: The SCSI Protocol requires specific Logical Unit Numbers for data read and/or write operations.

Size: How many megabytes, terabytes, gigabytes are going to be required for the test. Does the filesystem or LUN have the capacity that is required?

Load: How much load does the test need to place on the device under test?

Duration: How long should the test run? (Maximum test duration is 1000 hours)

Throughput: What throughput (packets/sec, kilobytes/sec, etc) is required or desired?

NPIV Initiator WWPN: Are virtual initiator WWPN being used or the physical WWPN assigned to the port? Are they provisioned in the SAN switch to be used by the Load DynamiX Appliance and access the Target WWPN?

Information Required Decision Matrix

Protocol	IP Addr	Ports	Dom/Mach Name	Auth.	User Names	File Systems	Files	URI	IQN	NPIV WWPN	WWPN	LUN	Size	Load	Duration	Through
CIFS-SMB	YES	YES	YES	YES	YES	YES	YES	NO	NO	NO	NO	NO	YES	YES	YES	YES
SMB2	YES	YES	YES	YES	YES	YES	YES	NO	NO	NO	NO	NO	YES	YES	YES	YES
NFSv2	YES	YES	NO	YES	YES	YES	YES	NO	NO	NO	NO	NO	YES	YES	YES	YES
NFSv3	YES	YES	NO	YES	YES	YES	YES	NO	NO	NO	NO	NO	YES	YES	YES	YES
NFSv4/v4.1	YES	YES	NO	YES	YES	YES	YES	NO	NO	NO	NO	NO	YES	YES	YES	YES
iSCSI	YES	YES	NO	YES	NO	NO	NO	NO	YES	NO	NO	YES	YES	YES	YES	YES
HTTP/S	YES	YES	NO	YES	YES	YES	YES	YES	NO	NO	NO	NO	YES	YES	YES	YES
FC/FCoE	NO	YES	NO	NO	NO	NO	NO	NO	NO	YES	YES	YES	YES	YES	YES	YES
HTTP Storage	YES	YES	NO	YES	NO	YES	YES	NO	NO	NO	NO	NO	YES	YES	YES	YES

Product summary

Capability	Description

Protocols	<ul style="list-style-type: none"> CIFS-SMB/SMB2/SMB3 Client and Server over TCP & NetBIOS, CIFS-SMB/SMB2 Asynchronous I/O Kerberos version 5, Keystone identity service SMB Dialects: All, including SMB2 2.02, 2.1, 3.0 and 3.1.1 NFS: v2, v3 and v4, v4.1 , pNFS, Client; v3 Server, Asynchronous I/O iSCSI Initiator and Target (RFC 3720), iSCSI/Fibre Channel Asynchronous I/O Fibre Channel and FCoE HTTP/HTTPS, HTTP/HTTPS Asynchronous I/O TCP/UDP Echo and Discard Protocols IPv4 and IPv6 HTTP Storage: Amazon S3, CDMI, Open Stack Cinder and OpenStack Swift, HTTP Storage Asynchronous I/O
Network configuration and control	<ul style="list-style-type: none"> IPv4,IPv6 Multiple IPs MAC Emulation VLAN emulation ARP , DNS and UDP, TCP SACK Delayed ACK Adjustable MTU size DCB/DCBX Port Delay Fibre Channel MPIO, ALUA and NPIV support
Scenarios	<ul style="list-style-type: none"> Configurable parameters including: <ul style="list-style-type: none"> SCSI, CIFS-SMB, SMB2, Kerberos v5, NFSv2, NFSv3, NFSv4, NFSv4.1, HTTP/HTTPS , CDMI, OpenStack Swift, Amazon S3, FC/FCoE Commands CIFS-SMB/SMB2 Chained Command support for clients and servers User-defined parameters Configurable pass/fail criteria Scenario Control Actions (Events, Threads, Loops) Functions (RANDOM, STRING, VARIABLE, LOOPINDEX, LOOPTOTAL, SCENARIOCOUNTER) Formula Project execution control using Test Execution Rules and Threads and Async Operations
Load Profile	<ul style="list-style-type: none"> Concurrent Scenarios (typically Users), Actions or TCP Connections New Per Second Scenarios (typically Users), Actions or TCP Connections New Scenarios per Interval of time Defines client or server load over time
User interface	<ul style="list-style-type: none"> Graphical application based on .NET Runs on a Windows workstation
Reporting/Troubleshooting	<ul style="list-style-type: none"> Network and protocol statistics CSV result export; Report wizard Network Traffic capture during a test
File system simulation	<ul style="list-style-type: none"> Windows, Unix Random, Sequential, Seeded Random and Physical File data sources
Automation	<ul style="list-style-type: none"> Execute Projects from Windows or Linux command prompt Share projects between the TDE and Load DynamiX Enterprise
Authentication	<ul style="list-style-type: none"> NTLM (CIFS-SMB, SMB2, HTTP/HTTPS, HTTP Storage), Kerberos (CIFS-SMB, SMB2, NFS, HTTP/HTTPS, HTTP Storage), HTTPS, AWS2, AWS4, Keystone Identity Service, iSCSI 1-WAY/2-WAY CHAP, NFSv4 ACL

Platforms	<ul style="list-style-type: none"> • Load DamiX 1G Series models 3000 and 3108: eight 1000BASE-T ports • Load DamiX 10G Series models 5000 and 5102: two 10Gbps SFP+ ports (optical fiber or active/passive direct attach copper cables) • Load DamiX FC Series model 6202/6204/6208: two, four or eight 8x16Gbps FC optical fiber SFP+ ports or FC Series model 6202E 2x10Gbps FCoE optical fiber SFP+ ports • Load DamiX 10G Series model 5108T eight 10GBASE-T ports • Load DamiX 10G Series model 5108S eight 10Gbps SFP+ ports (optical fiber or active/passive direct attach copper cables) • Load DamiX Unified Series U1022 and U1044 two or four 16Gbps FC optical fiber SFP+ ports and 2 or 4 10GBASE-T/10Gbps SFP+ optical ports • Thousands of simulated concurrent networked and object storage users • Multi-User (multiple Testers may share one Appliance) • 2U Appliance • See http://www.LoadDamiX.com/products for Load DamiX product details
-----------	--

Information Organization

Information in this document is organized as follows:

Chapters - introduction to or a detailed explanation of the capabilities of the product

Advanced Concepts - a drill down on complex or new TDE or protocol features

Appendices - details of TDE or protocol features not covered by a Chapter or Advanced Concept

References - alphabetically sorted lists of the commands by protocol supported by the Load DamiX TDE

Table of Contents

[Chapter 1 TDE and Appliance Introduction](#) – This chapter, an introduction to product capabilities

[Chapter 2 Technology References and Terminology](#) – Standards references and acronyms/terms used

[Chapter 3 Product Installation](#) – Software and Hardware Installation notes

[Chapter 4 Test Development Environment and GUI](#) – A detailed product overview and how to use the GUI

[Chapter 5 Test Creation](#) – Describes test components and how to use them to create a test

[Chapter 6 Executing Tests and Assessing Results](#) – How to execute tests and view results

[Advanced Concepts: User Parameters](#) – How to create and use User Parameters

[Advanced Concepts: Data File Systems & Data Verification](#) – How create and verify file data

[Advanced Concepts: Response Handling](#) – How to use Response Handlers to control Scenario behavior

[Advanced Concepts: HTTP/HTTPS](#) – Using the HTTP and HTTPS protocols in Projects

[Advanced Concepts: Variables and Aliases](#) – Using Variables in Action input fields and how to use Aliases

[Advanced Concepts: Test Execution Rules](#) – Using Test Execution Rules to manage Project execution behaviors

[Advanced Concepts: Chained Commands](#) – Creating Chained Commands (CIFS-SMB) and Compound Requests (SMB2)

[Advanced Concepts: Threads and Async Operations](#) – Using Actions in Threads and Asynchronous blocks

[Chapter 7 Troubleshooting Projects](#) – Troubleshooting tricks

[Chapter 8 Tips and FAQ](#) – Tips and answers to some frequently asked questions

[Appendix: Office](#) – Microsoft Office tools delivered with the TDE

[Appendix: Test Automation and LDX-E Integration](#) – How to use and extend Load DamiX Automation and how to integrate the TDE with an Load DamiX Enterprise server

- [Appendix: Client Side DFS Support](#) – Support for Windows Distributed File System
- [Appendix: Scenario Control Actions](#) – Scenario Control Action details
- [Appendix: NFS v3, v4, v4.1 Notes](#) – Locking, Delegation, Reconnect, Kerberos, Async I/O, pNFS, Owner ID, UID/GID,...
- [Appendix: Sample Projects](#) – Sample Project names and information
- [Appendix: Jumbo Frames and Delayed ACK](#) – How to use Jumbo Frames/Delayed ACK
- [Appendix: TCP/UDP Echo and Discard Protocols](#) – Support for TCP Echo and Discard Protocols
- [Appendix: Functions and Formula](#) – A description of Action input field Functions and Formula and examples
- [Appendix: Max TCP Open Connections and Open Rate](#) – Find the maximum open connections and connection open rate
- [Appendix: IPv6](#) - Support for the IPv6 protocol.
- [Appendix: DNS and UDP Protocols](#) - Support for the DNS and UDP protocols.
- [Appendix: DCB/DCBx](#) - Data Center Bridging protocol support.
- [Appendix: NTLM Flags](#) - a description of mechanisms used to set/unset 32 NTLM Flags in SMB, SMB2, SMB3, and HTTP/HTTPS Protocols.
- [Appendix: Change Notify and Change Notify Cancel Actions](#) - a detailed review of the behavior of these two Actions.
- [Appendix: Virtual Appliance Constraints & Licensing](#) - Virtual Appliance limitations and Licensing info.
- [Reference: CIFS-SMB Commands and Behaviors](#) – List of CIFS-SMB commands and behaviors
- [Reference: SMB2 Commands and Behaviors](#) – List of SMB2/2.1 commands and behaviors
- [Reference: SMB3.0 Commands and Behaviors](#) - SMB3.0 commands and behaviors
- [Reference: NFSv2 Command List](#) – List of NFSv2 commands
- [Reference: NFSv3 Commands and Behaviors](#) – List of NFSv3 commands & Async I/O notes
- [Reference: NFSv4, v4.1 Command List](#) – List of NFSv4 commands
- [Reference: Kerberos v5 Command List](#) – List of Kerberos v5 commands
- [Reference: HTTP/HTTPS Command List](#) – List of HTTP and HTTPS commands
- [Reference: iSCSI Commands and Behaviors](#) – iSCSI commands and behaviors
- [Reference: FC/iSCSI/SCSI Command and Behaviors](#) – List of Fibre Channel, SCSI and iSCSI commands and behaviors
- [Reference: TCP/UDP Echo and Discard Protocols Command List](#) – List of TCP/UDP Echo Protocol commands
- [Reference: Action Input Shorthand](#) – Discussion of the decimal shorthand allowed in some Action input fields
- [Reference: HTTP Storage Commands and Behaviors](#) - Discussion of support for HTTP Storage protocols
- [Reference: End of Life Statement](#) - End of Life information for older Load DynamiX software releases

Technical Assistance

For product assistance or clarification on information in this guide, contact Load DynamiX at [Load DynamiX Support](#).

When requesting product assistance from the Load DynamiX support team, information and files that will improve the speed and accuracy of the response are:

- Project goal and problem
- TDE release and build numbers
- Appliance Firmware release and build numbers if different from the TDE release and build numbers
- Appliance model (3000, 3108, 5000, 5102, 5108S, 5108T, 6202, 6202E, 6204, 6208, U1022, U1044)

- Test network topology or connection list
- TDE Unhandled Exception Log file (if the requested assistance relates to TDE unhandled exceptions see [Product Installation](#) for Log file location)
- Packaged Project (see below for packaging process)

Please provide all of the above information in your email to [Load DynamiX Support](#).

To package a Load DynamiX Project prior to sending it to Load DynamiX

- Open the Project in the TDE.
- Click on the **Export Project...** menu item in the TDE **File** menu.
- Follow the prompts to create a .ZIP file containing the Project and its Results and Data files.
- Attach this .ZIP file to your email to [Load DynamiX Support](#)
- Emails to [Load DynamiX Support](#) are automatically logged into the [Load DynamiX Help Desk](#)

For file upload/download (FTP as well), go to

[Load DynamiX File Sharing site](#)

Technology References and Terminology

Technology References and Terminology

References

Ref.	Document	
R.1	RFC 2224, NFS URL Scheme	http://www.networksorcery.com/enp/rfc/rfc2224.txt
R.2	RFC 2339, An Agreement Between the Internet Society, the IETF, and Sun Microsystems, Inc. in the matter of NFSv4 Protocols	http://www.networksorcery.com/enp/rfc/rfc2339.txt
R.3	RFC 2623, NFS Version 2 and Version 3 Security Issues and the NFS Protocol's Use of RPCSEC_GSS and Kerberos V5	http://www.networksorcery.com/enp/rfc/rfc2623.txt
R.4	RFC 2624, NFS Version 4 Design Considerations	http://www.networksorcery.com/enp/rfc/rfc2624.txt
R.5	RFC 3530, Network File System (NFS) version 4 Protocol	http://www.networksorcery.com/enp/rfc/rfc3530.txt
R.6	RFC 1094, NFS: Network File System Protocol Specification	http://www.networksorcery.com/enp/rfc/fc1094.txt
R.7	RFC 1813, NFS Version 3 Protocol Specification	http://www.networksorcery.com/enp/rfc/rfc1813.txt
R.8	RFC 3010, NFS version 4 Protocol	http://www.networksorcery.com/enp/rfc/rfc3010.txt
R.9	CIFS Protocol Description	http://msdn.microsoft.com/en-us/library/aa302240.aspx
R.10	Server Message Block (SMB) Protocol Specification	http://msdn.microsoft.com/en-us/library/cc212363.aspx
R.11	Server Message Block (SMB) Version 2.0 and 3.0 Protocol Specification	http://msdn.microsoft.com/en-us/library/cc246482.aspx
R.12	Microsoft Distributed File System	http://www.microsoft.com/windowsserver2003/technologies/storage/dfs/default.mspx
R.13	Kerberos Specification	http://tools.ietf.org/html/rfc4120
R.14	Windows NT Error Codes	http://msdn.microsoft.com/en-us/library/cc704588%28v=PROT.10%29.aspx
R.15	iSCSI RFC 3720	http://www.ietf.org/rfc/rfc3720.txt
R.16	HTTP /1.1 RFC 2616	http://tools.ietf.org/html/rfc2616
R.17	ECHO Protocol	http://tools.ietf.org/html/rfc862
R.18	DISCARD Protocol	http://tools.ietf.org/html/rfc863
R.19	NFSv4.1 Primary RFC	http://tools.ietf.org/html/rfc5661
R.20	Amazon S3	http://docs.aws.amazon.com/AmazonS3/latest/API/Welcome.html
R.21	CDMI	https://tools.ietf.org/html/rfc6208
R.22	OpenStack Swift	http://developer.openstack.org/api-guide/quick-start/
R.23	OpenStack Cinder	http://developer.openstack.org/api-guide/quick-start/
R.24	OpenStack Keystone	http://docs.openstack.org/developer/keystone/

Acronyms and Terms

Acronym/Term	Definition
TDE	Load DynamiX Test Development Environment (interchangeable with GUI)

DNS	Domain Name Service
DUT	Device Under Test
GUI	Graphical User Interface (interchangeable with TDE)
IP	Internet Protocol
NFS	Network File System protocol
SMB, SMB2	Server Message Block protocol and Server Message Block protocol version 2
iSCSI, SCSI	Internet Small Computer Systems Interface (iSCSI), Small Computer Systems Interface (SCSI)
FC	Fibre Channel transport
WAN	Wide Area Network
DFS	Window's Distributed File System
OPLOCKS	CIFS-SMB/SMB2 Opportunistic Locking Mechanism
UPF	User Parameter File
RESOURCE	A component of a Load DynamiX Project. Examples of Resources are Scenarios, Network Profiles, Load Profiles, User Parameter Files, Tracing Parameters, Test Execution Rules, etc.
PROJECT	A test developed in the Load DynamiX TDE. Projects are made up of a TimeLine and the Resources that are on the TimeLine and utilized during the execution of the Project. Also referred to as a Workload.
ACTIONS	The executable elements (e.g. protocol commands) of a Load DynamiX Project
SCENARIO	A component of a Project in which test Actions to be executed are defined
NETWORK PROFILE	The characteristics of a subnet of a Load DynamiX Project
LOAD PROFILE	The characteristics of execution load to be generated by a Scenario
VIRTUAL PORT	The combination of a network definition [Network Profile], test Actions [Scenario] and Load Profile that make up either the Client or Server side of a Project
APPLIANCE	The Load DynamiX hardware platform on which tests developed in the Load DynamiX TDE are executed
TESTER	The individual responsible for developing and/or running the tests that are developed in the Load DynamiX TDE and run on the Load DynamiX Appliance
JUMBO FRAMES	Jumbo Frames are Ethernet frames with greater than 1500 bytes of Payload. In the Load DynamiX environment, 1Gbps interfaces support Jumbo Frames up to 9216 (- headers) bytes, 10Gbps interfaces support Jumbo Frames up to 16128 (-headers) bytes. See Appendix: Jumbo Frames and Delayed ACK for details.
MTU	Maximum Transmission Unit - the maximum size of the Payload area of an Ethernet packet that the Load DynamiX Appliance will transmit. MTU is configurable by double clicking a Logical Port in a Timeline. The maximum supported MTU setting on a Load DynamiX 1Gbps interfaces is 9216. The maximum supported MTU setting on a Load DynamiX 10Gbps interfaces is 16128. See Appendix: Jumbo Frames and Delayed ACK for details.

DELAYED ACK	The ability for the Load DynamiX TCP stack to delay when TCP ACK messages are sent. Defined in terms of Milliseconds or bytes of unacknowledged data. See Appendix: Jumbo Frames and Delayed ACK for details.
LUN	SCSI Logical Unit Number - an identifier for a device being accessed via the SCSI protocol
HTTP/1.1	HyperText Transfer Protocol version 1.1
EVENT	A means to synchronize execution between cooperating Scenarios. See Appendix Load DynamiX Scenario Control Actions for details.
STRING VARIABLES	User defined variables that can be set to hold specific content and used in Scenario Action input fields. See Advanced Concepts: Variables and Aliases for details.

Product Installation

Product Installation

This chapter describes how to set up the Appliance, install the Load DynamiX TDE application, verify connectivity between the TDE and Appliance and install Protocol Licenses (if that has not already been done before shipment).

Support

Email product support questions to: Support@LoadDynamix.com

What ships with the Load DynamiX product:

- Appliance with mounting hardware (rails, etc)
- TDE software on a USB thumb drive
- Hard copy of an Appliance-specific Load DynamiX Quick Start Guide
- License file for Protocols purchased for this Appliance

APPLIANCE

Hardware Installation - Dimensions: Height 3.5", Width 17.2", Depth 25.5"

Installation into a Rack

The Load DynamiX Appliance shipping carton includes two sets of rail assemblies, two rail mounting brackets and the mounting screws required to install the system into the rack.

Optional: The shipping carton also includes two extension rails that can be attached to the Load DynamiX Appliance to allow it to be serviced without removing it entirely from the rack. These extension rails are not necessary for normal function of the device.

Temperature Considerations: Airflow on the Load DynamiX Appliance is from front to back. The front of the Appliance has the Reset and Power buttons, and the Admin and Test ports. The Front of the Load DynamiX Appliance must be installed on the cool side of the rack in which it is located. Installing the Load DynamiX Appliance with the back of the Appliance on the cool side of the rack can result in chassis overheating and product failure.

Communications Considerations: The Load DynamiX Appliance and TDE together create a test development and execution environment for performance, capacity and stress oriented networked storage tests. It is advisable to have high speed, low latency communications between the TDE and the Appliance to facilitate test development and results gathering and between the Appliance and the device being tested.

If the network link between the TDE and Appliance is not high speed or has high latency, it is suggested that the TDE be installed on a Windows VM near the Appliance and the TDE be accessed via Remote Desktop or some other remote access solution. To optimize Appliance to device under test communications, it is recommended that as little network infrastructure as possible be between the Appliance's test ports and the device under test. Network infrastructure can add unexpected delay and complexity during test execution.

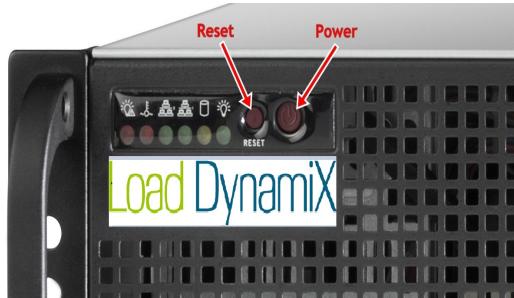
Providing Power

Plug the 2 power cords from the power supply units into a high-quality power source that offers

protection from electrical noise and power surges. It is recommended that you use an uninterruptable power supply (UPS). Independent circuits for the two power supplies are recommended. Both power supplies must be plugged in for proper function.

Power Down Procedure

To power down the Load DynamiX Appliance, press the Power button (the larger of the two Red buttons) on the front of the unit. This button is the one on the right. The smaller, recessed red button Resets the unit.



Admin Network, Time and Date Configuration



The Load DynamiX Appliances have two types of ports - Admin Ports and Test Ports (8x1000BASE-T ports on the 1G Series models 3000 and 3108, 2x10Gbps ports on the 10G Series models 5000 and 5102, 2x16Gbps Fibre Channel ports on the FC Series model 6202, 2x10Gbps Fibre Channel over Ethernet FC Series model 6202E, 8x16Gbps Fibre Channel ports on the FC Series model 6208, 8x10GBASE-T ports on the 10G Series model 5108T and 8x10Gbps ports on the 10G Series model 5108S, 2x16Gbps Fibre Channel ports and 2x10GBASE-T/10Gbps SFP+ ports on the Unified Series U1022, 4x16Gbps Fibre Channel ports and 4x10GBASE-T/10Gbps SFP+ ports on the Unified Series U1044).

The Test Ports are configured from the Load DynamiX TDE (the GUI) and it is highly recommended that the Test Ports and Admin Ports be connected to different networks and switches. The Test Ports can generate significant amounts of traffic, and if test traffic overwhelms the network that the Admin Port is on, it may be difficult to access the Admin Port when tests are running.

Appliance Admin User Interface

Depending on the Appliance model (3000, 5000, 3108, 5102, 6202, 6202E, 6208, 5108S or 5108T), the Appliance Admin User Interface can be used to set the IP address and related information or set the Date/Time (1G Series Model 3000 and 10G Series Model 5000) or set the IP address and related information; execute network diagnostics or set the Timezone/Date/Time (1G Series Model 3108, 10G Series Model 5102, FC Series Model 6202/6202E/6204/6208, 10G Series Models 5108S and 5108T).

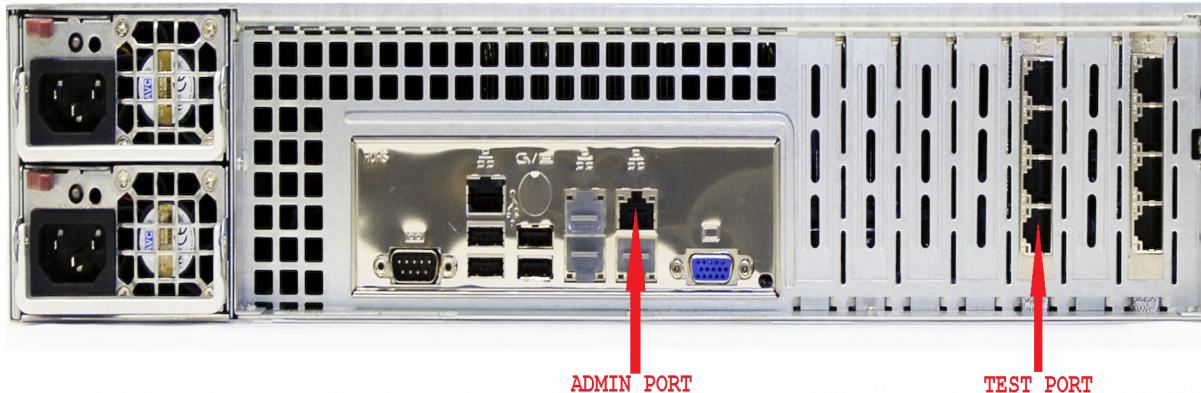
To access the Load DynamiX Appliance Admin User Interface, telnet/ssh into the IP address that is used to access the Appliance by the TDE.. The User ID and Password for the Load DynamiX Appliance Admin User Interface is:

User ID == config
Password == config

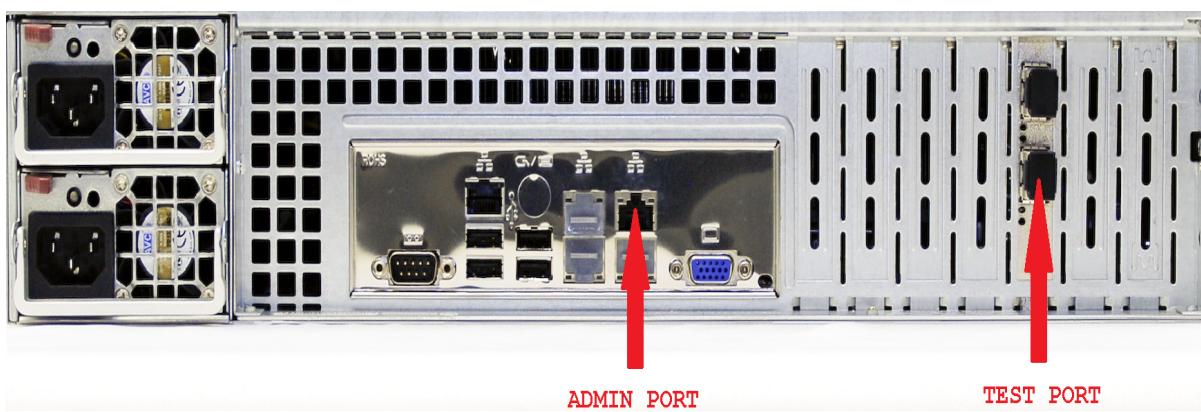
For a detailed discussion of the Admin User Interface see the platform Quick Start Guide which can be accessed via the Help drop down menu on the TDE. These PDF files are also installed in the User Guides and Documents folder. See above in the TDE Software section of this chapter for the location of the User Guides and Documents folder.

- Load DynamiX 3000/5000 Appliance Admin User Interface, please see the Load DynamiX 3000 5000 Quick Start Guide
- Load DynamiX 6202/6202E/6204/6208 Appliance Admin User Interface, please see the Load DynamiX 6202 6204 6208 and Unified Series Quick Start Guide
- Load DynamiX 3108/5102 Appliance Admin User Interface, please see the Load DynamiX 3108 5102 5108T 5108S Quick Start Guide
- Load DynamiX 5108T/5108S Appliance Admin User Interface, please see the Load DynamiX 3108 5102 5108T 5108S Quick Start Guide
- Load DynamiX U1022 and U1044 Unified Series Appliance Admin User Interface, please see the Load DynamiX FC Series Model 6202 6202E 6204 6208 and Unified Series Quick Start Guide

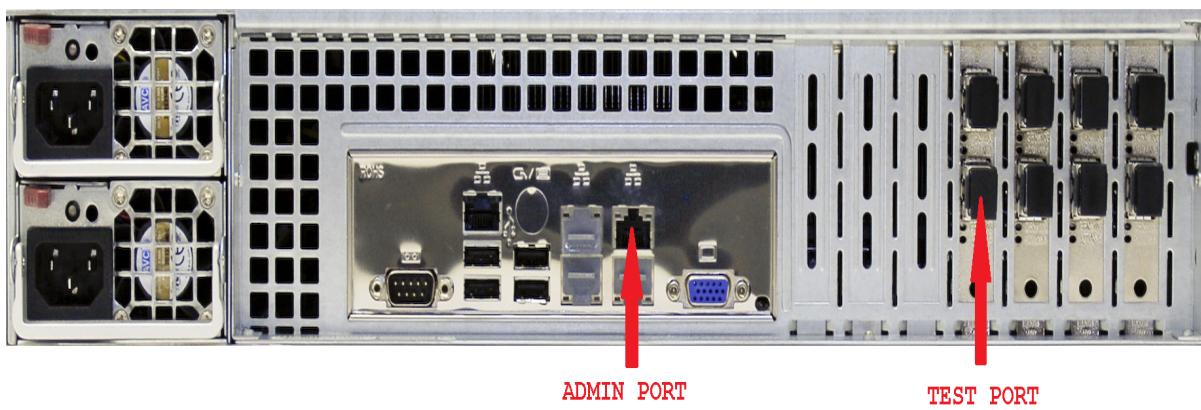
Load DynamiX 1G Series Model 3000/3108 Connections



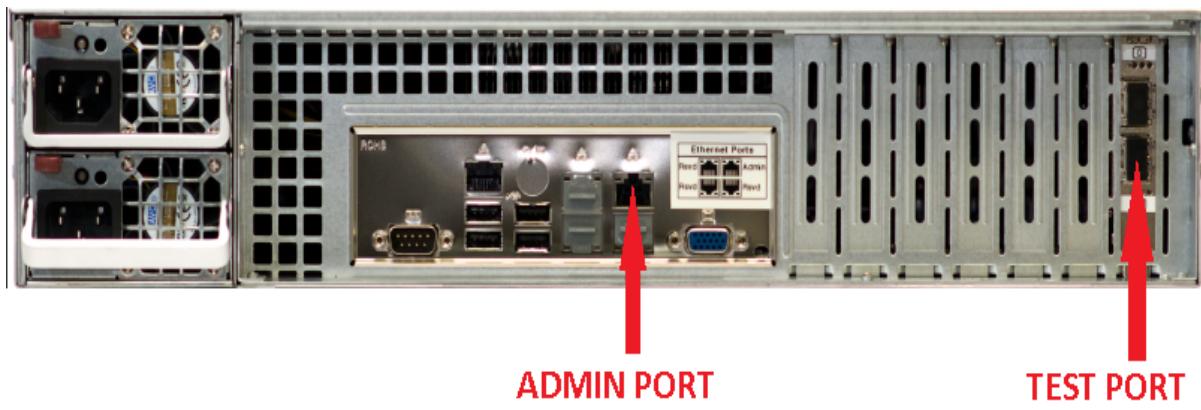
Load DynamiX 10G Series Model 5000/5102 Connections



Load DynamiX 10G Series Model 5108 Connections 5108



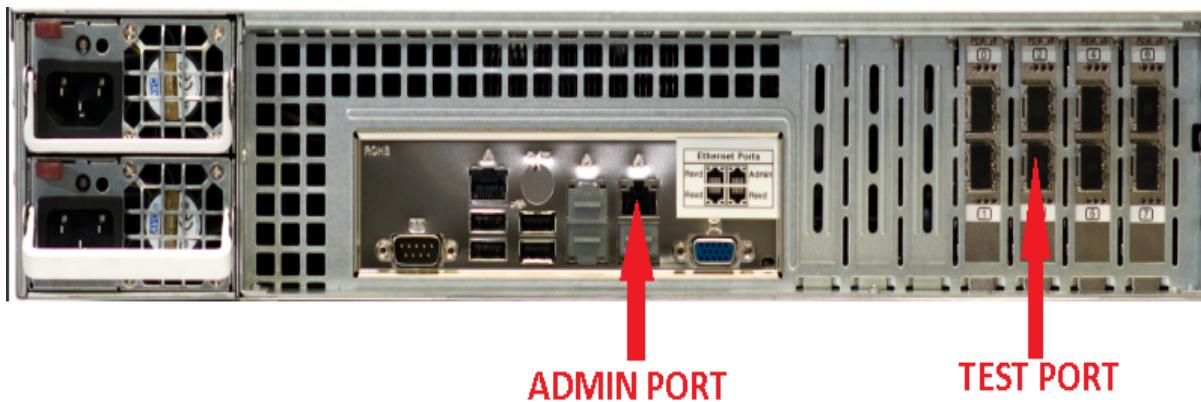
Load DynamiX FC Series Model 6202/6202E/6204/6208 Connections 6202/6202E



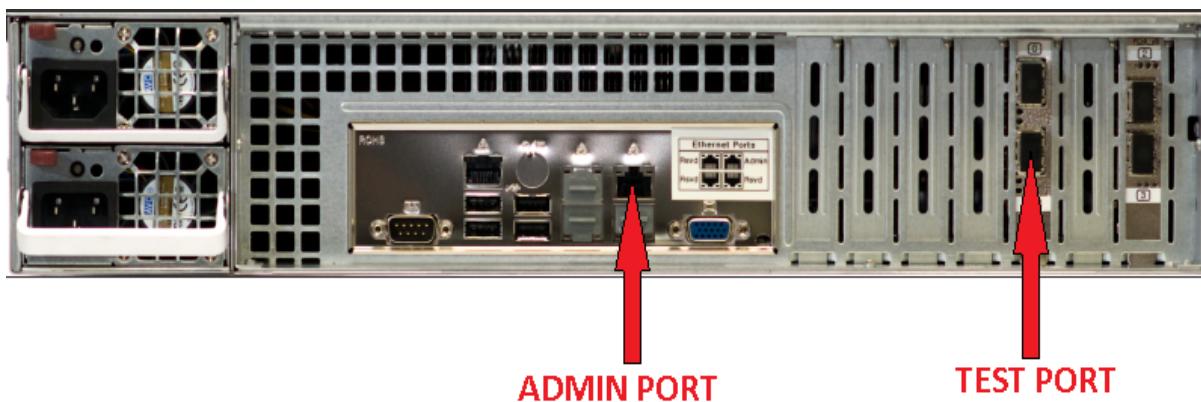
6204/6204E



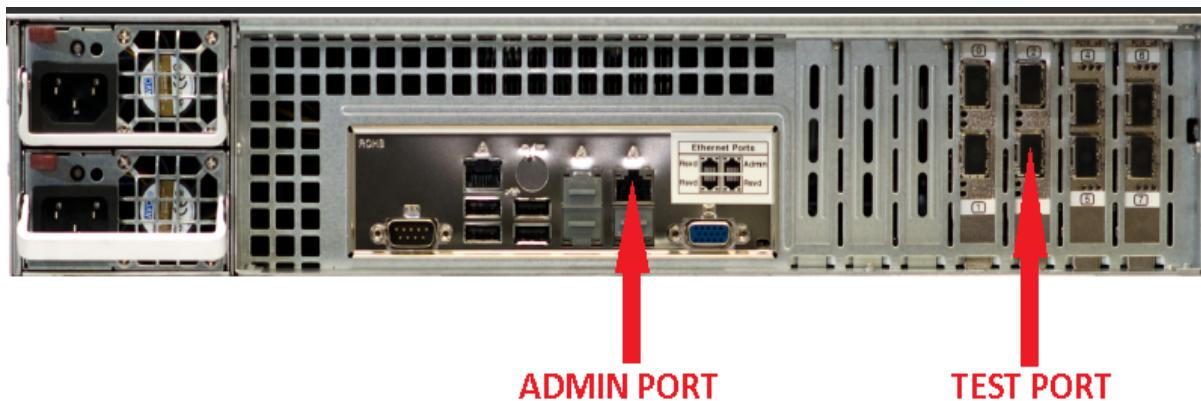
6208



Load DynamiX Unified Series Model U1022/U1044 Connections
U1022



U1044



Load DynamiX Appliance Admin and Test Port Cabling

Admin Ports

All Load DynamiX Appliance Admin Ports use standard 1GbE Cat 5 Ethernet cables with RJ45 connectors.

Test Ports

The Load DynamiX 1G Series Models 3000 and 3108 1000BASE-T Test Ports use standard Cat 5 Ethernet cables with RJ45 connectors.

The Load DynamiX FC Series Model 6202, 6202E, 6204, 6208 Fibre Channel Test Ports contain modular SFP+ transceivers that use an 850nm laser which requires Multi-mode fiber optic cables with

an LC connector. The Load DynamiX FC0E series model 6202E Test Ports contain an FCoE-specific SFP+ transceivers that use an 850nm laser which requires Multi-mode fiber optic cables with an LC connector. Compatible active or passive Direct Attach Cables may also be used with the 6202E Test Ports. See below for transceiver support details.

The Load DynamiX 10G Series Models 5000, 5102, 5104, and 5108 10GbE Test Ports ship with SFP+ optical transceivers that use an 850nm laser which requires Multi-Mode fiber optic cable with an LC connector. Compatible active or passive Direct Attach Cables may also be used with the 5000, 5102 and 5108S Test Ports. See below for transceiver support details.

The Load DynamiX Unified Series Model U1022 and U1044 10GbE and Fibre Channel Test Ports contain modular SFP+ transceivers that use an 850nm laser which requires Multi-mode fiber optic cables with an LC connector. Compatible active or passive Direct Attach Cables may also be used with the U1022 and U1044 Test Ports. See below for transceiver support details.

Load DynamiX 10G and Fibre Channel Appliance Transceiver Support

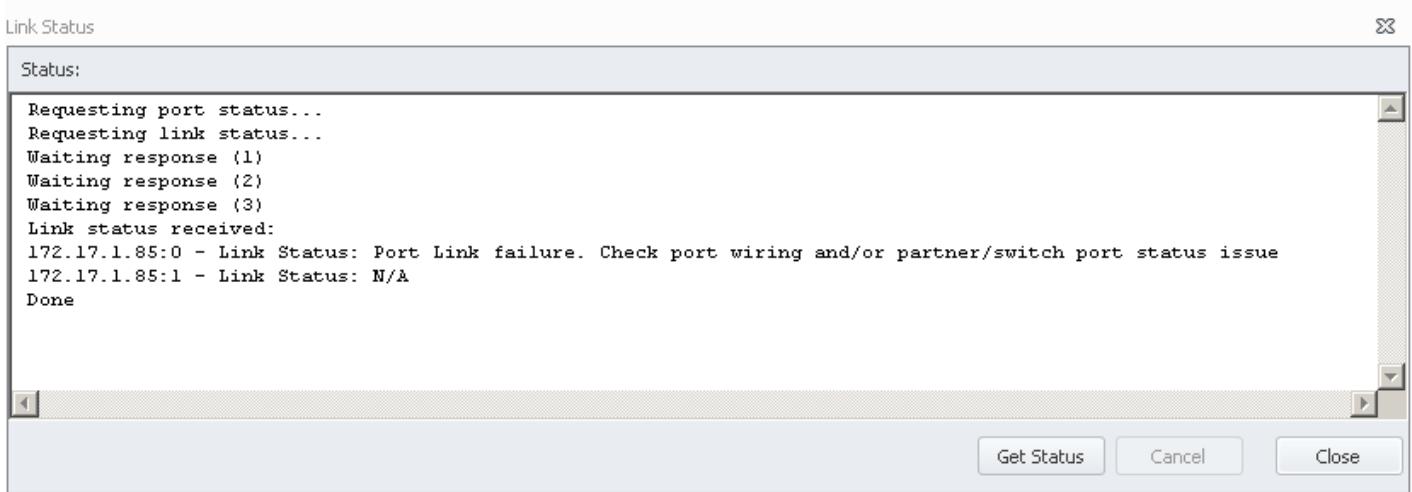
Other compatible SFP+ optical transceivers or SFP+ passive or active DA (Direct Attach) cables (sometimes referred to as "Twinax" cables) will work with the Load DynamiX 10G Series Appliance. Active Direct Attach copper cables from Cisco, Arista and Brocade have been shown to work as well as all Passive Direct Attach copper cables.

On Appliances with SFP+ optical 10GbE interfaces (5108S, U1022 or U1044), to verify that an optical SFP+ cable other than that which is shipped with the Load DynamiX 10G Appliance, connect two Load DynamiX 10GbE Test Ports in a back to back configuration with the transceiver/cable combination and run one of the Load DynamiX back-to-back Sample Projects such as CIFS-SMB Full Duplex Payload. If the Project runs successfully then the optical SFP+ transceiver or DA SFP+ cable is compatible with the Load DynamiX 10GbE Ethernet ports.

For this transceiver/cable combination to work with a DUT or a 10GbE switch, the transceiver/cable must also be compatible with the target connection (DUT or switch). If the back-to-back test is successful but a test using a connection to a DUT or switch is not, it is possible that the transceiver/cable combination is not compatible with the target. You can verify link status using the TDE Ports & Appliances > Appliances tab entry for the Load DynamiX Appliance but link status does not guarantee that traffic can be sent over this connection. When running a test to verify a working connection, include a Tracing Resource in the project. If the PCAP file that results from the Tracing Resource contains only ARP packets then the transceiver/cable combination is incompatible with the target even if the Link Status appears OK.

For Fibre Channel SFP+ optical interfaces, back-to-back tests are not available. To test for compatibility on a Fibre Channel Appliance, connect the cable to the Appliance and FC switch and execute a sample Fibre Channel Project. If the Project succeeds then the cable is compatible with the Appliance.

For additional debugging purposes, the Appliance Link Status messages (from Ports & Appliances > Appliances > Link Status) for incompatible SFP+ transceiver/cable combinations may produce different kinds of messages. The messages below indicate incompatibility for two different kinds of SFP+ transceiver/cable combinations.



When executing a Project using an incompatible SFP+ transceiver/cable, the Output window and Client log files contain error and warning messages related to the SFP+ status:

Output window:

```
Server Port 4(172.17.1.85:1) Statistics updated - Port time: 0ms
Analyzing log file...
  Error: 172.17.1.85:1 - appliance error: <ERROR Device [1]: SFP+ Transceiver is not found, or is not properly inserted.>
  Warning: 172.17.1.85:1 - appliance warning: No devices running, exiting...
  Error: 172.17.1.85:1 - error occurred. Execution stopped
Stopping test at port list: 172.17.1.85:0,1
Waiting while ports are stopping...
  172.17.1.85:1 - Idle
  172.17.1.85:0 - Idle
Receiving files...
Receiving pcap files...
Analyzing log file...
  Error: 172.17.1.85:0 - appliance error: <ERROR Device [0]: Port Link failure. Check port wiring and/or switch port status.>
  Warning: 172.17.1.85:0 - appliance warning: No devices running, exiting...
  Error: 172.17.1.85:0 - error occurred. Execution stopped
Test finished - 4 errors, 2 warnings
```

The Client log file contains information extracted from the SFP+ transceiver: The Vendor OUI (vendor identifier) and the Capabilities that the transceiver provides. See the [Vendor OUI reference](#) for more details regarding the mapping of values to vendors.

The first byte of the Capabilities indicates the presence of 10GbE capable SFP+ : values that the Load DynamiX Appliance will accept are 0x10 and 0x20.

The second byte indicates the presence of 1GbE capable SFP+ : values that the Load DynamiX Appliance will accept are 0x01 and 0x02 (in case the transceiver is multi-speed capable).

The third byte indicates if the transceiver is part of an Active or Passive Direct Attach transceiver/cable combination : values that the Load DynamiX Appliance will accept are 0x04 and 0x08.

All other potential values in these 3 bytes will result in the Load DynamiX Appliance rejecting the attached SFP+ transceiver. See the [Capabilities reference](#) for more details.

Client log:

	Line	Type	Date / Time	Text
▶	0	Status	4/5/2013 5:11:50 PM	Load DynamiX Framework [Version 1.35.27763-Internal]
	1	Status	4/5/2013 5:11:50 PM	Copyright 2008 - 2014 Load DynamiX Inc.
ⓘ	2	Info	4/5/2013 5:11:51 PM	Device [0]: SFP+ Fiber Transceiver initialization successful.
⚡	3	Debug	4/5/2013 5:11:51 PM	Device [0]: SFP+ Transceiver Vendor OUI [0x001B21], capabilities [0x10][0x01][0x00].
⚡	4	Debug	4/5/2013 5:11:51 PM	Device [0]: Negotiating link...
⚡	5	Debug	4/5/2013 5:11:56 PM	Stopping execution...
⚡	6	Debug	4/5/2013 5:11:56 PM	Device [0]: Failed to negotiate link.
✖	7	Error	4/5/2013 5:11:56 PM	<ERROR Device [0]: Port Link failure. Check port wiring and/or switch port status.>
⚠	8	Warning	4/5/2013 5:11:56 PM	No devices running, exiting...

Messages of the form “<ERROR Device[x] Generic Failure: Status Code [y]>” also indicate the presence of an incompatible optical or DA transceiver plugged into the Load DynamiX 5000 10GbE port.

TDE software

Default Installation Directories

{InstallationFolder} = Load DynamiX TDE [Defined during the installation process]

{UserLoginName} = Login ID of the User who installed the TDE

Content:

- Program executables: Various product binary files such as the TDE and LdxCmd executables and product .dll files
- Program data: Sample tests and resources
- Projects: User created or modified Project folders
- Resources: User created or modified Resource files
- User Guides and Documents: PDF version of online help, quick start guides
- Scripts: Sample Perl and Tcl scripts
- Mono: Binary, text and XML files necessary to run Load DynamiX Projects on a Linux platform with Mono
- Log: Location for log files generated by TDE or LdxCmd during Project execution (e.g. TDE exceptions)

System Requirements (the Management Workstation)

- Windows PC (Windows 7, 8, 8.1 and 10), 2 Gigahertz or higher processor clock
- 2 GB of RAM or higher
- 500 megabytes (MB) of available hard disk space
- **Windows .NET framework version 4.x (see below for instructions on how to download and verify versions)**

Windows 7, 8, 8.1 and 10

Program executables: C:\Program Files\LoadDynamiX\{InstallationFolder}

ProgramData: C:\ProgramData\LoadDynamiX\{InstallationFolder}

Projects: C:\Users\{UserLoginName}\Documents\LoadDynamiX\My Projects

Resources: C:\Users\{UserLoginName}\Documents\LoadDynamiX\My Resources

User Guides and Documents: C:\ProgramData\LoadDynamiX\{InstallationFolder}\Load DynamiX Docs

Read-Only Sample Projects: C:\ProgramData\LoadDynamiX\{InstallationFolder}\Load DynamiX

Projects

Scripts: C:\Program Data\LoadDynamiX\{InstallationFolder}\Load DynamiX Docs\scripts

Mono: C:\Program Data\LoadDynamiX\{InstallationFolder}\Mono

Log: C:\Users\{UserLoginName}\AppData\Local\Load DynamiX\

Notes:

1. On some versions of the Windows operating system, the system folder C:\ProgramData may have a space between Program and Data and it may not (e.g. ProgramData as seen above for Windows 10 Professional)
2. A PDF version of the Load DynamiX Appliance-specific Quick Start Guide is installed with the TDE in the User Guides and Documents folder.

Installing the application

The Load DynamiX application is packaged on a thumb drive, delivered via email or downloaded from the Load DynamiX FTP site (LoadDynamiX.Egnyte.com).

Windows .NET framework version 4.x must be installed on the Management WorkStation before installing the Load DynamiX TDE. Windows 7 and earlier versions may require .NET Framework 4.x to be installed.

To verify that .NET v4.0 is installed on the Management WorkStation, from a command window prompt, issue the following commands in a command window:

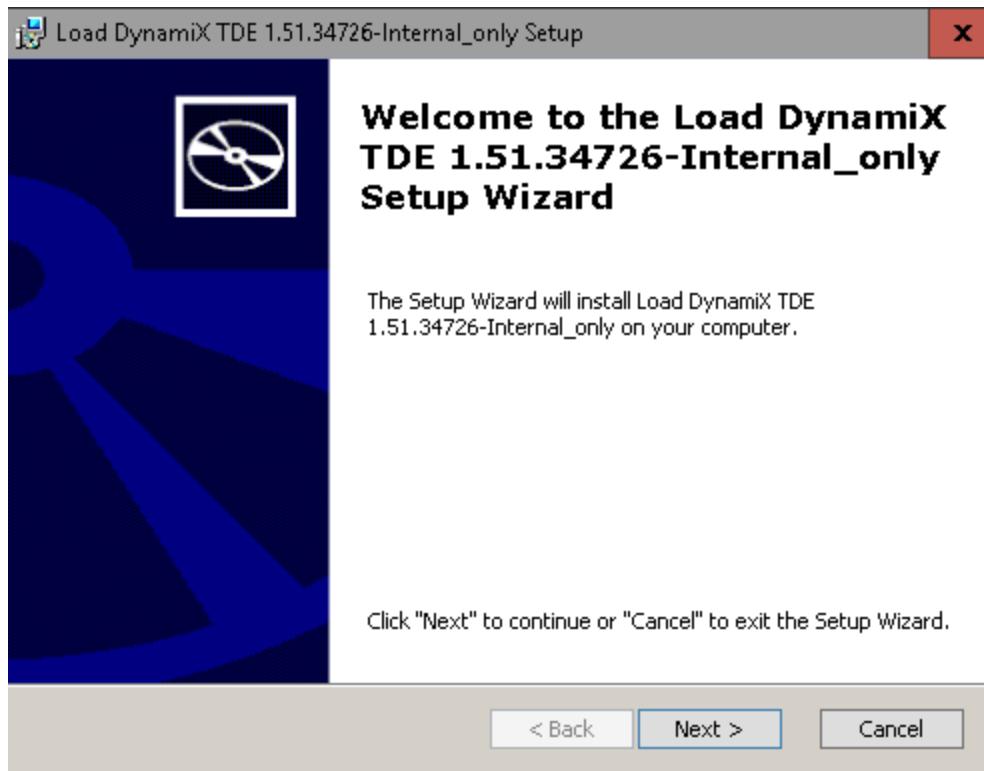
- dir C:\WINDOWS\Microsoft.NET\Framework*^
- dir C:\WINDOWS\Microsoft.NET\Framework64*^

To install .NET Framework 4.x on a Management Workstation running Windows 7 or earlier:

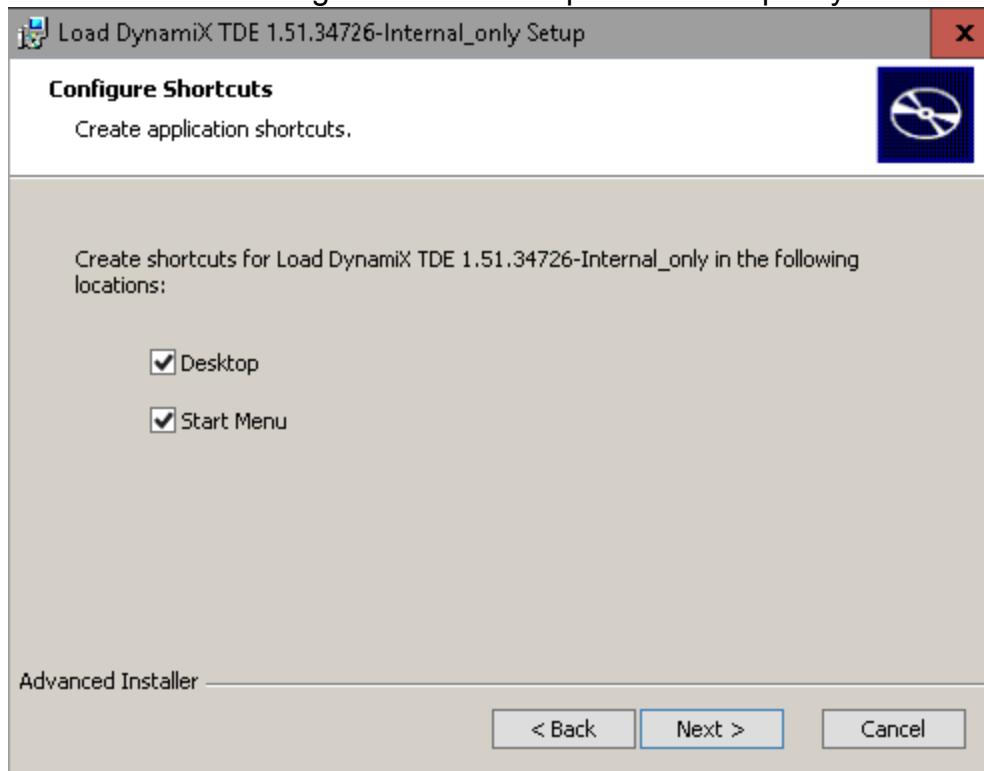
- If the Management WorkStation has access to the Internet, download the .NET 4.x bootstrap loader at: <https://www.microsoft.com/en-us/download/confirmation.aspx?id=42643>
- If the Management WorkStation does not have Internet access, using a system that does, go to the URL displayed above, click Instructions, look for the phrase “Full Redistributable Package” follow the instructions to download the full .NET 4.x package, then install it on the Load DynamiX WorkStation.

To install the Load DynamiX TDE application

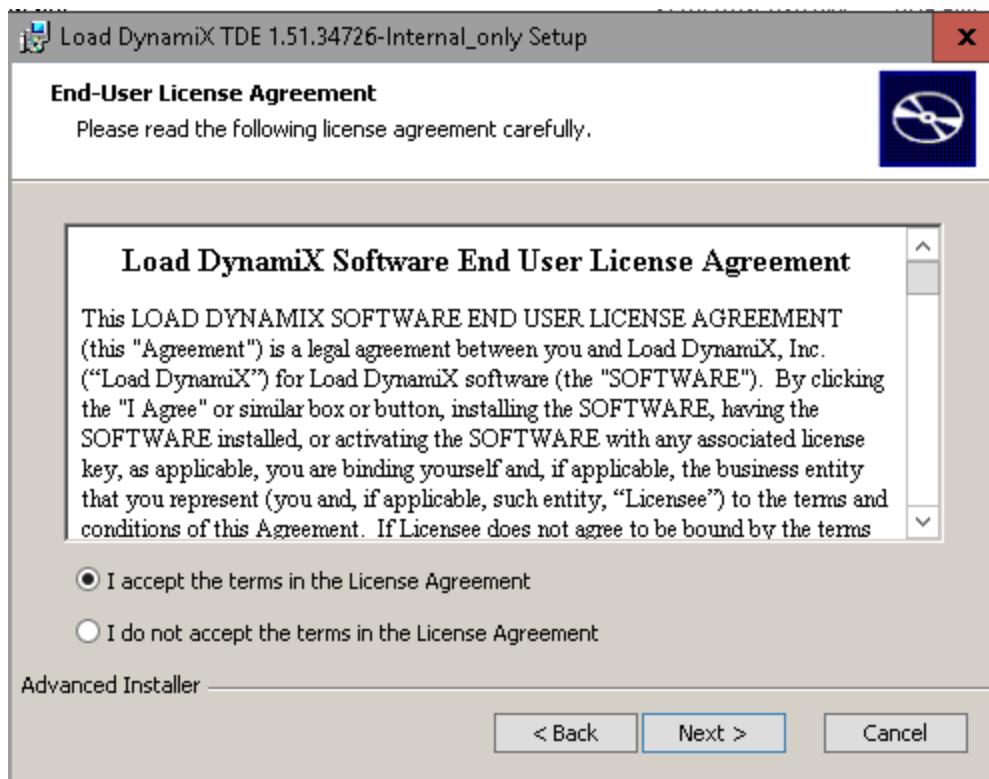
1. If starting from Thumb Drive, insert the Thumb Drive into your Windows system. The setup program should bring up the Windows installer. If starting from email or an FTP'd version, double-click the Load DynamiX TDE.exe file.
2. Follow the instructions to install the application. The installer copies the files and creates the folder structure to run the application. The default installation folder is C:\Program Files\Load DynamiX\Load DynamiX TDE (Load DynamiX TDE is referred to throughout this document as the {InstallationFolder}).



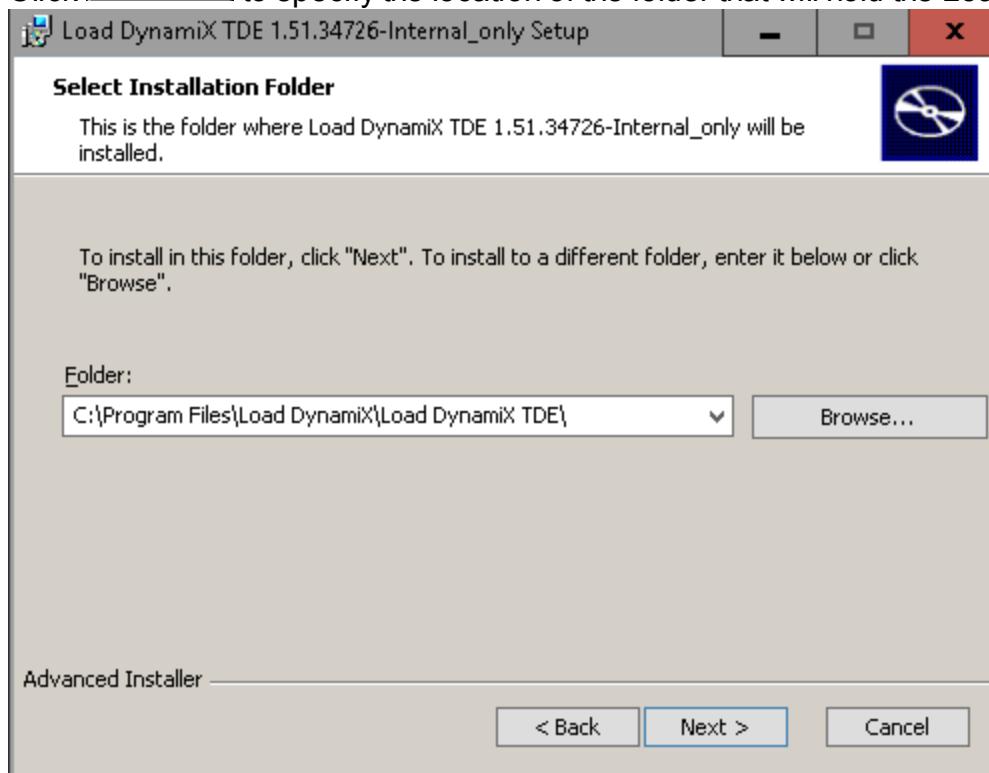
Click **Next >** to begin the installation process and specify where TDE shortcuts are created.



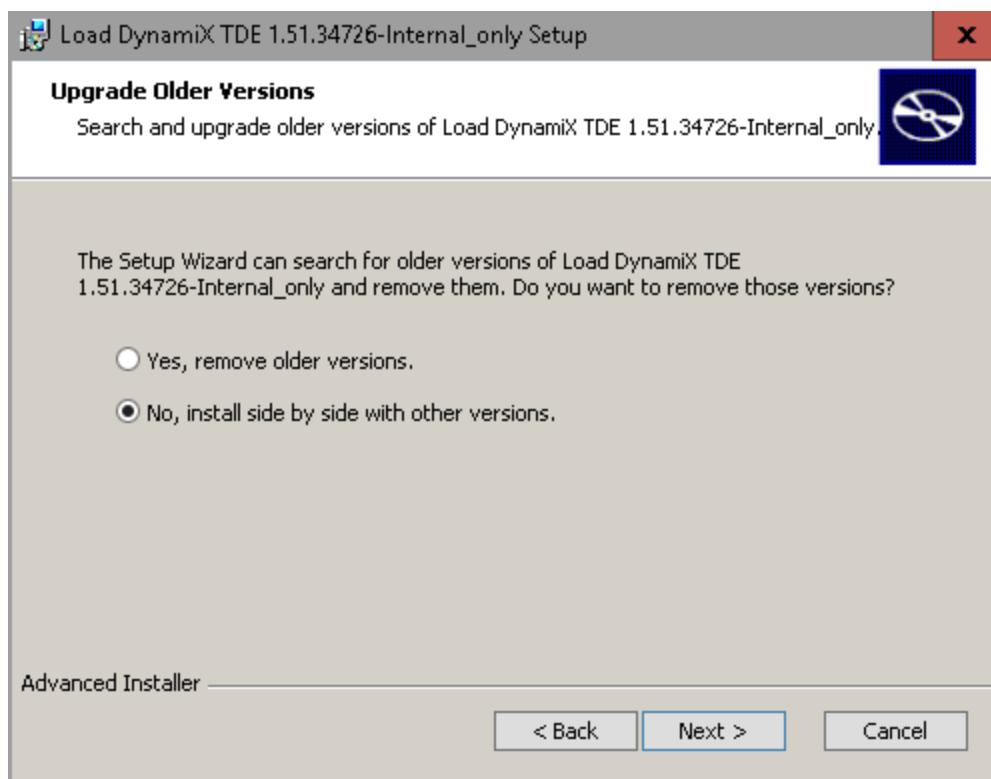
Click **Next >** to agree to the Load DynamiX End User License agreement.



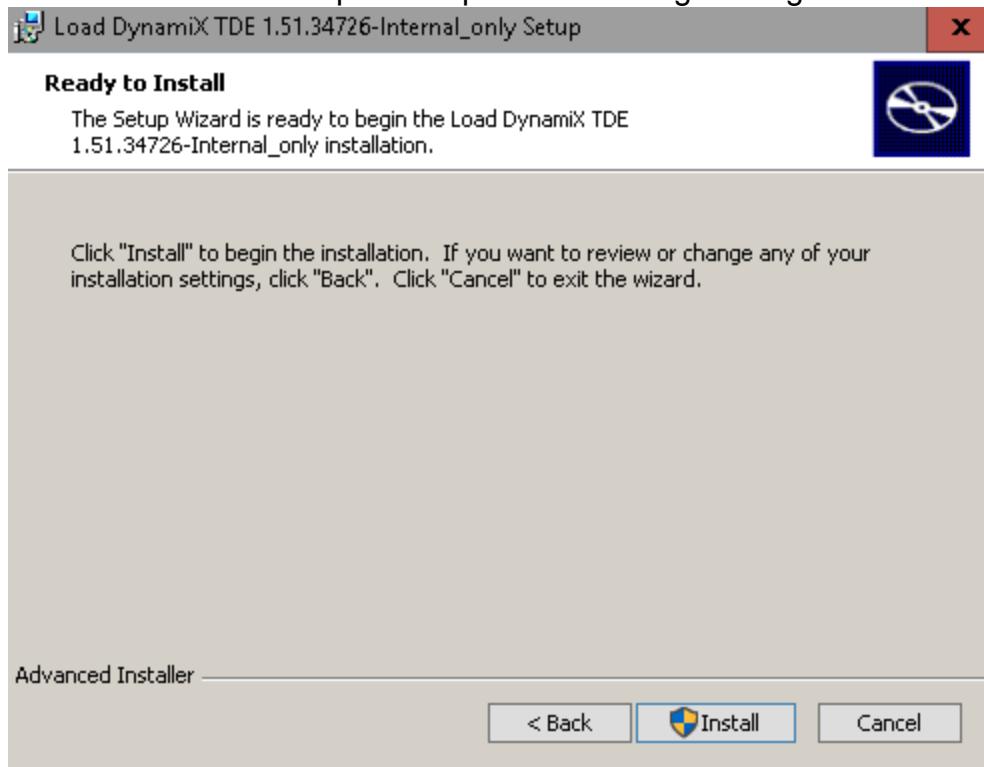
Click **Next >** to specify the location of the folder that will hold the Load DynamiX TDE binaries.



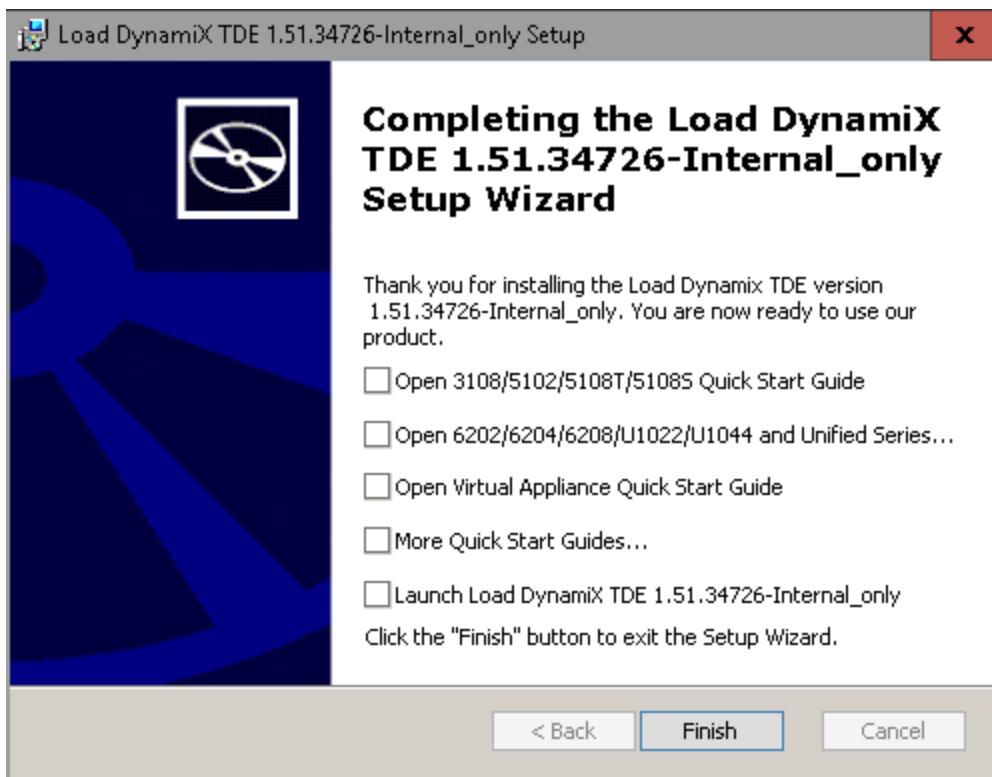
Click **Next >** to specify if the installation is to preserve previous installations or to replace them (installing side by side is necessary if more than one version of the Load DynamiX TDE is installed on the same workstation).



Click **Next >** to complete the pre-install data gathering.



Click **Install** to install the TDE.



Click **Finish** to complete the installation of the TDE and (optionally) Launch the TDE and/or open an Appliance-specific Quick Start Guide or the folder that contains all Load DynamiX end user documentation.

Starting the application

From the installation folder (or desktop), double-click the Load DynamiX application icon  to start the application.

Using the application

It is possible to have multiple different versions of the TDE installed on the same desktop/laptop and interact with different Appliances running different versions of the Load DynamiX Appliance Firmware. Testers can have multiple instances of the TDE GUI (same or different versions) running at the same time, interacting with the same or different instances of the Appliance test platform thus being able to control multiple tests from the same desktop or laptop. One nuance of having multiple GUIs running at the same time is that Appliance administration (assigning Appliance Physical Ports to Test Logical Ports) is shared among the running instances. Debugged Projects can be run by the Load DynamiX automation command: LdxCmd.exe. See [Appendix: Test Automation and LDX-E Integration](#) for a more detailed discussion of test automation.

The PC on which the TDE was installed will be referred to in this chapter as the WorkStation.

WorkStation to Appliance Communications

The Load DynamiX TDE and Automation tools communicate with Load DynamiX Appliances using the HTTP protocol. For the TDE to function properly, the TDE must be able to exchange HTTP packets with the Appliance. To test communications between WorkStation and Appliance try the following:

- From a command prompt on the WorkStation, Ping the Appliance IP address (Windows

command: ping <IP Addr>)

- From a command prompt on the WorkStation, run a trace route (Windows command: tracert <IP Addr>)
- From a browser on the WorkStation, type the Appliance IP Addr into the address bar and the word "Load DynamiX" should appear in the browser window.

If any of the above tests fail, the WorkStation and Appliance will not be able to communicate.

WorkStation HTTP Proxy Configuration

The Load DynamiX TDE and Automation tools communicate with a Load DynamiX Appliance using the HTTP protocol. If the WorkStation is configured to perform HTTP operations through an HTTP proxy then that proxy must be made aware of the Load DynamiX Appliance Admin Port IP addresses for successful communications with Appliances. If an HTTP proxy has been enabled on the WorkStation, either add the Load DynamiX Appliance Admin Port IP addresses to the HTTP proxy or disable the proxy setting.

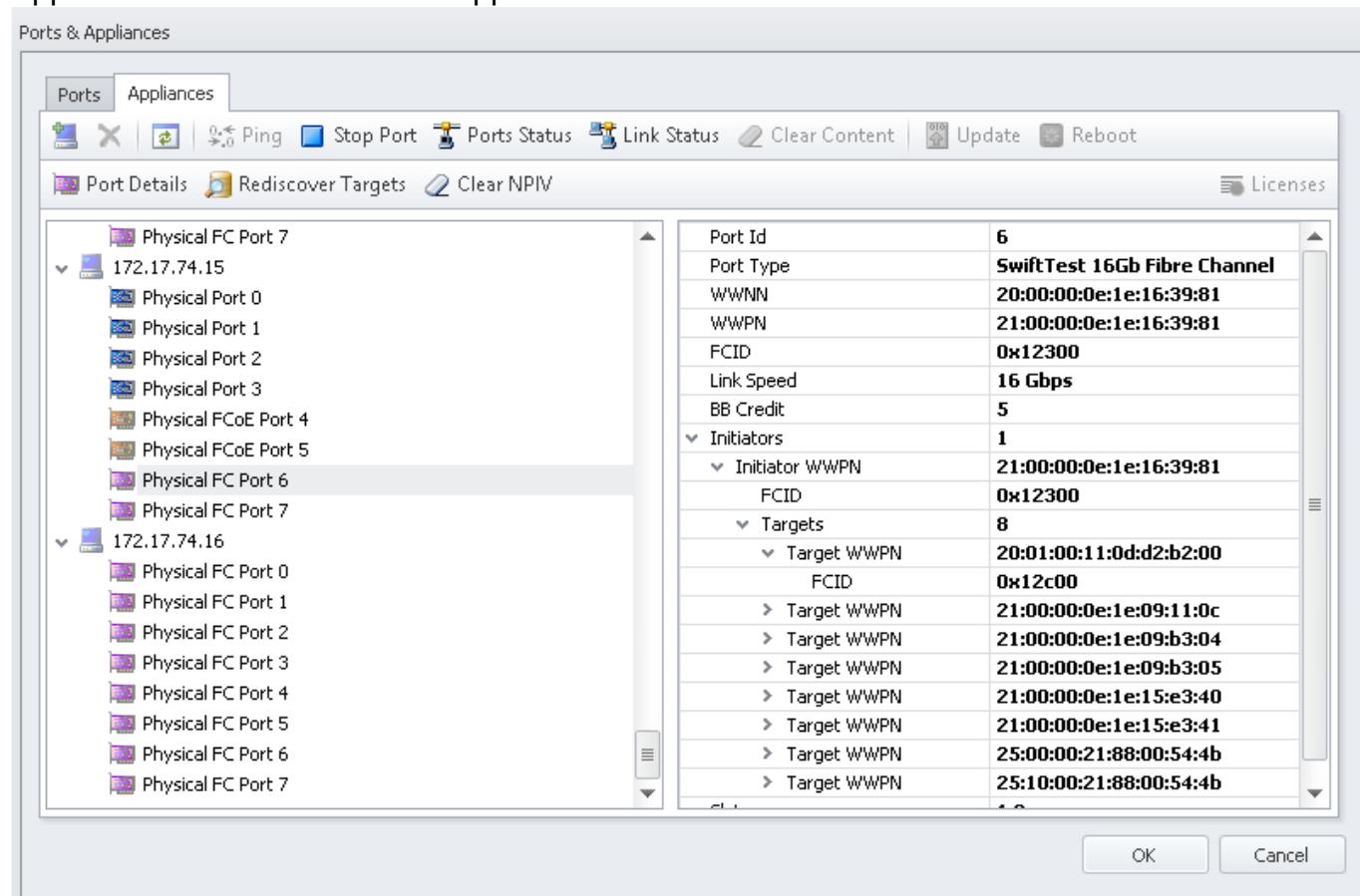
Determining TDE and Appliance Software Versions

TDE

The Main Toolbar of the TDE contains a Help button. Click Help>About to get the TDE Software Version.

APPLIANCE

The Main Toolbar of the TDE contains a Ports & Appliances button . Click the Ports & Appliances button then click the Appliances tab.



All operations (other than Ping) require HTTP access between the TDE and the Appliance. If operations from this window other than Ping do not work, a simple test of HTTP access is to open a browser window on the WorkStation and type the IP address of the Appliance into the browser's address bar. If the browser window displays the text "Load DynamiX" then the HTTP connection is operating correctly and some other issue may be impacting access to the Appliance.

Highlight the Appliance IP address and click the Refresh Appliance Info button  . The information on the right hand side of the window will contain the version of currently installed Appliance Software for that Appliance.

Licensing

Load DynamiX licenses hardware Appliances on a per-Protocol basis. This means that, on a per-Appliance basis, each Protocol that is to be used on that Appliance must be enabled in the License that is associated with that Appliance. **An Appliance will NOT execute Projects that contain Protocols for which it is NOT licensed.** The TDE itself is not licensed but the TDE is the tool that is used to administer licenses for Appliances. Once an Appliance is Licensed, downgrading to an older release that does not support Licenses will not undo Activation or any Licenses installed.

Licensing Process Terminology:

Protocols: The basic unit of licensing. Purchase of the Load DynamiX product requires purchasing the use of various protocols - for example CIFS-SMB, NFSv3, iSCSI possibly Client and Server. Appliances will be licensed for all of the Protocols that a user has purchased.

Temporary Licenses: A Temporary License is a license for one or more Protocols that is valid up to a certain Date. It is useful when evaluating a new Protocol before deciding to purchase the license for that Protocol.

Permanent Licenses: Most licenses are Permanent. When a user purchases the SMB2 Client and Server Protocols, they are purchased permanently and will be licensed as such.

Emergency Licenses: If for some reason an Appliance license does not work, the user can create an Emergency License from the License Management window to enable all Protocols for 7 days. Once the Emergency License expires, it is NOT possible to create a new Emergency License until a Temporary or Permanent License has been installed.

Activation: Preparing an Appliance to be licensed. The Activation process installs an Activation Record on the Appliance and enables the Appliance to accept Protocol licenses. Activation of Appliances installed in customer facilities is only required for Appliances that were shipped before May 2013. The easiest way to check if an Appliance is ready for licensing is to open the Ports & Appliances Window and select the Appliances Tab.

Load DynamiX licenses Virtual Appliances per-Protocol and number of running instances. See Appendix: [Virtual Appliance Constraints & Licensing](#) for details of Licensing for the Load DynamiX Virtual Appliance. Also, see a list of the constraints placed on the Load DynamiX Virtual Appliance.

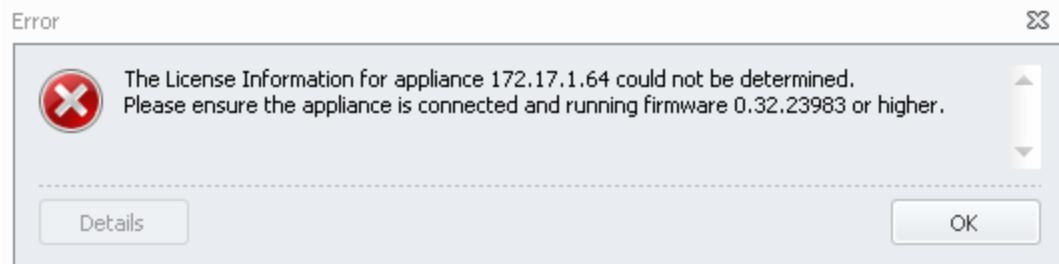
Ports & Appliances

The screenshot shows the 'Ports & Appliances' dialog with the 'Ports' tab selected. In the left pane, under 'Port Details', there is a tree view of ports. The root node is 'Physical FC Port 7'. Under it, there are two nodes: '172.17.74.15' and '172.17.74.16'. Each node has several child nodes representing different port types. On the right side, there is a detailed table of port properties:

Port Id	6
Port Type	SwiftTest 16Gb Fibre Channel
WWNN	20:00:00:0e:1e:16:39:81
WWPN	21:00:00:0e:1e:16:39:81
FCID	0x12300
Link Speed	16 Gbps
BB Credit	5
Initiators	1
Initiator WWPN	21:00:00:0e:1e:16:39:81
FCID	0x12300
Targets	8
Target WWPN	20:01:00:11:0d:d2:b2:00
FCID	0x12c00
Target WWPN	21:00:00:0e:1e:09:11:0c
Target WWPN	21:00:00:0e:1e:09:b3:04
Target WWPN	21:00:00:0e:1e:09:b3:05
Target WWPN	21:00:00:0e:1e:15:e3:40
Target WWPN	21:00:00:0e:1e:15:e3:41
Target WWPN	25:00:00:21:88:00:54:4b
Target WWPN	25:10:00:21:88:00:54:4b

At the bottom right are 'OK' and 'Cancel' buttons.

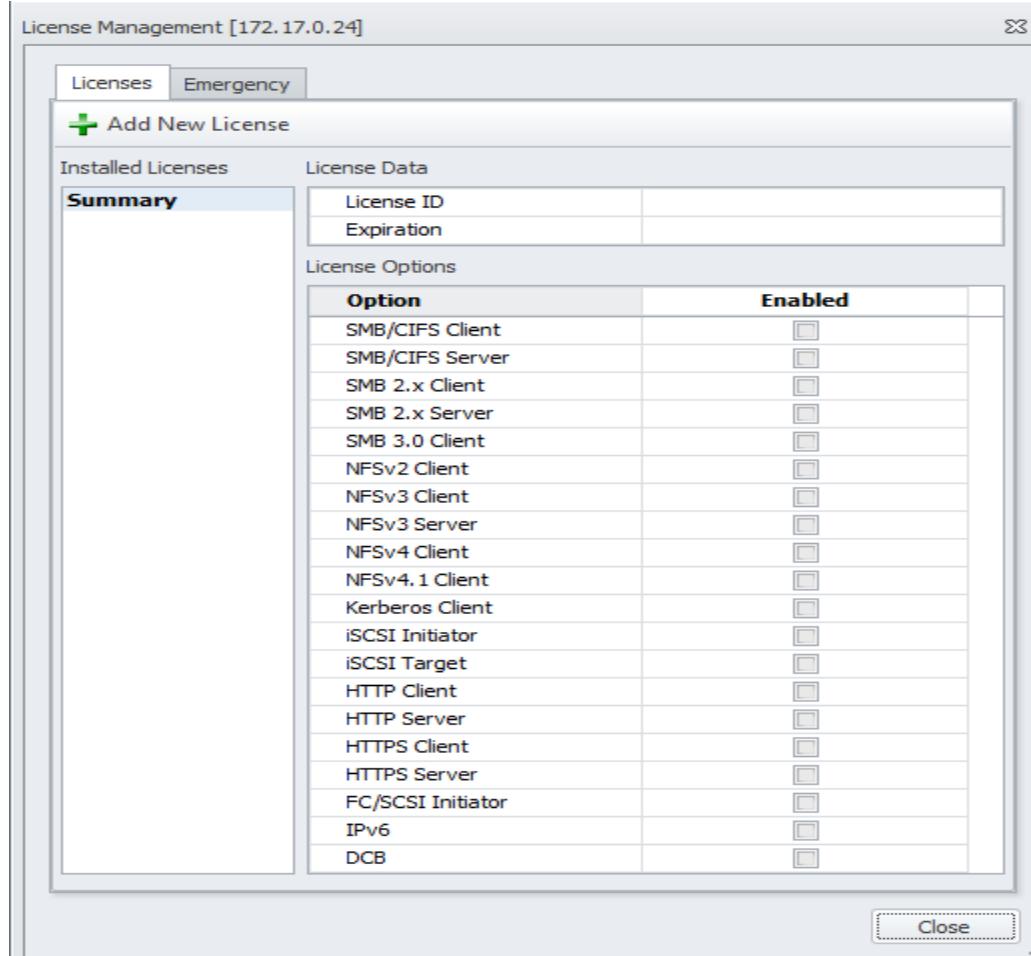
Click the Licenses button and if the following dialog appears, the Appliance is not currently not running the appropriate revision of the Load DynamiX Firmware



If instead of the error dialog above, the user gets the following message then the Appliance is ready to be Activated. If the Appliance has already been Activated then either the user will see the empty License window show in the Ready For Licenses section or the Licensed window shown in the Licensed section.

The screenshot shows the 'Activation Management' dialog. A message box displays: 'System requires activation and requires license. 1) Request an activation. Save activation file and e-mail to SwiftTest support at: support@swiftest.com. 2) Submit received data.' Below this, there are two radio buttons: 'Request Activation' (selected) and 'Submit Activation'. At the bottom are 'Request' and 'Cancel' buttons.

Ready For Licenses: An Appliance is ready for licenses once it has been Activated. Clicking on the License button  in the Ports & Appliances > Appliances Tab will open the following License Management window which shows no licenses installed yet. This Appliance will not run any Load DynamiX Projects because none of the Protocols are licensed.



Licensed: When the License Management window shows active licenses then the Appliance is ready to run Projects that contain the licensed Protocols (**and ONLY the licensed Protocols**). The following License Management window shows active licenses.

License Management [172.17.1.49]



Licenses Emergency

Add New License

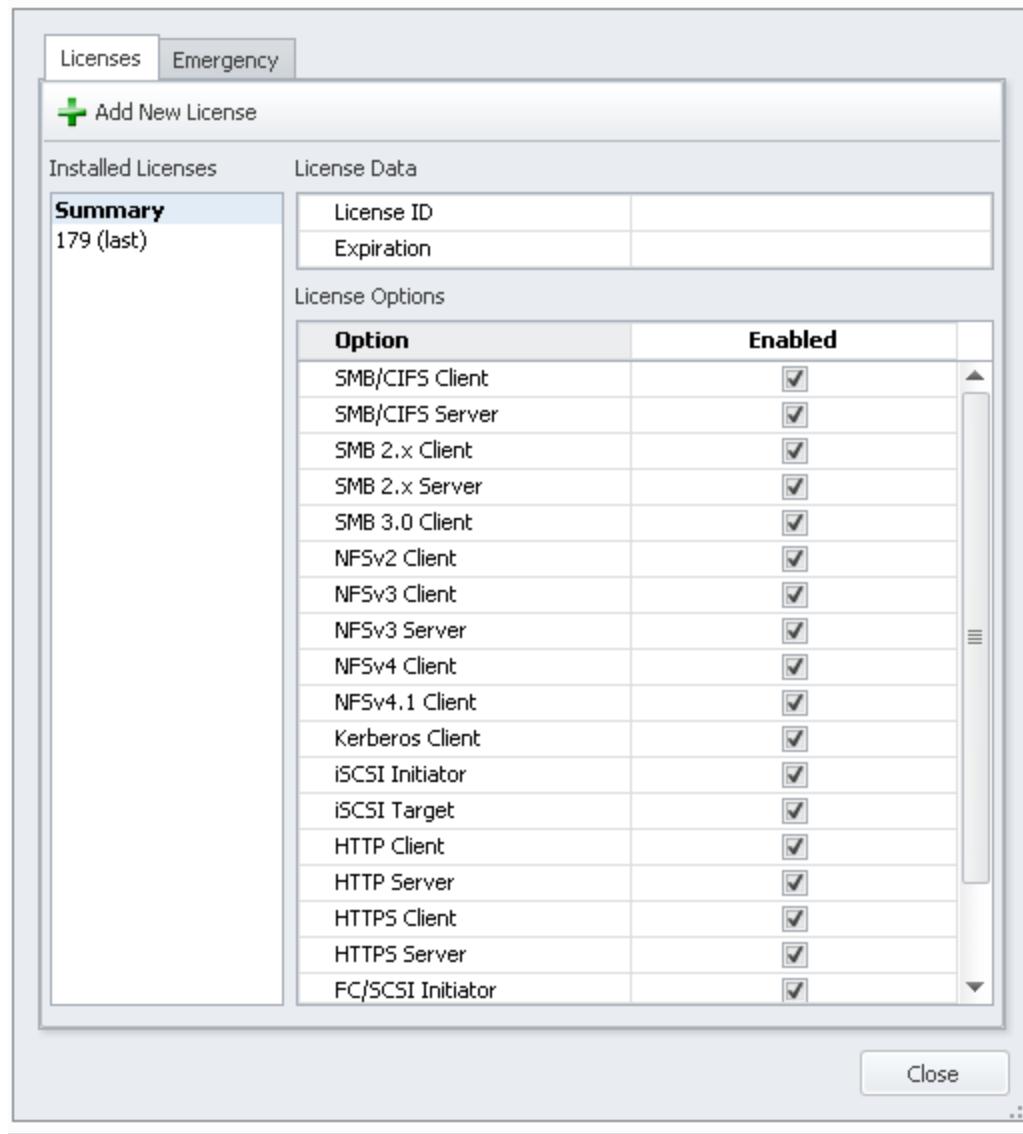
Installed Licenses	License Data																																						
Summary 179 (last)	<table border="1"> <tr> <td>License ID</td> <td></td> </tr> <tr> <td>Expiration</td> <td></td> </tr> </table>	License ID		Expiration																																			
License ID																																							
Expiration																																							
License Options <table border="1"> <thead> <tr> <th>Option</th> <th>Enabled</th> </tr> </thead> <tbody> <tr><td>SMB/CIFS Client</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>SMB/CIFS Server</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>SMB 2.x Client</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>SMB 2.x Server</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>SMB 3.0 Client</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>NFSv2 Client</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>NFSv3 Client</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>NFSv3 Server</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>NFSv4 Client</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>NFSv4.1 Client</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>Kerberos Client</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>iSCSI Initiator</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>iSCSI Target</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>HTTP Client</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>HTTP Server</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>HTTPS Client</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>HTTPS Server</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>FC/SCSI Initiator</td><td><input checked="" type="checkbox"/></td></tr> </tbody> </table>		Option	Enabled	SMB/CIFS Client	<input checked="" type="checkbox"/>	SMB/CIFS Server	<input checked="" type="checkbox"/>	SMB 2.x Client	<input checked="" type="checkbox"/>	SMB 2.x Server	<input checked="" type="checkbox"/>	SMB 3.0 Client	<input checked="" type="checkbox"/>	NFSv2 Client	<input checked="" type="checkbox"/>	NFSv3 Client	<input checked="" type="checkbox"/>	NFSv3 Server	<input checked="" type="checkbox"/>	NFSv4 Client	<input checked="" type="checkbox"/>	NFSv4.1 Client	<input checked="" type="checkbox"/>	Kerberos Client	<input checked="" type="checkbox"/>	iSCSI Initiator	<input checked="" type="checkbox"/>	iSCSI Target	<input checked="" type="checkbox"/>	HTTP Client	<input checked="" type="checkbox"/>	HTTP Server	<input checked="" type="checkbox"/>	HTTPS Client	<input checked="" type="checkbox"/>	HTTPS Server	<input checked="" type="checkbox"/>	FC/SCSI Initiator	<input checked="" type="checkbox"/>
Option	Enabled																																						
SMB/CIFS Client	<input checked="" type="checkbox"/>																																						
SMB/CIFS Server	<input checked="" type="checkbox"/>																																						
SMB 2.x Client	<input checked="" type="checkbox"/>																																						
SMB 2.x Server	<input checked="" type="checkbox"/>																																						
SMB 3.0 Client	<input checked="" type="checkbox"/>																																						
NFSv2 Client	<input checked="" type="checkbox"/>																																						
NFSv3 Client	<input checked="" type="checkbox"/>																																						
NFSv3 Server	<input checked="" type="checkbox"/>																																						
NFSv4 Client	<input checked="" type="checkbox"/>																																						
NFSv4.1 Client	<input checked="" type="checkbox"/>																																						
Kerberos Client	<input checked="" type="checkbox"/>																																						
iSCSI Initiator	<input checked="" type="checkbox"/>																																						
iSCSI Target	<input checked="" type="checkbox"/>																																						
HTTP Client	<input checked="" type="checkbox"/>																																						
HTTP Server	<input checked="" type="checkbox"/>																																						
HTTPS Client	<input checked="" type="checkbox"/>																																						
HTTPS Server	<input checked="" type="checkbox"/>																																						
FC/SCSI Initiator	<input checked="" type="checkbox"/>																																						

Close

Licensing Process:**This process must be repeated for every Appliance that is to be Licensed.**

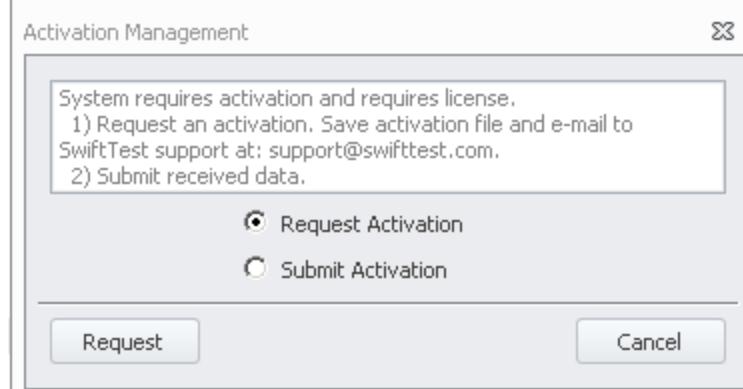
If an Appliance is Licensed, the set of Protocols which are licensed an Appliance will appear when the Ports & Appliances > Appliances Tab, Licenses button is pressed. Licensed Protocols will be indicated by check marks in the Enabled column as seen here.

License Management [172.17.1.49]



If the Appliance is not Activated then open the Ports & Appliances window > Appliances tab.

Highlight the Appliance to be Activated and click the Licenses button . When the Activation dialog appears



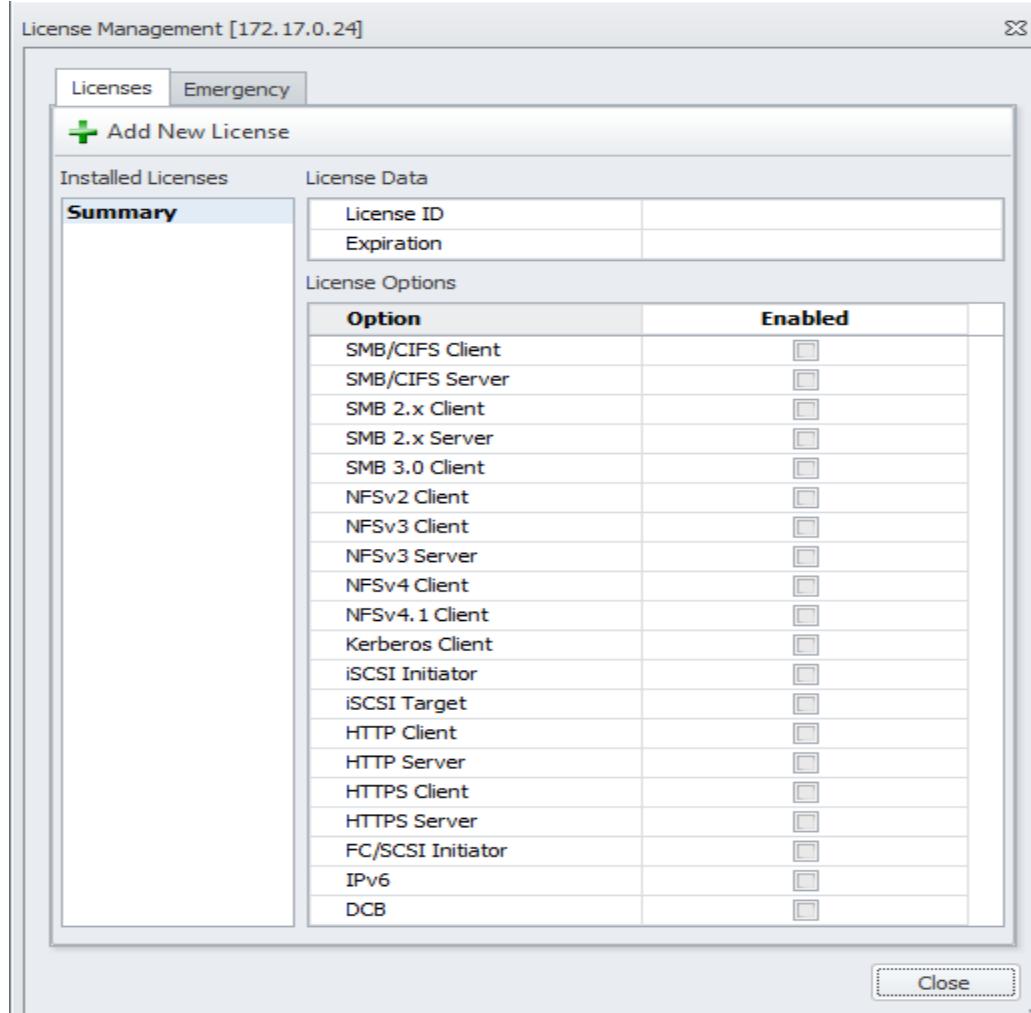
click the Request Activation radio button and then click . The TDE will open a dialog box which allows the user to save the Activation File on his computer's hard drive. The file will be named <Appliance Mac Address>_request.txt.

This file must be emailed to [Load DynamiX Support](#) so that Support can create the Activation Record and License File for this Appliance. Save the file to a folder and then email it as an attachment to [Load DynamiX Support](#). When [Load DynamiX Support](#) responds to an email they will provide the user with two files:

Activation Record: <Appliance Mac Address>_<Appliance Serial Number>.activation.txt
 License File: <Appliance Serial Number>_license.txt

Save these files to the computer's hard drive.

Click the Licenses button  in the Ports & Appliances >Appliance tab with the Appliance to be licensed highlighted and then click the Submit Activation radio button and select the <Appliance Mac Address>_<Appliance Serial Number>.activation.txt file. The TDE will deploy the Activation Record and then immediately open the License Management window.



Click the Add New License button  and select the License File: <Appliance Serial Number>_license.txt and the contents of the License File will be displayed in the window .

Add License File



C:\Users\SwiftTest\Downloads\ST0041_license (Permanent).txt

Select License File

```
----- BEGIN -----  
SwiftTest SN: ST0041. License ID: 178.  
  
Expiration: Permanent  
  
Includes license options:  
  
SMB/CIFS Server  
SMB 2.x Client  
SMB 2.x Server  
SMB 3.0 Client  
NFSv2 Client  
NFSv3 Client  
NFSv3 Server  
NFSv4 Client  
NFSv4.1 Client  
Kerberos Client  
iSCSI/SCSI Initiator  
iSCSI/SCSI Target  
HTTP Client  
HTTP Server  
HTTPS Client  
HTTPS Server  
FC/SCSI Initiator  
IPv6
```

Submit License

Cancel

Submit License

Click **Submit License** to deploy the License File and the supported Protocols will be displayed on the right side of the window.

License Management [172.17.1.49]



Licenses Emergency

Add New License

Installed Licenses	License Data																																						
Summary 179 (last)	<table border="1"> <tr> <td>License ID</td> <td></td> </tr> <tr> <td>Expiration</td> <td></td> </tr> </table>	License ID		Expiration																																			
License ID																																							
Expiration																																							
License Options <table border="1"> <thead> <tr> <th>Option</th> <th>Enabled</th> </tr> </thead> <tbody> <tr><td>SMB/CIFS Client</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>SMB/CIFS Server</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>SMB 2.x Client</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>SMB 2.x Server</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>SMB 3.0 Client</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>NFSv2 Client</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>NFSv3 Client</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>NFSv3 Server</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>NFSv4 Client</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>NFSv4.1 Client</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>Kerberos Client</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>iSCSI Initiator</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>iSCSI Target</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>HTTP Client</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>HTTP Server</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>HTTPS Client</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>HTTPS Server</td><td><input checked="" type="checkbox"/></td></tr> <tr><td>FC/SCSI Initiator</td><td><input checked="" type="checkbox"/></td></tr> </tbody> </table>		Option	Enabled	SMB/CIFS Client	<input checked="" type="checkbox"/>	SMB/CIFS Server	<input checked="" type="checkbox"/>	SMB 2.x Client	<input checked="" type="checkbox"/>	SMB 2.x Server	<input checked="" type="checkbox"/>	SMB 3.0 Client	<input checked="" type="checkbox"/>	NFSv2 Client	<input checked="" type="checkbox"/>	NFSv3 Client	<input checked="" type="checkbox"/>	NFSv3 Server	<input checked="" type="checkbox"/>	NFSv4 Client	<input checked="" type="checkbox"/>	NFSv4.1 Client	<input checked="" type="checkbox"/>	Kerberos Client	<input checked="" type="checkbox"/>	iSCSI Initiator	<input checked="" type="checkbox"/>	iSCSI Target	<input checked="" type="checkbox"/>	HTTP Client	<input checked="" type="checkbox"/>	HTTP Server	<input checked="" type="checkbox"/>	HTTPS Client	<input checked="" type="checkbox"/>	HTTPS Server	<input checked="" type="checkbox"/>	FC/SCSI Initiator	<input checked="" type="checkbox"/>
Option	Enabled																																						
SMB/CIFS Client	<input checked="" type="checkbox"/>																																						
SMB/CIFS Server	<input checked="" type="checkbox"/>																																						
SMB 2.x Client	<input checked="" type="checkbox"/>																																						
SMB 2.x Server	<input checked="" type="checkbox"/>																																						
SMB 3.0 Client	<input checked="" type="checkbox"/>																																						
NFSv2 Client	<input checked="" type="checkbox"/>																																						
NFSv3 Client	<input checked="" type="checkbox"/>																																						
NFSv3 Server	<input checked="" type="checkbox"/>																																						
NFSv4 Client	<input checked="" type="checkbox"/>																																						
NFSv4.1 Client	<input checked="" type="checkbox"/>																																						
Kerberos Client	<input checked="" type="checkbox"/>																																						
iSCSI Initiator	<input checked="" type="checkbox"/>																																						
iSCSI Target	<input checked="" type="checkbox"/>																																						
HTTP Client	<input checked="" type="checkbox"/>																																						
HTTP Server	<input checked="" type="checkbox"/>																																						
HTTPS Client	<input checked="" type="checkbox"/>																																						
HTTPS Server	<input checked="" type="checkbox"/>																																						
FC/SCSI Initiator	<input checked="" type="checkbox"/>																																						

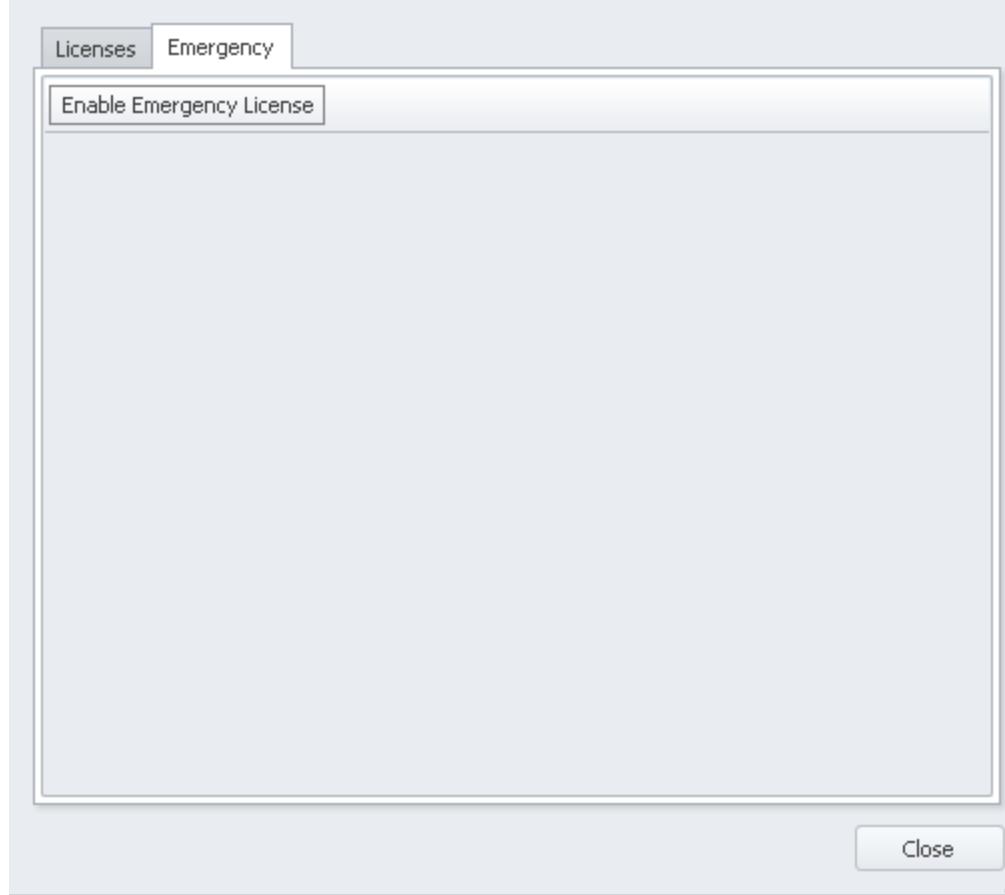
Close

The Appliance is now ready to run projects that contain the Protocols that appear on the right side of the window.

Emergency License

If for some reason, the License sent by Load DynamiX for a specific Appliance does not work, the user may create an Emergency License which will enable all Protocols for 7 days. To enable an Emergency License, open the License Management window by clicking the Licenses button in the Appliances Tab. Then click the Emergency Tab and click Enable Emergency. A 7 day License for all Protocols will be enabled. Once an Emergency License expires, it cannot be enabled again until a Permanent or Temporary License has been installed.

License Management [172.17.1.49]



Test Development Environment and GUI

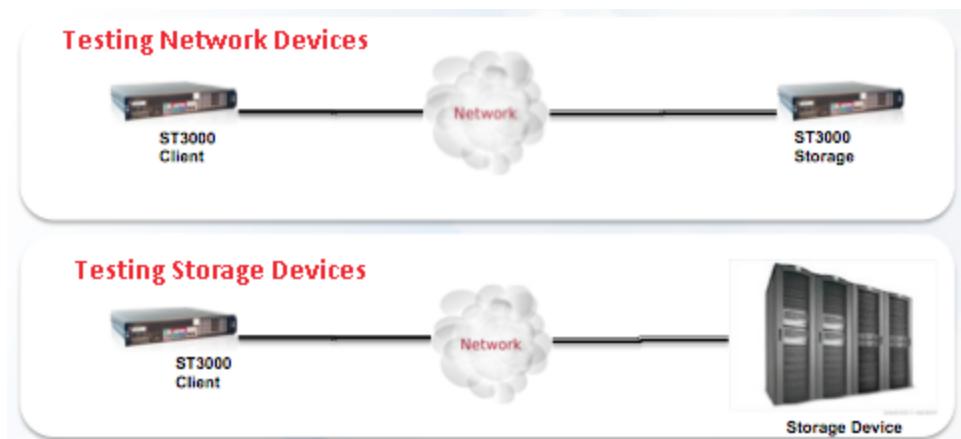
Load DynamiX Test Development Environment and GUI

This chapter provides an overview of the Load DynamiX Test Development Environment (TDE) for developing and executing test Projects against network storage devices and services and a detailed discussion of the capabilities of the Load DynamiX TDE's Graphical User Interface (GUI).

Windows .NET framework version 4.x must be installed - see [Product Installation section for details.](#)

Functional Overview

The Load DynamiX product comes with an Appliance that can have any Test Port configured as a Client or Server. The Load DynamiX Client can simulate large numbers of clients accessing Fibre Channel, Object (CDMI, Open Stack Swift), HTTP, HTTPS, CIFS-SMB, SMB2, iSCSI, or NFS-based devices or services. For this application the figure below illustrates the typical test environments in which the Load DynamiX Appliance and software deployed - testing network device such as WAN accelerators and testing storage devices using the CIFS-SMB, NFS or iSCSI storage protocols.



Important components of the test environment:

- Load DynamiX Test Development Environment (TDE) – A graphical software application (using Microsoft® Windows® .NET) used by a test or development engineer (the Tester) to create, modify, and execute tests and review test results. Multiple components make up a Project (these are described in detail in this and the [Test Creation section](#)). The TDE creates and downloads tests to the Load DynamiX Appliance to be executed.
- Load DynamiX Appliance (Appliance) – Test Ports on the Appliance are configured as Logical Ports operating as a Load DynamiX Client or Load DynamiX Server, generating and responding to Fibre Channel, Object (CDMI, Open Stack Swift), HTTP, HTTPS, CIFS-SMB, iSCSI, or NFS traffic.
- Load DynamiX Client – Simulates up to many thousands of end users accessing Fibre Channel, Object (CDMI, Open Stack Swift), HTTP, HTTPS, CIFS-SMB, iSCSI, or NFS servers.
- Load DynamiX Server – Simulates HTTP, CIFS-SMB, iSCSI or NFSv3 servers.

Projects

Tests are organized as Projects. A Project consists of:

- Timeline – The time-ordered display of the resources (test components) that make up the test.

- Resources – Test components that execute during the Project Timeline.

For a list of the kinds of information typically required to design a Load DynamiX Project, see the Information Typically Required for Project Design section in the [Introduction chapter](#).

Projects are stored in folders that contain the files describing the Timeline and Resources required. The default location for all user created or modified Projects is the Projects folder (see the [Product Installation chapter](#) for the default location of this folder).

See [Test Creation](#) for details on creating a Project.

Resources

Resources are the components used to construct a test. The Sample Resources folder (accessed via the Resources Library) contains templates that can be customized to meet Project needs. Resources are dragged into the Project Explorer or Timeline to associate them with a Project and edited for specific Project needs. Some Resource templates are empty and require Tester-defined content (Sample Client Scenario, Sample Server Scenario, Sample User Parameters, etc.) and some are more ready to use with minimal editing (any of the Advanced Load Profile Samples).

Resources Library	
Name	Type
> My Resources	
> Sample Resources	
> Advanced Load Profile Samples	
Sample Client Scenario	Client Scenario
Sample File System	Data File System
Sample Load Profile	Load Profile
Sample Network Profile	Network Profile
Sample Server Scenario	Server Scenario
Sample Test Exec Rules Errors	Test Execution Rules
Sample Test Exec Rules Notify	Test Execution Rules
Sample Test Exec Rules Warning	Test Execution Rules
Sample User Parameters	User Parameters

The My Resources sub-folder can be used to save Tester-developed Resources. This folder can contain sub-folders so that Testers can organize their saved Resources by Resource type or Protocol type or whatever categorization the Tester prefers. See the [Test Creation](#) for details on adding and using Resources in a Project.

Toolbox

The Toolbox contains the Actions that correspond to Fibre Channel, HTTP, HTTPS, iSCSI, NFS v2, v3, v4, and v4.1, CIFS-SMB, SMB2, SMB3, HTTP Storage(Amazon S3, CDMI, OpenStack Swift), TCP Echo/Discard protocol commands, Kerberos commands and Load DynamiX Scenario Control Actions. Actions are used in Client or Server Scenarios to create the ordered sequences of operations that make up a Load DynamiX Project. Each Action has a set of properties that are the input to that Action. The properties/input for an Action can be viewed and updated once the Action is in a Scenario.

Usability tips:

- Dragging an Action from the Toolbox onto an existing Scenario will insert the dragged Action above the Action it is dropped onto.

- Holding the Ctrl key down when the Action is dropped will cause the new Action to replace the action it is dropped onto.
- Double-clicking the Action in the Toolbox will insert it into the current scenario below the currently highlighted Action or at the bottom of the Scenario if no Action is highlighted.
- Holding the Ctrl key down when double-clicking an Action in the Toolbox will cause it to replace the currently highlighted Action.

Test Components

The Required column indicates which components are mandatory and which are optional in Load DynamiX test Projects.

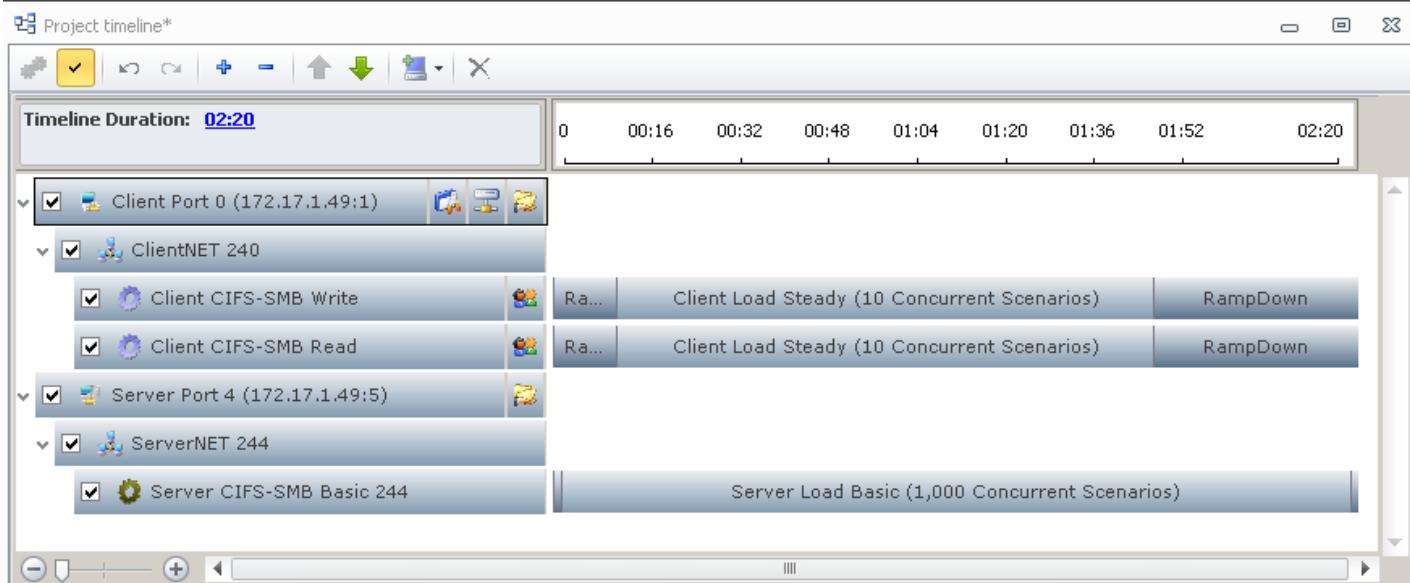
Component	Required	Description
Timeline	Mandatory	Graphical view of the test – it includes the individual components that make up a test (resources, duration, etc).
Logical Ports	Mandatory	Port(s) used by the Appliance when executing a Scenario. Logical Ports contain the definition of the Scenario(s) to be executed and the network (Network Profile) that the Scenario will execute in.
Physical Ports	Mandatory	Ports on the Load DynamiX Appliance that must be linked to Logical Ports before a test can be run.
Scenario	Mandatory	Consists of one or more Fibre Channel , CIFS-SMB , SMB2 , NFS v2 , v3 , v4 , Kerberos , iSCSI , HTTP/HTTPS , TCP Echo , Object(CDMI or OpenStack Swift) , or Scenario Control Actions for execution by the Client or Server during a test. Actions contain input fields that may allow Functions or Formula. Functions allow creating contents variables for input fields from elements that are either strings, random numbers or input from User Parameter files. Functions always create data that are strings. Formula return mathematical results.
Load Profile	Mandatory	Amount of load to be generated for each Scenario.
Network Profile	Mandatory	Network (IPv4/IPv6, DNS, TCP, Mac/Man, Fibre Channel) characteristics for Client and/or Server subnets in a Project.
User Parameters	Optional	Mechanism that allows parameterization of Scenario Action inputs.
Data File System	Optional	Set of files that a client can write to a Device Under Test (real server or a Load DynamiX Server), and files and directories that can be read from a Load DynamiX Server by a Client during a test.
Test Execution Rules	Optional	Rules based on tests of statistics or counter values used to control Project behavior at run time.
Text	Optional	Text that can be saved with Projects to describe anything about the Project.

Timeline

Testers construct a Timeline containing the test components that make up a Project. Minimally the timeline includes the Mandatory test components:

- Logical Port(s) : Port(s) used by the Appliance when executing a Scenario.
- Network Profile(s) : Network characteristics for subnets on a Logical Port.
- Client or Server Scenario(s) : One or more [Fibre Channel](#), [SCSI](#), [CIFS-SMB](#), [SMB2](#), [NFS v2](#), [v3](#), [v4](#), [v4.1](#), [Kerberos](#), [iSCSI](#), [HTTP/HTTPS](#), [Object \(CDMI, OpenStack Swift\)](#) or [Scenario Control](#) Actions for execution by the Client or Server during a test.
- Load Profile(s) : Amount of load to be generated for each Scenario.

- Test Duration : The length of time that the project will execute. Maximum Test Duration for a Project executing on a Physical Test Port is 1000 hours. For the maximum Test Duration of a Virtual Appliance port see [Appendix: Virtual Appliance Constraints and Licensing](#).

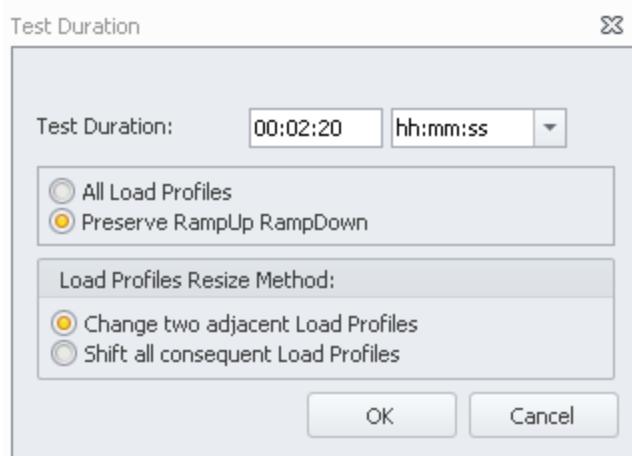


Timeline Duration

Multiple uses:

- Configuration: To specify the length of time a Project will execute including ramp up and ramp down time and all Load Profiles
- Configuration: To specify the behavior of RampUp, RampDown and Load Profiles when the overall Duration is changed
- Configuration: To specify the behavior of Load Profiles when resized (time increased or decreased)
- Runtime: When a Project is executing, the elapsed time of the run.

When the Tester clicks the Timeline Duration value, the following interface is presented



Test Duration: Specify the overall length of Project execution in a variety of formats (days/hours/minutes/seconds, hours/minutes/seconds, seconds, etc)

Duration Change behaviors (default value is Preserve RampUp and RampDown): applies only when the overall Duration is changed.

- All Load Profiles: When the Test Duration value is changed, change all Load Profile times

- proportionally to the change in overall Duration.
- Preserve RampUp and RampDown: When the Test Duration value is changed, leave the Ramp Up and Ramp Down times as they are.

Load Profile Resize behaviors (default behavior is Change two adjacent Load Profiles): applies only when Load Profiles are resized on the TDE timeline

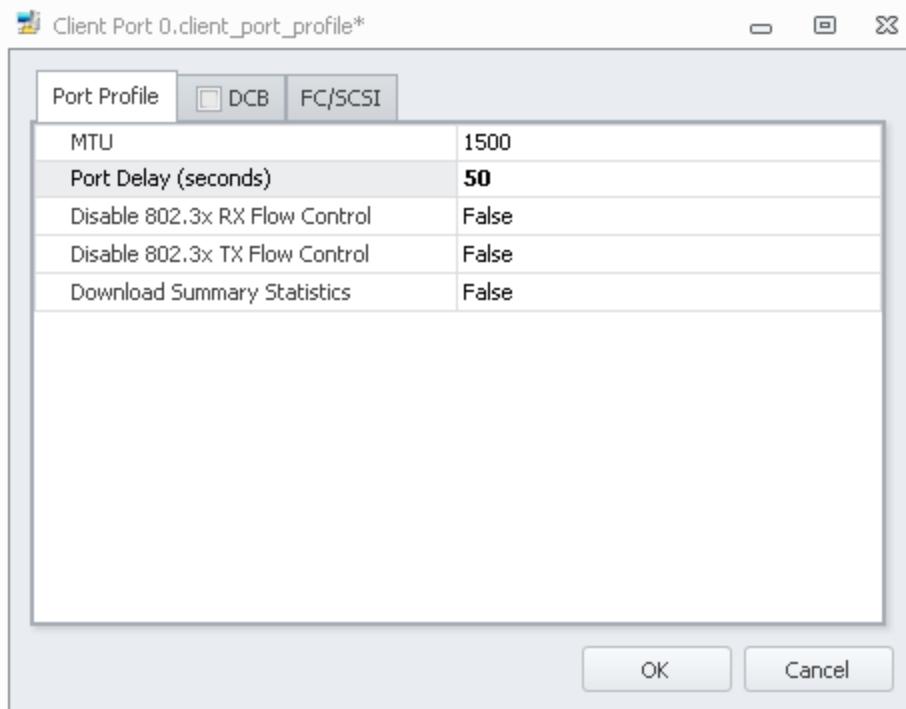
- Change two adjacent Load Profiles: Shift a Load Profile right edge to the left (decrease Load Profile Duration and leave overall Timeline Duration the same) or to the right (increase Load Profile Duration and leave overall Timeline Duration the same).
- Shift all consequent Load Profiles: Shift a Load Profile right edge to the left (decrease Load Profile Duration, decrease overall Timeline Duration) or to the right (increase Load Profile Duration, increase overall Timeline Duration).

Physical Ports

Port(s) used by the Appliance when executing a Scenario. An Appliance has eight physical 1000BASE-T ports (Load DynamiX 1G Series Model 3000/3108) or two physical 10-Gbps ports (Load DynamiX 10G Series Model 5000/5102) or eight 10GBASE-T ports (Load DynamiX 10G Series Model 5108T) or eight 10Gbps ports (Load DynamiX 10G Series Model 5108S) or two 10Gbps Fibre Channel over Ethernet ports (Load DynamiX FC Series Model 6202E) or two 16Gbps Fibre Channel ports (Load DynamiX FC Series Model 6202) or four 16Gbps Fibre Channel ports (Load DynamiX FC Series Model 6204) or eight 16Gbps Fibre Channel ports (Load DynamiX FC Series Model 6208) or a combination of 16Gbps FC Series and 10G Series physical ports (U1022 and U1044 Unified Series) that can run either Client or Server Scenarios. Physical Ports are assigned to Logical Ports in the Ports & Appliances window. Physical Ports from one or more Appliances can be used in a Project.

Logical Ports

Port(s) used by the Appliance when executing a Scenario. Each Logical Port can have multiple Network Profiles assigned to it (each Network Profile in the Project must have at least one Scenario and each Scenario must have at least one Load Profile). Logical Ports are added to a Project in the Timeline window by dragging a Client or Server Logical Port from the Toolbox. Physical Ports are assigned to Logical Ports in the Ports & Appliances window.



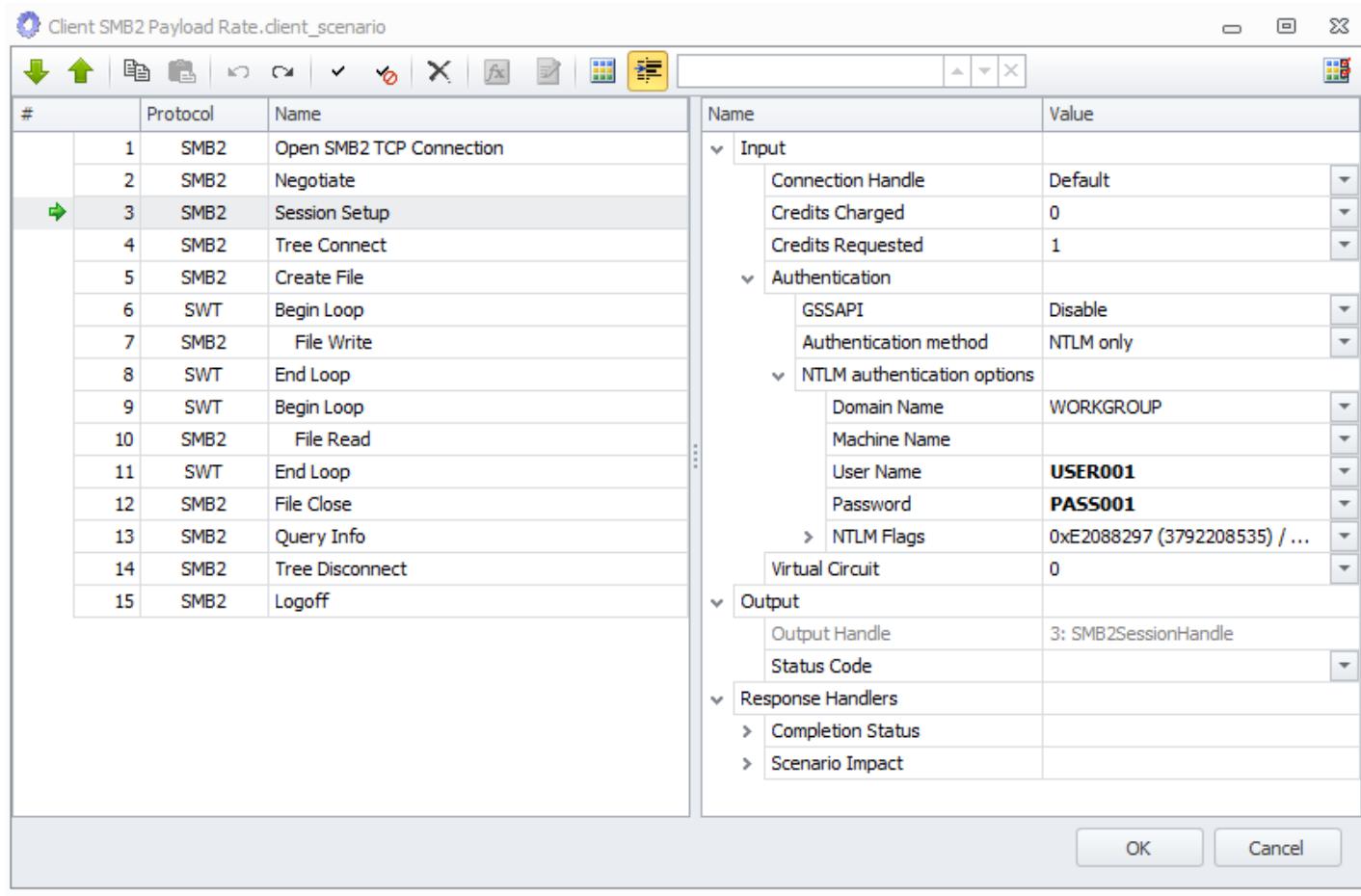
Logical Ports contain several properties that may be important to correct Project execution:

- The MTU property. See [Appendix: Jumbo Frames and Delayed ACK](#) for details on MTU size. Default == 1500.
- The Port Delay property. To activate the Ethernet ports that are used in this Project but delay the Project starting for the specified number of seconds. Useful in Spanning Tree environments because of the time it takes (up to 30 seconds) for Spanning Tree activities to complete. Useful in Fibre Channel environments because of the time it takes for the Fibre Channel driver to detect the network configuration it is connected to. Default == 0.
- The Disable 802.3x Rx and Tx Control properties. To disable data link layer flow control for received and/or sent packets. Default == Enabled.
- The Download Summary Statistics property. To enable downloading of the summary statistics file at Project completion. Default == Disabled. See more information on Summary Statistics in [Appendix: Jumbo Frames and Delay ACK](#).
- DCB Tab. See [Appendix: DCB/DCBX](#) for DCB and DCBX control details. Default == Disabled.
- FC/SCSI Tab. See [Reference FC/SCSI/iSCSI Commands and Behaviors](#) for MPIO Enabled and Inactivity Timer, IO Timeout, Port Queue Depth and Per LUN Statistics control details. MPIO default Enabled == False, Inactivity Timer default == 30,000, IO Timeout default == 5000, Per LUN Statistics default == False, Port Queue Depth default == 32.

Scenario

A Tester creates one or more Scenarios for each Client or Server in the Project. A Scenario consists of one or more Actions. One Action exists for each HTTP/HTTPS, CIFS-SMB, NFS, Kerberos, SCSI, Fibre Channel, iSCSI or HTTP Storage command supported by the Appliance. Links to protocol reference materials are provided in the [References and Terminology section](#).

The figure below shows a Client Scenario using SMB2 Actions.



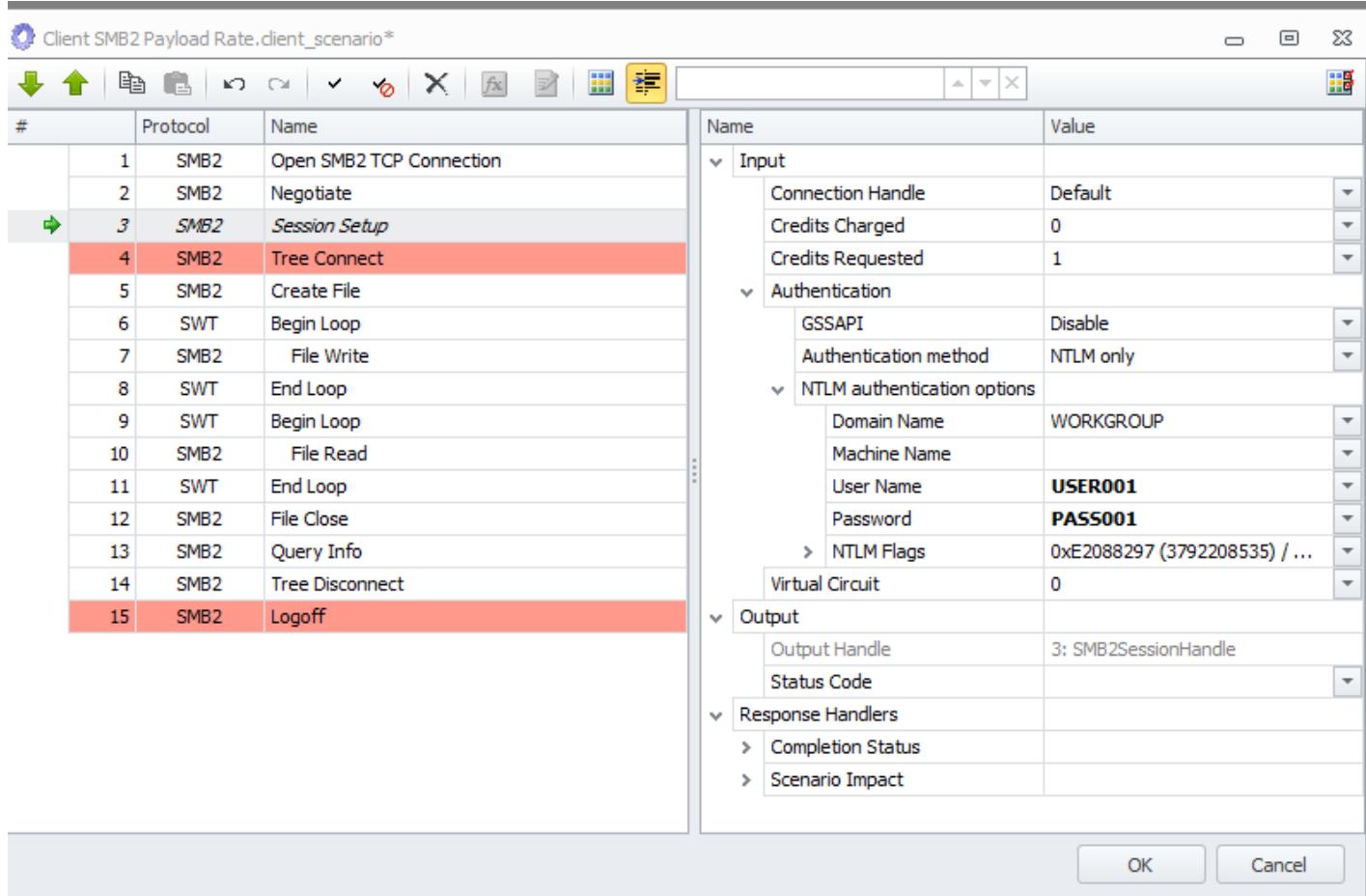
Action Inputs

Inputs to Actions (file names, block count, user id, password, etc.) control the behaviors of the Actions. This Help document does not attempt to define all of the Inputs to all of the Actions supported by the Appliance. In the Advanced Concept, Appendix and Reference sections of this document, details of some of the more complicated Action's inputs and behaviors are described.

Scenario Editor

One of the troubleshooting features of the Scenario Editor is its ability to determine the impact of Enabling or Disabling Actions on the rest of the Actions in a Scenario. The feature is based on the concept of "Handles" which are the output of one Action and input to another. In certain Load DynamiX Actions, the output of the Action is a Handle of a specific type. For example, in the Scenario above, the output of the Session Setup Action is a Handle of type **SMB2SessionHandle**. This Handle is input to the next Action in the Scenario - the **Tree Connect** Action and to the **Session Logoff** Action at the bottom of the Scenario.

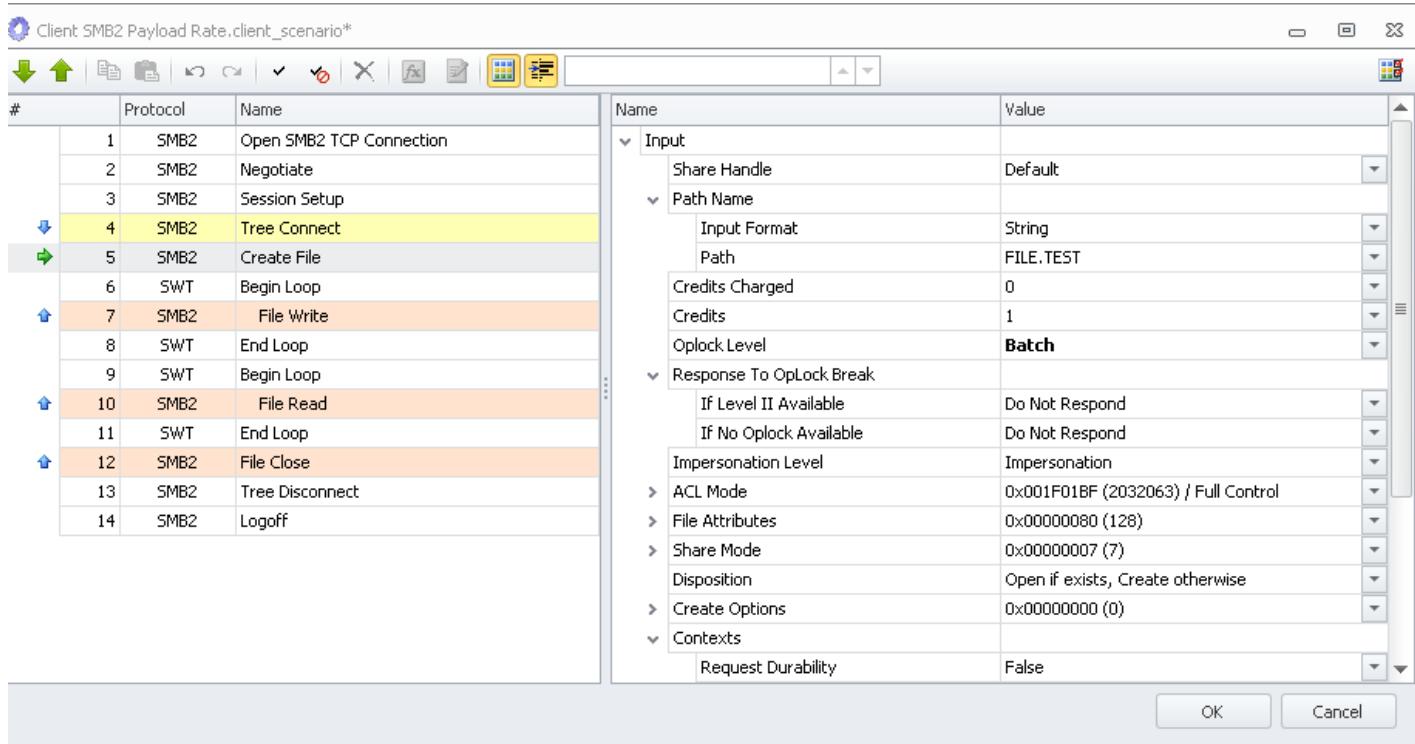
If the **Session Setup** Action were disabled or missing, the **Tree Connect** Action would not have the **SMB2SessionHandle** that it requires to make a successful Tree Connection so the Scenario would be guaranteed to fail. To help the Tester recognize this situation, the Scenario Editor provides warnings when this type of condition (missing Handle) is detected. The Scenario Editor issues the warning by highlighting the impacted Action(s) in **RED**. So, when the Session Setup Action is Disabled or if it were missing, the following is shown in the Scenario Editor. The **Session Setup** Action is highlighted in **Italics** to indicate that it has been Disabled and the **Tree Connect** and **Session Logoff** Actions are highlighted in **RED** to indicate that they are missing a required input. Actions are also flagged in **RED** if any of the inputs required by the Action are not present or available (such as a require Handle not being present in the Scenario).



These kinds of issues would also be detected when the Scenario is compiled (see [Executing Tests and Assessing Results](#)).

Another feature of the Scenario Editor is Action hierarchy coloring. In Load DynamiX Scenarios, Actions output a Handle which can be used by other Actions as input. For example, in the SMB2 Scenario screenshot above, the output of the line 3 (**Session Setup** Action) is a handle of type SMB2SessionHandle. This handle is used as input by the line 4 (**Tree Connect** Action). Throughout the Scenario there are a series of relationships built based on the output of a Handle by one Action and the use of that Handle as input by one or more other Actions. For example, the output of the **Create File** Action is used as input by the **File Write**, **File Read** and **File Close** Actions.

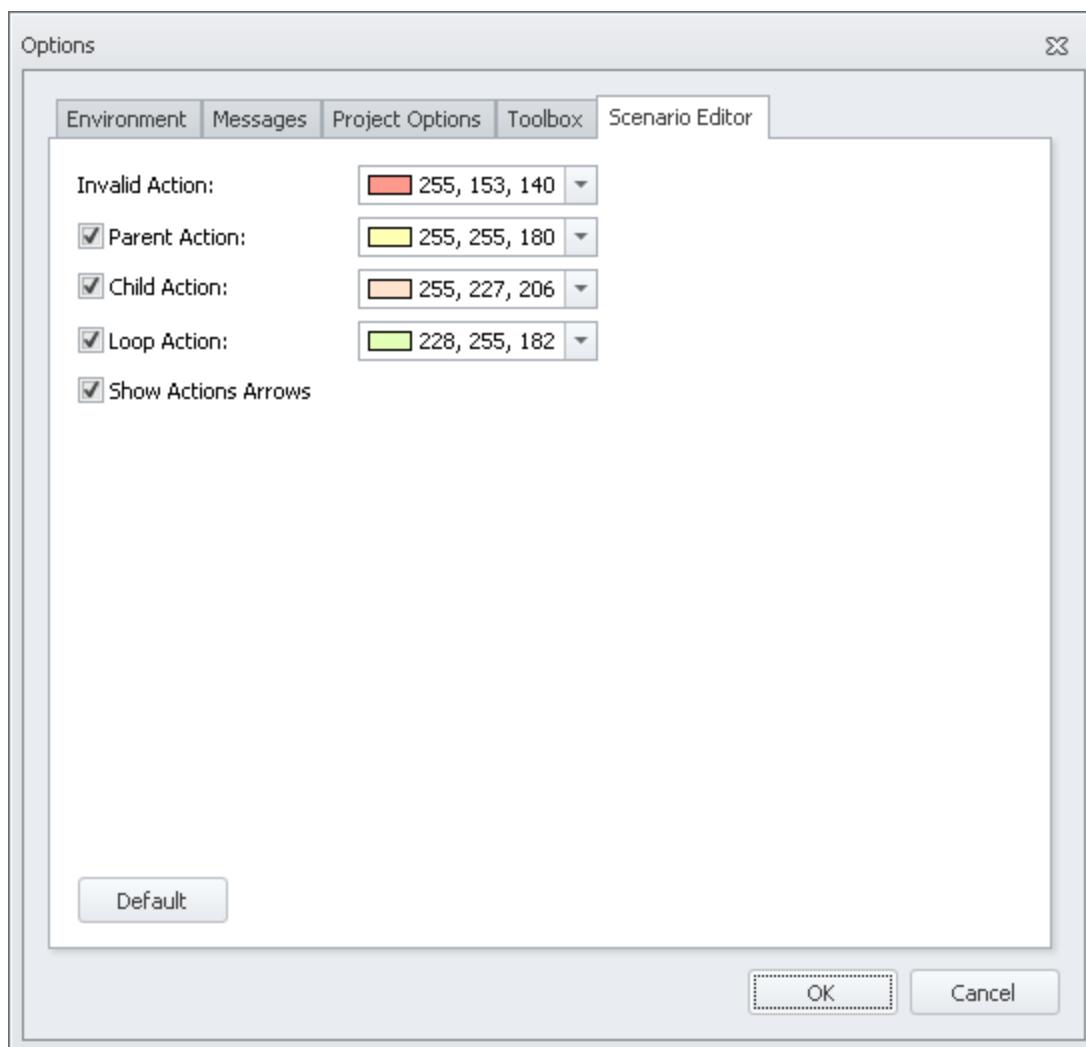
If the **Create File** Action is highlighted (by clicking on the **Create File** Action) and the Action Dependency Highlighting button is enabled, the Scenario Editor highlights the Handle relationships for the Create File Action. This button functions as the On/Off switch for coloring. If NO coloring is required, click the button into the Off state (no yellow color background on the button). If coloring is desired, click it into the On state (yellow background as shown above)



Notice the color schemes and arrow icons. The dark blue Action is the **Create File** Action that was highlighted initially. It has a green arrow on the left. The Tree Connect Action is highlighted in yellow and it has a blue down arrow on the left indicating that this is the Parent Action of the highlighted Action (i.e. the highlighted Action is dependent on its Parent Action output Handle). The File Write, **File Read** and **File Close** Actions are highlighted in a salmon color and they all have blue up arrows on the left indicating that they are Child Actions (i.e. they are dependent on the Handle output by the highlighted Action). This feature can be very helpful in debugging complex Scenarios that contain many Actions with many different relationship trees.

The color scheme and the arrow icons can be configured using the Scenario Editor window below. This window can be opened two ways

- Select View>Options>Scenario Editor from the main toolbar.
- Click the Scenario Editor Options button on the right side of the Scenario window.



To change the color scheme simply change the color for the highlighting action (Invalid Action defaults to red, Parent Action defaults to yellow and Child Action defaults to salmon. To enable or disable the coloring of Actions for specific relationships, click the Check Boxes to the left of the colors.

To enable or disable the presence of arrow icons, click the check box to the left of Show Actions Arrows.

Click **OK** to save changes, **Cancel** to exit the window without making changes, and **Default** to change settings back to the factory defaults.

Indentation

This Scenario demonstrates the use of the Indentation highlight button . When the Indentation highlight button is enabled, the Scenario elements are shown indented where the indentation is appropriate. Below all of the Scenario Actions except the File Write and File Read Actions are not indented, whereas the **File Write** and **File Read** Actions are indented because they are within Begin Loop - End Loop blocks.

The screenshot shows the 'Client SMB2 Payload Rate.client_scenario*' dialog box. On the left is a list of steps:

#	Protocol	Name
1	SMB2	Open SMB2 TCP Connection
2	SMB2	Negotiate
3	SMB2	Session Setup
4	SMB2	Tree Connect
5	SMB2	Create File
6	SWT	Begin Loop
7	SMB2	File Write
8	SWT	End Loop
9	SWT	Begin Loop
10	SMB2	File Read
11	SWT	End Loop
12	SMB2	File Close
13	SMB2	Tree Disconnect
14	SMB2	Logoff

On the right, the configuration details for step 4 ('Tree Connect') are shown:

Name	Value
Input	
Share Handle	Default
Path Name	
Input Format	String
Path	FILE.TEST
Credits Charged	0
Credits	1
Oplock Level	Batch
Response To OpLock Break	
If Level II Available	Do Not Respond
If No Oplock Available	Do Not Respond
Impersonation Level	Impersonation
ACL Mode	0x001F01BF (2032063) / Full Control
File Attributes	0x00000080 (128)
Share Mode	0x00000007 (7)
Disposition	Open if exists, Create otherwise
Create Options	0x00000000 (0)
Contexts	
Request Durability	False

Buttons at the bottom right: OK and Cancel.

Scenario Search

This Scenario also demonstrates the Scenario Search field . Text typed into this field will cause that text to be searched for in the current Scenario. In the example below "tree" has been typed into the Scenario Search field. The first instance in Line 3 is highlighted and a count of 1/3 indicates that there are 3 instances of the text "tree" in this Scenario. Clicking the down arrow or up arrow buttons will highlight the appropriate entry of the text "tree".

The screenshot shows the same dialog box as above, but with the search term 'tree' entered in the search field. The result highlights the 'Tree Connect' step in the list and shows a '1/3' indicator next to it. The configuration details for this step are displayed on the right.

#	Protocol	Name
1	SMB2	Open SMB2 TCP Connection
2	SMB2	Negotiate
3	SMB2	Session Setup
4	SMB2	Tree Connect
5	SMB2	Create File
6	SWT	Begin Loop
7	SMB2	File Write
8	SWT	End Loop
9	SWT	Begin Loop
10	SMB2	File Read
11	SWT	End Loop
12	SMB2	File Close
13	SMB2	Query Info
14	SMB2	Tree Disconnect
15	SMB2	Logoff

Name	Value
Input	
Session Handle	Default
Credits Charged	0
Credits Requested	1
Share Name	
Input Format	String (full \\SERVER\\SHARE)
Share Name	\\172.16.244.1
Output	
Output Handle	4: SMB2ShareHandle
Status Code	
Response Handlers	
Completion Status	
Scenario Impact	

Buttons at the bottom right: OK and Cancel.

Load Profile (see [Test Creation](#) for Load Profile details)

Associated with each Scenario is a Load Profile that controls how much load is generated by the Scenario. The following Load Types of load metrics can be specified:

- Concurrent Scenarios.
- New Scenarios per second.
- Concurrent Actions.
- New Actions per second.
- Concurrent Connections.
- New Connections per second.
- Bandwidth (Kbps).
- Throughput (Kbps).
- New Scenarios per Interval.

Two parameters control the behavior of Scenarios, depending on which Load Type is specified:

Max Number of Scenarios: The maximum total number of Scenarios that will be allowed. Default == Unlimited.

Maximum Concurrent Scenarios: The maximum number of Concurrent Scenarios that will be allowed. Default == Unlimited.

Unchecking "Unlimited" when these fields are enabled allows the Tester to specify values instead of Unlimited.

You can also specify how quickly the peak load is reached by specifying the number of steps to take ramping up or down to the desired load. If no steps are specified, the Appliance will ramp the project linearly to meet get to the load specified over the specified Ramp Up time. For more details on Load Specification use see the [Test Creation](#).

Network Profile (see [Test Creation](#) for Network Profile details)

Associated with each Scenario is a Network Profile that specifies the characteristics of the network that the test will operate in. See the [Test Creation section](#) for details of defining Network Profiles.

User Parameters

User Parameters are variables that can be used by a Scenario to control the execution of each Scenario instance. For example, you can parameterize a file name such that every instance of a Scenario operates on a different file. File names can be pre-defined or they can be dynamically generated.

User Parameters are created using a spreadsheet-like template composed of rows and columns. A User Parameter file can be created within the Load DynamiX TDE in which case the spreadsheet initially has eight columns and one row. User Parameter files can also be imported using .CSV files in which case, both the number of columns and rows is determined by the input CSV file. User Parameter files can be expanded, shortened and updated within the Load DynamiX TDE. User Parameter file columns contain similar kinds of information such as file names or user names or passwords. Scenarios cycle through User Parameter file columns one element at a time, picking up a new instance of the data from a column whenever a new instance of a Scenario is executed. If, during the execution of a Scenario, the end of a column is reached, the Scenario starts using data from the top of the column again in a circular list fashion.

See [Advanced Concepts: User Parameters](#) for more details on User Parameters.

Data File System

Files can be optionally assigned to a Scenario for writing to or reading from a Load DynamiX Server or DUT. The types of files can be used:

- Random – Creates a file with 64 bit integer values that are random whenever it is accessed.
- Sequential – Creates a file with 64 bit integer values that are sequential (e.g. 1,2,3,4,...).
- Physical – Creates a file using the contents of a real file.
- Seeded Random (seed) - Creates a file with 64 bit integer values that are random but are repeated whenever the file is accessed in a Scenario.

In addition to specifying file types, the Tester can also vary file properties such as name, size and attributes (e.g., Read-Only, Hidden, System, Directory, Archive, Device etc.).

Load DynamiX Appliance Ports when acting as "Server" (e.g. SMB, SMB2, NFSv3, or HTTP Server) have 1GB of ram memory to be used as virtual disk space. When files are created and written to in this virtual disk space by a Project, the files will consume the virtual disk space unless deleted during the Project's execution. Thus,

- Files created by one Scenario can be accessed by all future instances of any Scenario during that Project's execution time.
- Files are not shared across Server Ports. For example, files created on Server Port X cannot be accessed by Clients examining a Server on Port Y.
- All Server files are erased when the Server Scenario terminates at the end of the Project's execution time.

See [Advanced Concepts: Data File Systems and Data Verification](#) for more details on Data File Systems.

Test Construction

The Test Development Environment makes the process for constructing a test straight-forward:

- Create a Project (start from scratch or copy an existing Project).
 - From the main toolbar select File>New Project or open a Project from the Project Library.
- Create the Logical Ports and Network Profiles that will support the test run.
 - Click the Add Logical Port from the Timeline editor toolbar.
- Drag a Network Profile onto the Timeline from the Resource Library.
 - Create the Scenario(s) for the test run by selecting the Actions from the Toolbox that make up the test logic.
- Double click Actions in the Toolbox to add them to a Client or Server Scenario.
 - Drag a Load Profile from the Resources Library onto the Timeline.
- Construct the Timeline for the test run.
 - Click the Timeline Duration set the desired Project execution length.
- Run the test and analyze the results.
 - Click the Start Project button on the main toolbar.

It is recommended that a new Tester build their first Project using one of the many Sample Projects provided with the TDE.

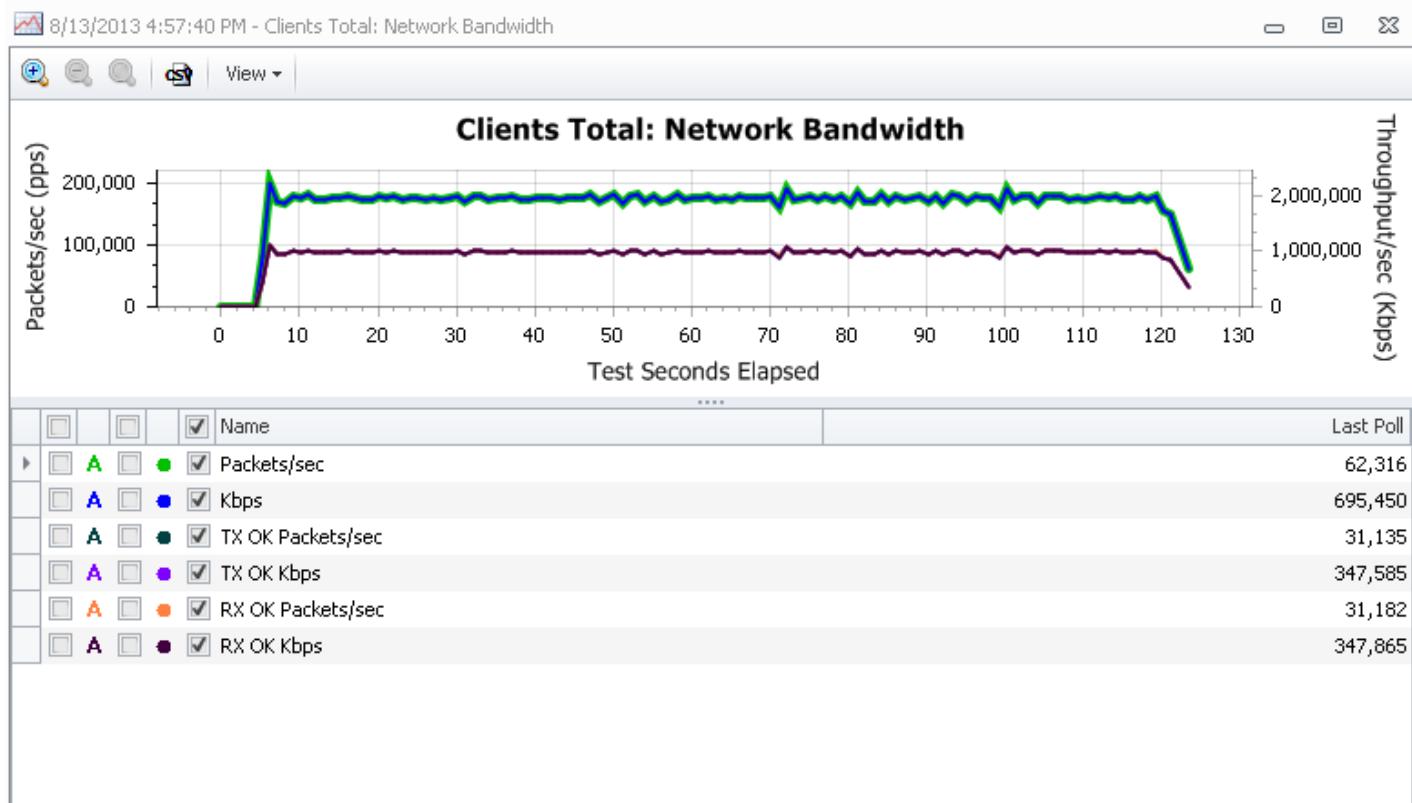
Test Analysis and Reporting (see [Executing Tests and Assessing Results](#) for Results details)

The Load DynamiX Test Development Environment provides graphical displays of network and protocol statistics to assist users in analyzing the results of a test run. Graphs can be viewed in their

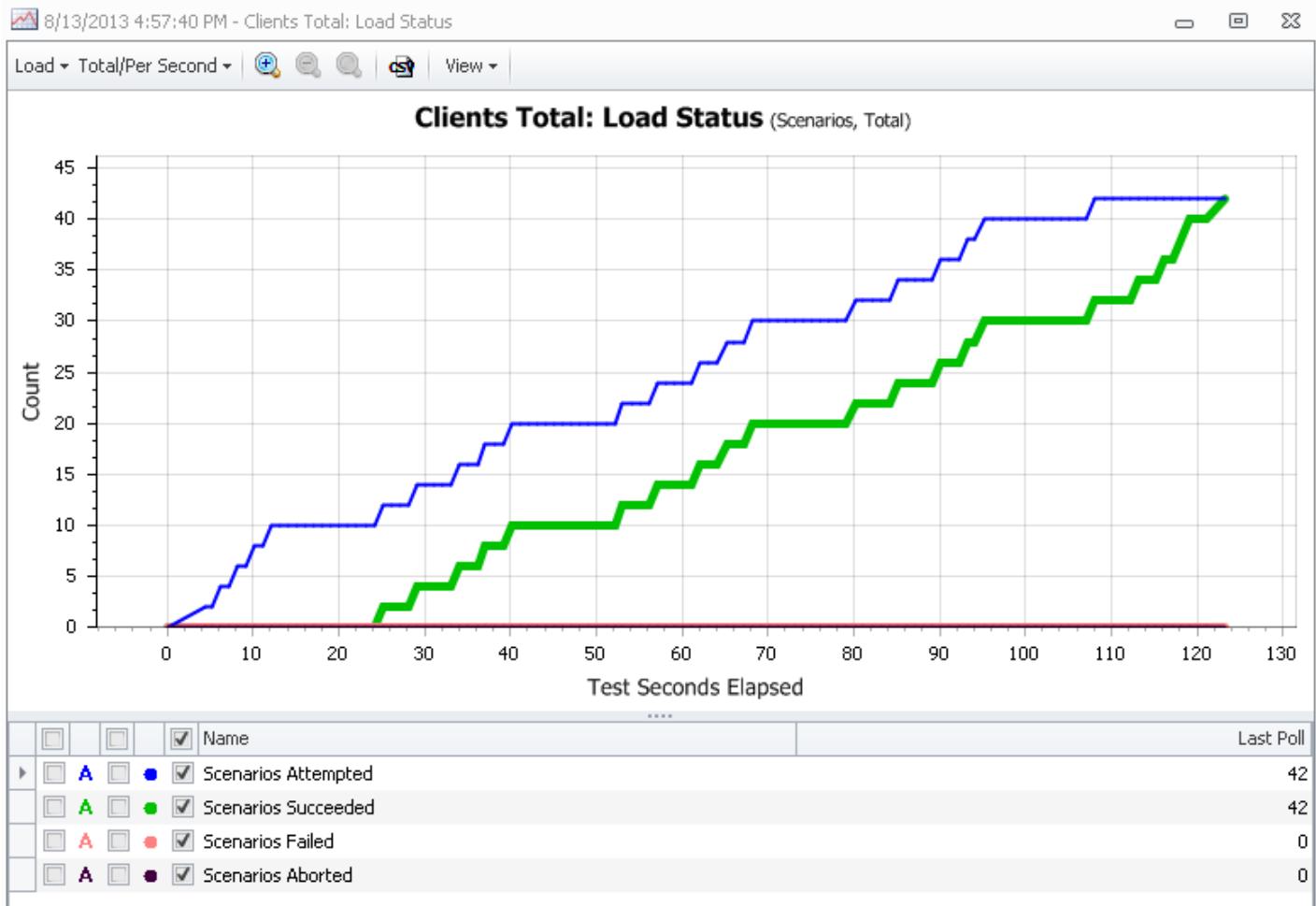
final state once the Project has completed its run or they can be viewed real-time while the Project is executing. Graph elements (bars, lines, etc) and counters are updated dynamically. Test results can be exported to a CSV file for further analysis once the Project has completed execution. The Load DynamiX product also generates text log files and can generate PCAP packet capture output during the test run.

The following are examples of the graphical data generated by a test run.

- Client and Server data rate (packets/second) statistics



- Scenario summary (total Scenario attempts and number successful, failed, and aborted)



Graph Legend and Counters

In all Results folder graphs, the region below the graph is dedicated to the legend for the graph and counter values. In the graph below, the bottom portion of the graph is dedicated the Legend and the Counters. The left portion is the Legend and the right portion are the Counters. The values in the Counters are updated dynamically when the Project is executing. The graph itself displays the items that are selected in the Legend box (Attempts and/or Succeeds and/or Fails and/or Aborts).

Note that most graphs can display more than one view of the data being collected. For example, in the graph below, it is possible to graph Scenarios (shown), Actions or Connections (using the Load drop down menu) and either Total (shown) or Per Second using the Total/Per Second drop down menu. In the case below, the number of Scenarios executing is being graphed over the period of time that the Project executes. The **Blue** line is Attempted. The **Green** line is Succeeded and the **Red** line is Failed. In the top box, selecting (check mark) Attempted, Succeeded, Failed or Aborted causes them to be

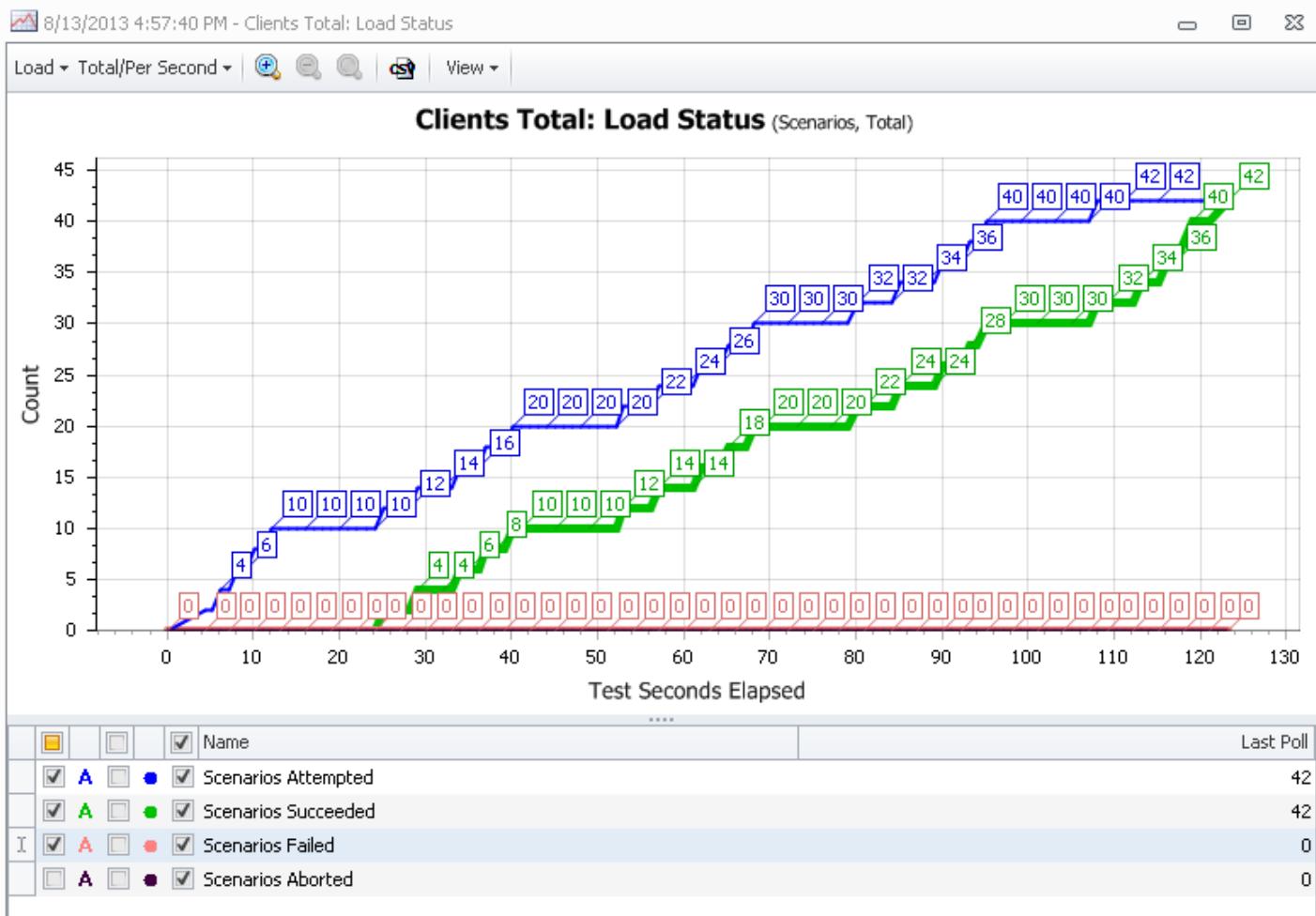


graphed. Checking the As causes the numeric values to be shown on the graph. Checking the



DOTs causes the data sample points to be shown. Dots help clarify data when graph lines overlap.

All lines have values shown at various times during execution if the As are selected. The counter boxes show the final values for the data items being measured.



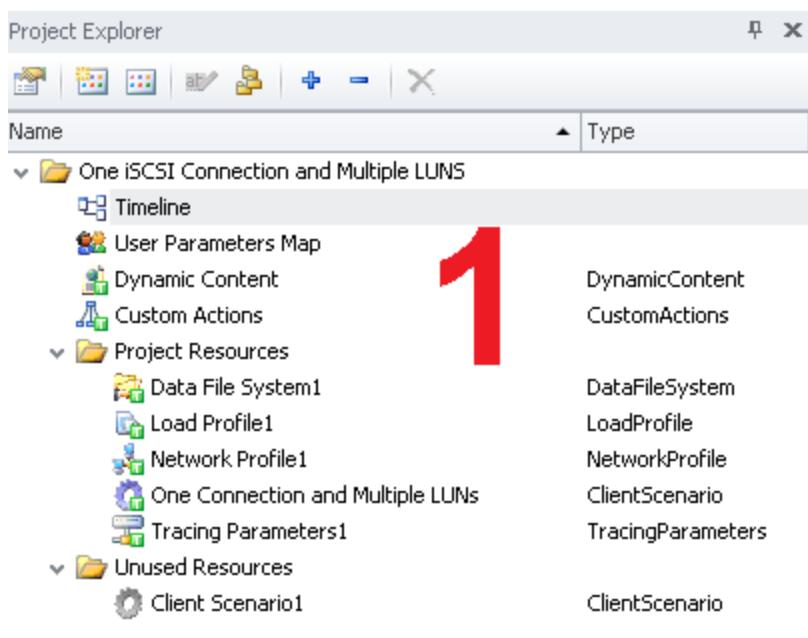
Using the Load DynamiX TDE Graphical User Interface

This section provides information on how to use the Load DynamiX Test Development Environment (TDE) Graphical User Interface (GUI).

Home Page

By default, the Load DynamiX Test Development Environment home page has 6 regions displayed.

Region number one is the Project Explorer window. The Resources in the currently open Project are displayed here. Resources not currently in use in the Project will be in the Unused Resources folder in this window.

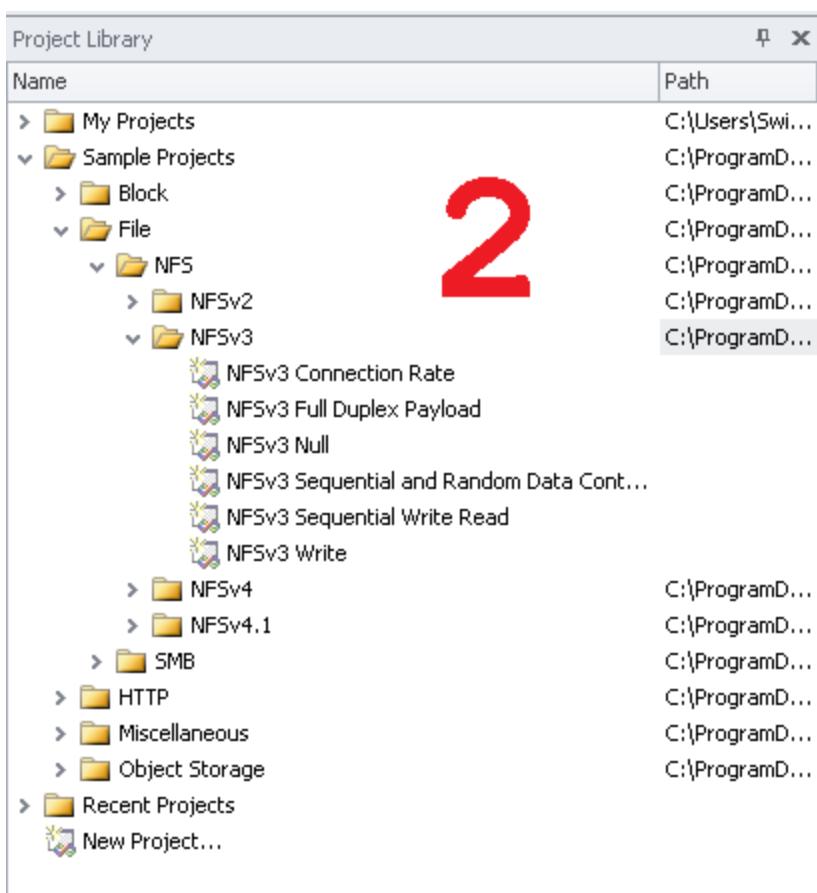


Associated with Region number one is the New Resource Creation Toolbar



which allows the Tester to create new TDE Resources such as Load Profiles, Client Scenarios, Tracing Parameters, User Parameter Files, etc and add them to the currently open Project in the Unused Resources folder.

Region number two contains the Resources Library and Project Library. The Resources Library contains resource templates that can be added to Projects as needed. It also contains a folder where resources to be shared among Projects are saved (My Resources). The Project Library contains the My Projects folder which holds user created or modified Projects and the Sample Projects folder which is a set of sample tests (see [Appendix: Load DynamiX Sample Projects](#)). The Sample Projects folder comes with tests segmented into separate sub-folders by protocol category (Block [block-oriented protocol samples], File [file-oriented protocol samples], HTTP samples, Miscellaneous [high performance 10GbE samples, specific functionality samples and IPv6 samples] and HTTP Storage [CDMI, OpenStack and Amazon S3 samples]). The Block and File Folders are sub-divided into SMB, NFS, Fibre Channel, and iSCSI sub-folders. The screenshot below shows the contents of the NFSv3 sub-folder within the File-oriented protocols folder.



Region number three displays the current Project's Compile/Execution Output, or the Project Summary.

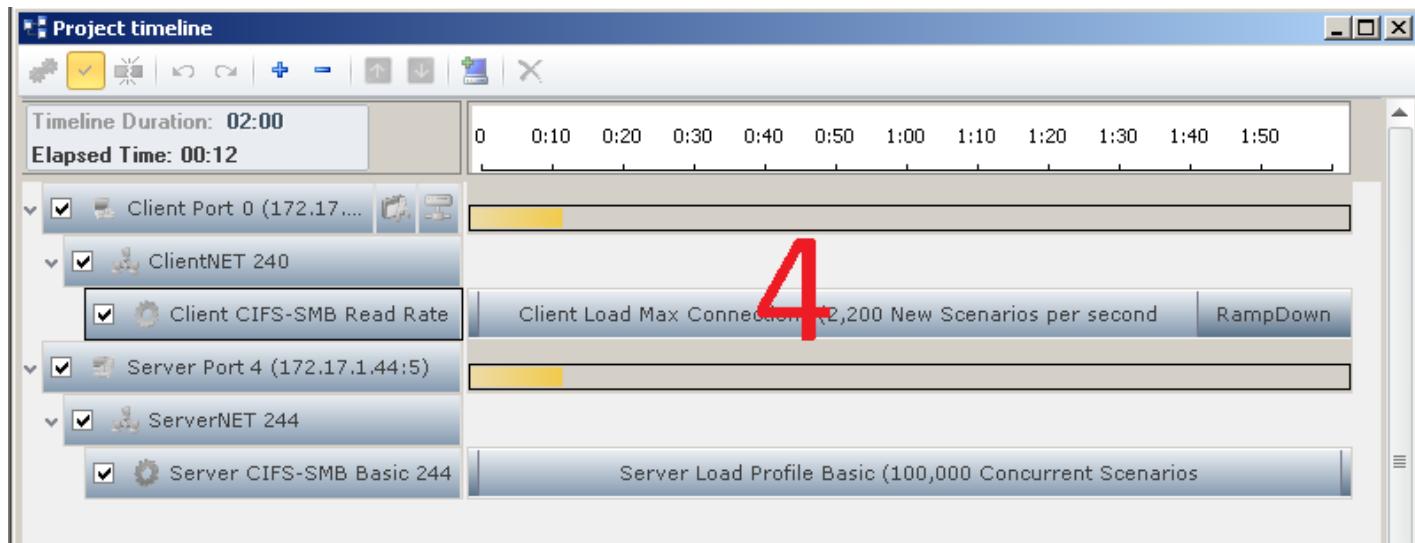
```

Output Client Port 0(172.17.1.49 port 3).log Server Port 4(172.17.1.49 port 7).log
uploading data content...
start test
starting test at ports: 172.17.1.49:7,3...

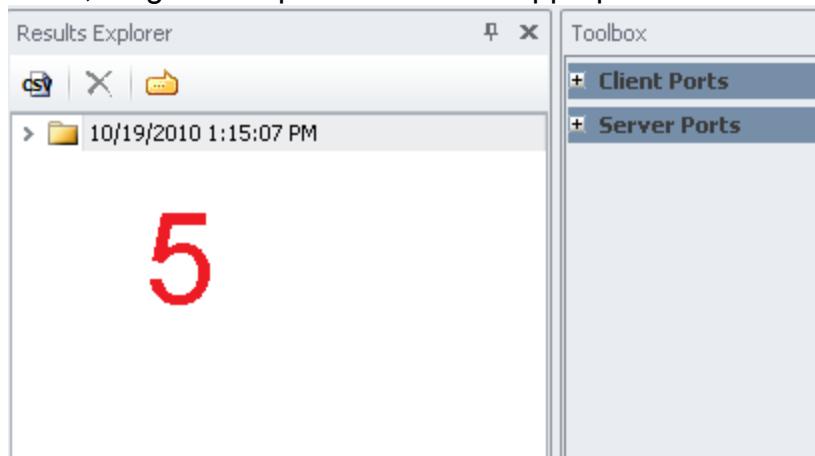
waiting while ports are starting...
  172.17.1.49:7 - Running
  172.17.1.49:3 - Running
Server Port 4(172.17.1.49:7) Statistics updated - Port time: 0ms
Client Port 0(172.17.1.49:3) Statistics updated - Port time: 0ms

```

Region number four is the main Timeline workspace. This is where Project files will open for editing and viewing.



Region number five opens when the Toolbox or the Results Explorer buttons are clicked. The content in the Toolbox depends on the currently active window. When the Timeline window is open, the Toolbox contains the available client and server Logical Ports. When a Client or Server Scenario is open, the Toolbox contains the protocol-specific Actions and Load DynamiX Scenario Control Actions that Testers use to create client and server Scenarios. To use any of the Toolbox items, drag and drop them wherever appropriate.



The Results Explorer button contains the Results folders. Each test execution produces a dated Results folder which contains results files from that execution of the Project.

Region number six is the Main Toolbar and Menus.



Some general GUI behaviors to be aware of:

- Dragging your mouse across an icon (mouse-over) displays the ToolTip text for that icon.
- Dragging any window within the program will bring up a docking assistant. Hold the mouse over one of the icons to see a preview of the new layout. Drop the window on the icon to set it to that layout.
- A button/icon is only displayed if it is active for an operation – otherwise, it is grayed out (and not selectable).
- Right-clicking GUI object displays the available operations that can be performed on that object.

- Default values are presented for all Action inputs.
- Drop down menus for many Action inputs will provide choices for the input value, often based on previously entered information (e.g., file names, locations, IP addresses, etc.).

Common GUI Operations

Various Toolbars in the Load DynamiX TDE GUI contain icons representing operations that have common functionality:



Undo the previous operation (Ctrl+Y)



Redo the last operation (CTRL+Shift+Z)



Delete the selected item(s)



Move selected item(s) Up



Move the selected item(s) Down



Enable or Disable the selected item(s)



Copy item to clipboard



Paste item from clipboard



Expand/Collapse elements



Add item (examples: add Client or Server Logical Port or add Appliance IP address)

Main Toolbar

The following options are available from the main GUI toolbar (see below for what can be done from each drop down menu):

- File
- Edit
- View
- Project
- Window
- Help

File Dropdown Menu

The File drop-down menu allows you to perform basic Windows-like file operations:

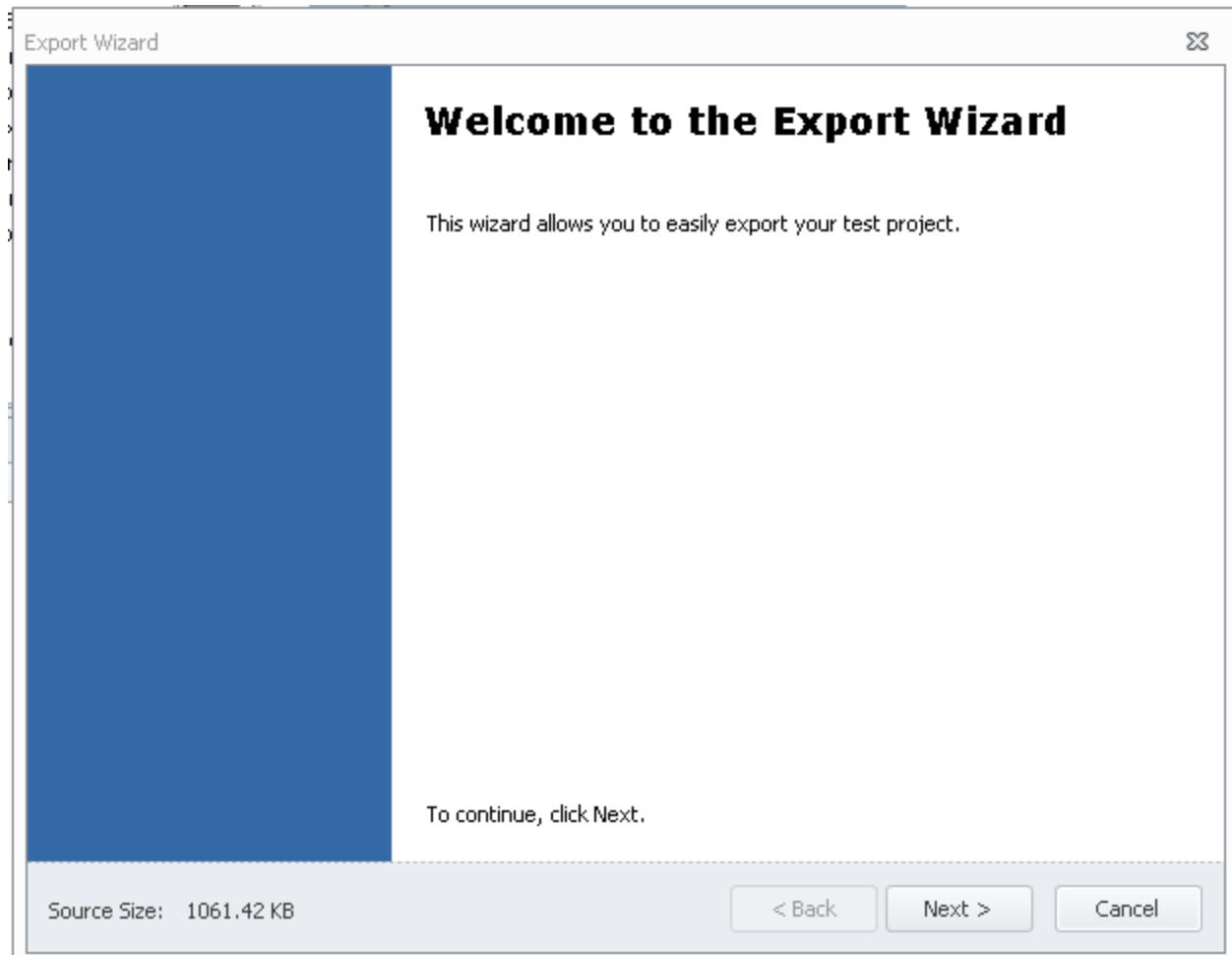
Operation	Description
New File	Create a new Resource or Project file
New Project	Create a new Project - empty or from a template
Open	Open a previously saved file Resource or Project file
Open Project	Open a previously created Project
Close	Close the currently selected file (Resource or Project file)
Close Project	Close the currently open Project

Save	Save the currently open Resource
Save As	Save the currently open Resource with a new name
Save Project As	Save the currently open project with a new name
Save All	Save all Resources in the currently open Project
Export Project	Save the Project files in a .zip file
Import Project	Extract the contents of a .zip file into a Project folder
Recent Projects	The ten most recently opened Projects (does not display if no recent Projects)
Exit	Exit the TDE

Import Project / Export Project

It may be necessary to "package" the files that make up a Load DynamiX Project so that the Project can be sent (email or FTP) to others. The Export Project function in the File Menu creates a ZIP file containing user-specified contents. The Import Project function reads an Project .zip file created by the Export Project function and creates a Project folder in a user-specified location.

In the File drop down menu, click Export Project



Click **Next >** to specify Resource and Data files that are to be included in the Export.

Export Wizard

**Data Content Files**

Choose which data content files should be exported.

Include data content files in the export?

- All Files (files with an absolute path will be added to the Data folder)
 Files in project Data folder only
 No

Which sets of files will be included in the export?

- All
 Only those used in timeline
 Include Project Log
 Include Global Log

Source Size: 1061.42 KB

< Back

Next >

Cancel

If any Data File Systems are present in the Project, "Include data content in the export?" question will be present, otherwise it will be grayed out. Make selections to include the Data File Systems (or not) and whether to include All Resource file (default) or just those in the TimeLine. Also, Export will save the Project and Global log files which may be of use in analyzing TDE or Project behavior.

Click [Next >](#) to select the Results folders that are to be included.

Export Wizard

**Statistics results**

Choose which results folders should be exported.

Choose results to include:

- None
- All
- Last three items
- Specified:

 3_17_2014 2-31-52 PM

Source Size: 1061.42 KB

< Back

Next >

Cancel

The default is to include None of the Results folders but if it is desirable to include the Results Folders, select All, Last three items or Specified to pick the Results Folders to be included. In the example above, All was selected to pick the single Results Folder present. Click [Next >](#) to select the Backup folders to include.

Export Wizard

X

Backups

Choose which backup files should be exported.

Choose backups to include:

- None
- All
- Last three items
- Specified:

Source Size: 1061.42 KB

< Back

Next >

Cancel

If the Backups of prior versions of the Project are desired and present, select All, Last three items or Specified to include them in the Export. Click  to specify the location and file name for the Export .zip file.

Export Wizard

**Destination**

Choose destination file name for your export.

File Name:

C:\Users\SwiftTest\Documents\SwiftTest\My Projects\CIFS-SMB Full Duplex Echo.zip



Source Size: 1061.42 KB

< Back

Next >

Cancel

Notice the Export interface has been calculating the size of the input to the Export process. The resulting .zip file will be smaller than the source but if the resulting .zip file is larger than desired, consider eliminating Results Folders and Backups from the Export. Click [Next >](#) to create the .zip file and then click [Finish](#) to complete the Export.

The opposite of Exporting a Project is to Import a Project. In the File drop down menu, click Import Project

Import Wizard

×

Welcome to Import Wizard

The Import Wizard allows you to easily import your test project.

To continue, click Next.

< Back

Next >

Cancel

Click **Next >** to specify the location of the .zip file to be imported and the target folder.

Import Wizard

X

Import Project

Specify the location of the archived project and the destination where it should be extracted.

Project to Import (*.zip):

C:\Users\SwiftTest\Documents\SwiftTest\My Projects\CIFS-SMB Full Duplex Echo.zip

...

Destination Folder:

C:\Users\SwiftTest\Documents\SwiftTest\My Projects

...

 Project Exists, Overwrite? Project Exists, Rename Import?

Project Name: CIFS-SMB Full Duplex Echo

< Back

Next >

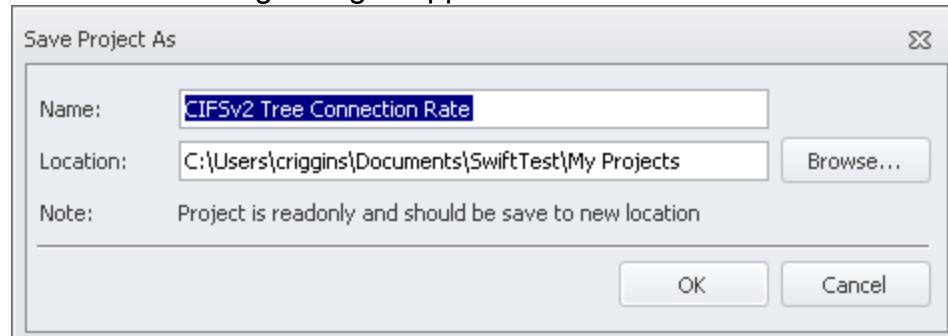
Cancel

The  button to the right of the Project to Import line allows the user to browse for the .zip file to be imported. The Destination Folder line allows the user to specify where to insert the Imported Project folder and files. This is by default the location of the My Projects folder. If a project by this name already exists in the Destination Folder, use the New Project Name field to specify a new Project name. Click  to Import the Project files and finish the process.

The .zip file selected for Project Import must have been created by the Project Export function.

Loading Projects into the TDE

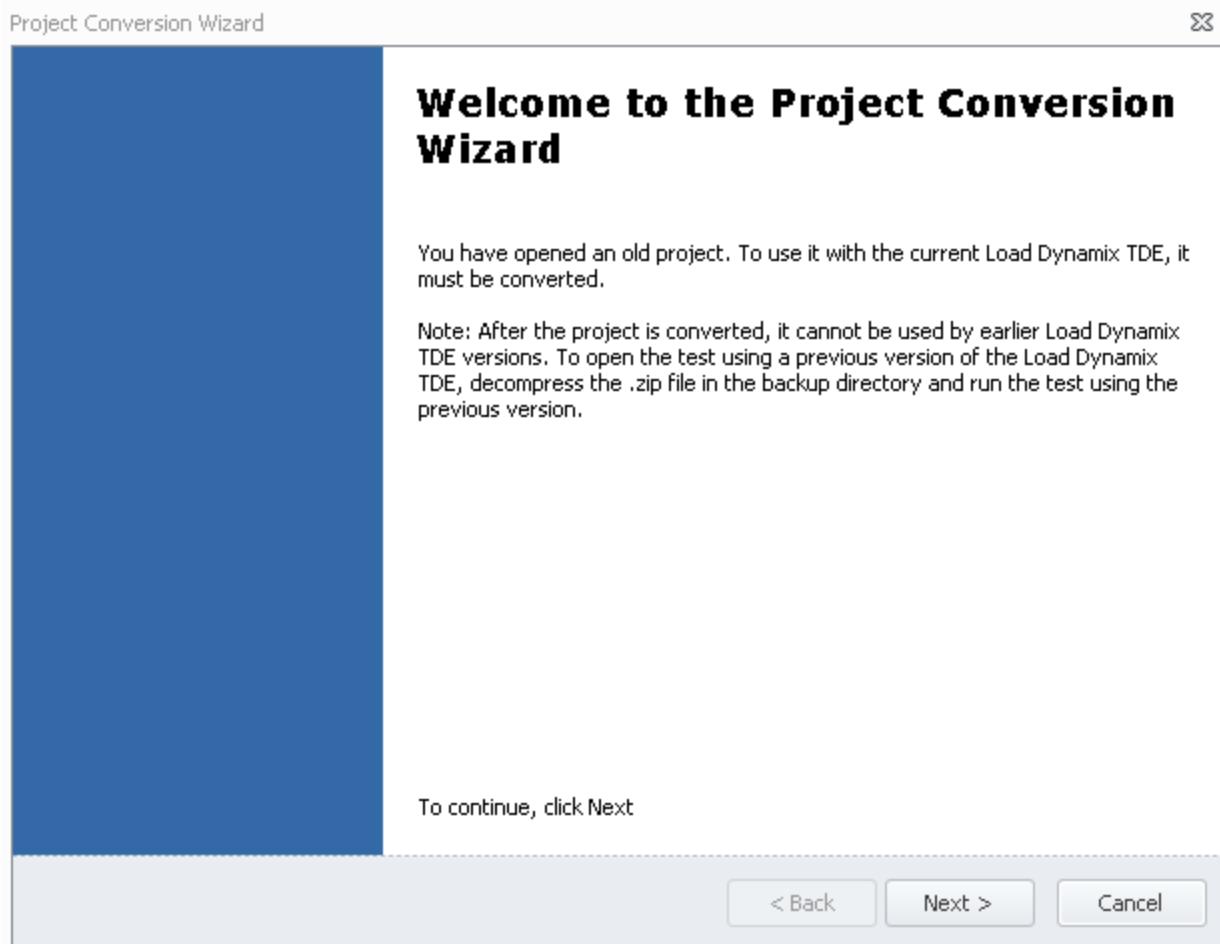
When a Project is opened (by double clicking a name in the Project Library or by using the Open Project operation), the project Timeline, Resources and Results are loaded into the appropriate regions of the GUI. If a Project is Read-Only (all Load DynamiX Sample Projects are shipped that way), a copy must be made of that project before it will be opened. Opening a Read-Only Project will cause the following dialog to appear:



Clicking OK will save a copy of the Project in the My Projects folder that is no longer Read-Only and this Project will show up in the My Projects list in the Project Library.

Project Conversion Process

If a Project being opened was created by a prior release of the Load Dynamix TDE, it must be converted to be compatible with the current release. This process will be automatically invoked when a down-rev project is opened. **A converted Project is only usable in the release it was converted for, so Projects that are only to be run on older release versions should not be converted.** The conversion process does make a Backup copy of the Project so it is possible to return to a prior version by extracting that version from the ..zip file stored in the Project's Backup folder. The Welcome screen gives the user the ability to select to proceed with or to the conversion.



The next step in the process is optional. The Wizard will ask where to store the Backup information (by default it is in the Project's Backup folder). Choosing not to make a Backup will save disc space but precludes the user from being able to go back to a previous version.

Project Conversion Wizard

**Project Backup**

Choose project backup folder and file name



Destination Path:

C:\Users\SwiftTest\Documents\SwiftTest\My Projects\R36 Samples\CIFS-SMB Read Compound Action



The Backup file is the Project's contents stored in a .zip file in the designated directory. To recover the pre-converted Project, simply extract the contents of the Backup file into a new folder in My Projects (or wherever Projects are stored). Once is selected, the conversion process begins. A successful conversion process looks like with the Output window below showing the conversion steps (which versions converted) for each resource:

Update Progress

File backup and conversion progress

Name	Status	Detail
Backup	OK	...
Client CIFS-SMB Read Rate	OK	...
Server CIFS-SMB Basic 244	OK	...

< Back

Next >

Cancel

Output

Output Error List

```
Project backup "C:\Users\SwiftTest\Documents\SwiftTest\My Projects\R36 Samples\Conversion Test"
Client CIFS-SMB Read Rate.client_scenario:
Converted to version 22.7 successfully
Converted to version 22.8 successfully
Server CIFS-SMB Basic 244.server_scenario:
Converted to version 22.7 successfully
Converted to version 22.8 successfully
```

Clicking  will bring up a progress bar briefly and then the final screen will appear with the Output window showing the full set of files that were converted.

Completing the Conversion Wizard

Project successfully converted

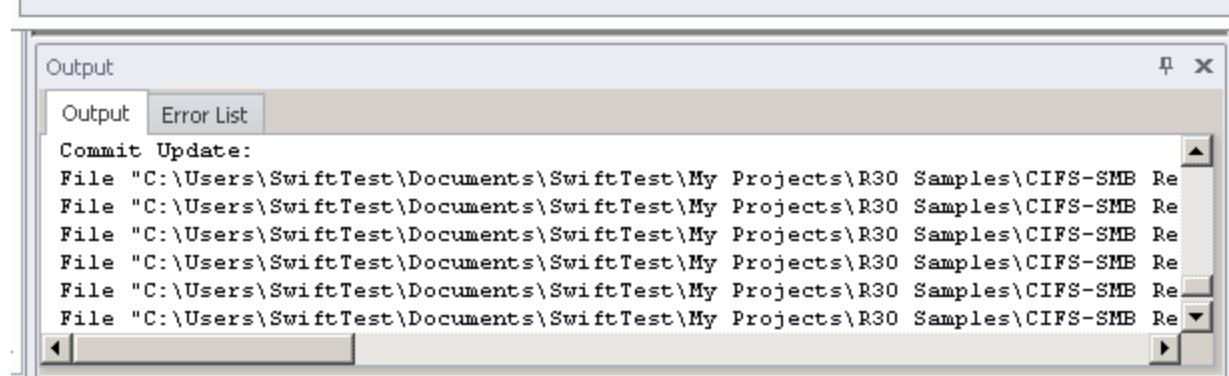
Click [here](#) to view the log file

To close this wizard, click Finish

< Back

Finish

Cancel



To see the complete details of the conversion process, click on the word [here](#) in this screen to see the conversion log file.

```

Backup:
File "C:\Users\criggins\Documents\SwiftTest\My Projects\NFSv4 Sequential write Read\Client NFS"
File "C:\Users\criggins\Documents\SwiftTest\My Projects\NFSv4 Sequential write Read\Files.user"
File "C:\Users\criggins\Documents\SwiftTest\My Projects\NFSv4 Sequential write Read\client Load"
File "C:\Users\criggins\Documents\SwiftTest\My Projects\NFSv4 Sequential write Read\clientNET"
File "C:\Users\criggins\Documents\SwiftTest\My Projects\NFSv4 Sequential write Read\Tracing Pair"
Project backup "C:\Users\criggins\Documents\SwiftTest\My Projects\NFSv4 Sequential write Read\NFSv4 Sequential write Read.swift_test:
Converted to version 8.0 successfully

Client NFSv4 Sequential write Read.client_scenario:
Converted to version 20.0 successfully
Converted to version 21.0 successfully
Converted to version 22.0 successfully

ClientNET 240.network:
Converted to version 3.8 successfully

Commit Update:
File "C:\Users\criggins\Documents\SwiftTest\My Projects\NFSv4 Sequential write Read\Update\cli...
File "C:\Users\criggins\Documents\SwiftTest\My Projects\NFSv4 Sequential write Read\Update\NFS...
File "C:\Users\criggins\Documents\SwiftTest\My Projects\NFSv4 Sequential write Read\Update\NFS...

```

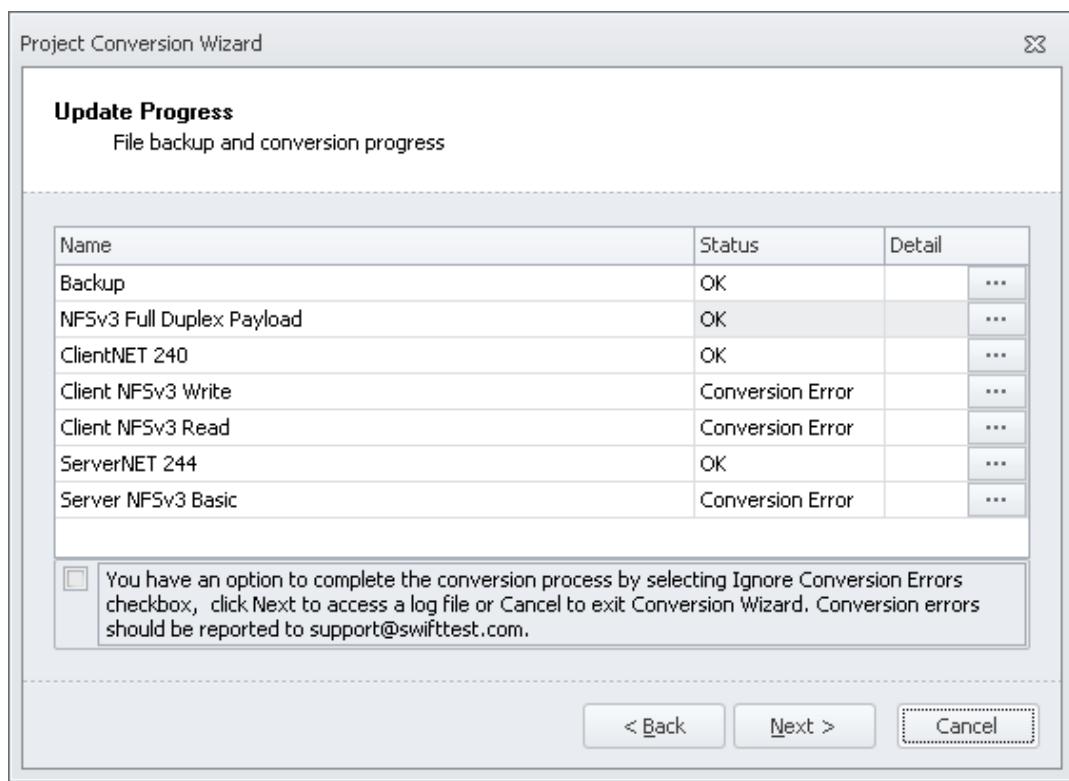
Close this log file and return to the Wizard Complete screen and click to complete the conversion process. The converted project will be opened in the GUI.

Bulk Project Conversion

If there are many Projects to convert, it may be more efficient to use the LdxCmd /upgrade option to convert Projects from the command line or in a script. See [Appendix: Test Automaton](#) for LdxCmd /upgrade command syntax.

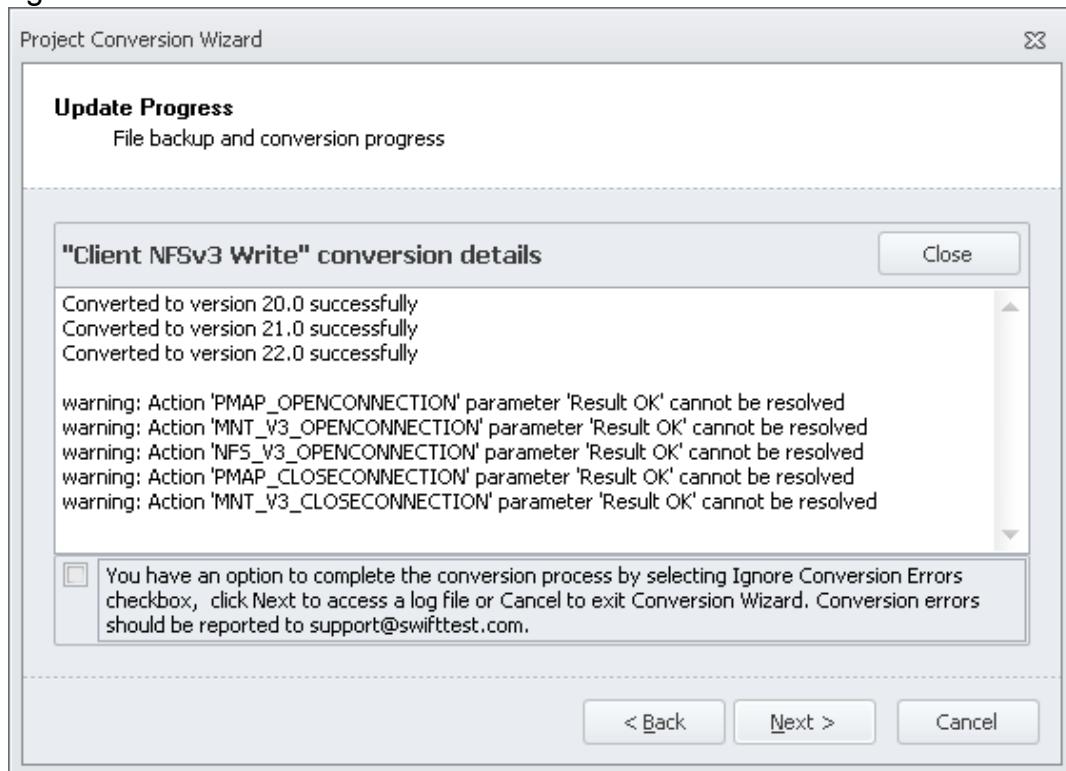
Project Conversion Errors

If there was a problem with any of the Project files that are being converted, the upgrade Progress screen will indicate that there were issues and give the user the ability to exit the conversion process or continue.



Click **Cancel** to exit the process or **Next >** to see the conversion log. If it is desired to proceed with the conversion, then check the box in the lower portion of the display and click **Next >**. Conversion errors should be reported to Load Dynamix support (Support@LoadDynamix.com)

To see the details of a specific set of the conversion errors, click the **...** in the Detail column to the right of the word Error.



Once there is a Project loaded into the TDE, the rest of the TDE's features and capabilities become operable.

Edit Dropdown Menu

The Edit drop-down menu allows you to perform basic Windows-like edit operations:

Undo  (Ctrl+Y),

Redo  (Ctrl+Shift+Z),

Cut  (Ctrl+X),

Copy  (Ctrl+C),

Paste  (Ctrl+V),

Find (Ctrl+F),

Replace (Ctrl+H),

Select All,

Clear,

Delete .

View Dropdown Menu

The View drop-down menu gives access to five of the TDE regions described earlier in this section:

Project Explorer,

Results Explorer,

Project Library,

Results Library

Toolbox

Output

Project Summary

Toolbars (enable/disable toolbars in the Main Toolbar)

Options

The Options window contains various TDE options/preferences which can be viewed and (in some cases) configured:

Environment Tab

NAME	VALUE	CONFIGURABLE?
Projects location	File path to the My Projects folder	NO
Resources location	File path to the Load DynamiX Resources folder	NO
My Resources location	File path to the My Resources folder	NO
Results location	File path to the Results folder	YES
Reload last project at startup	Checkbox to enable/disable	YES
Switch windows layout when test is running	Checkbox to enable/disable	YES
Open timeline when project is loaded	Checkbox to enable/disable	YES
Clear User Settings on Exit	Click to clear settings on Exit	NO
Export	Click to export TDE layout and user settings	NO
Import	Click to import TDE layout and user settings	NO
Default	Click to restore default TDE layout and user settings	NO

Notes:

- Whether to switch the windows layout while a test is running (default: no) – this option allows you to change how the windows display during a test run (e.g., you may want to limit which windows are displayed or where they are displayed)

Messages Tab

NAME	VALUE	CONFIGURABLE?
Show message when Project is read only	Checkbox to enable/disable	YES
Show message when Compilation fails	Checkbox to enable/disable	YES
Show message when deleting an Action	Checkbox to enable/disable	YES

Project Options Tab

NAME	VALUE	CONFIGURABLE?
Store data in Absolute Path or Relative Path	Radio button to enable one of two choices	YES
Automatically convert/save statistics to CSV	Checkbox to enable/disable	YES
Enable Advanced Load Profile	Checkbox to enable/disable	YES
Use absolute time in statistics charts and CSV	Checkbox to enable/disable	YES

Notes:

- Absolute time is only available for the results retrieved from the appliance upgraded to firmware version 5.6 or higher. Absolute timestamps are provided by the appliance and adjusted to current timezone. Absolute timestamps are displayed in statistics charts and in exported CSV files, replacing relative timestamps since start of the project run.

Toolbox Tab

NAME	VALUE	CONFIGURABLE?
Protocols	List of all supported Protocols and Checkbox to enable or disable presence in Toolbox	YES

Scenario Editor Tab

NAME	VALUE	CONFIGURABLE?
Invalid Action	Color combo box	YES, color is configurable
Parent Action	Checkbox to enable/disable and color combo box	YES, enable/disable feature, color is configurable
Child Action	Checkbox to enable/disable and color combo box	YES, enable/disable feature, color is configurable
Loop Action	Checkbox to enable/disable and color combo box	YES, enable/disable feature, color is configurable
Thread Action	Checkbox to enable/disable and color combo box	YES, enable/disable feature, color is configurable
Async Action	Checkbox to enable/disable and color combo box	YES, enable/disable feature, color is configurable
Show Actions Arrows	Checkbox to enable/disable	YES

Project Dropdown Menu

The Project drop-down menu is available when a Project is open in the Project Explorer window.

Option	Description
Add New Item	Opens the Add New Item dialog, add a new Project Resource
Import Existing Item	Opens a browser in the current Project folder, import a Resource from that Project or some other Project
Compile Project	Converts the Project for execution on the Appliance
Cancel	Cancels the compile process
Start Test	Executes the Project (F5)
Validate Test	When the Project is started via "Validate Test (CTRL + F5)" option in the menu or toolbar, the Project execution proceeds as usual, except that the duration and parameters of all included Load Profiles are disabled. Instead, each Scenario is executed once, and Project execution completes immediately after that.
Stop Test	Cancels the currently running Project (cannot be resumed – must be restarted from the beginning of the Project). Collected statistics, log files and PCAP data is downloaded from the Appliance. (Shift+F5)
Abort Test	Aborts the current test without downloading any log files or PCAP files.
Generate Automation File	Generate the XML files necessary to Automate the execution of this Project. See Appendix: Test Automation and LDX-E Integration for details
Import Automation File	Import (load) a Project into the TDE via its AutomationConfig.XML file.
LDX-E Server Login	Create a connection between the TDE and an LDX-E server by logging in. The server IP address and user credentials must be known. See Appendix: Test Automation and LDX-E Integration for details.
LDX-E Project Search	Once logged in, review the set of LDX-E Projects that are available to bring into the TDE. See Appendix: Test Automation and LDX-E Integration for details.
Sync Project from LDX-E	Once Logged in, Sync (import) the current TDE project from the LDX-E server if it has been saved to the LDX-E server. See Appendix: Test Automation and LDX-E Integration for details.
Save Project to LDX-E	Once Logged in, Save the current Project to the LDX-E server. See Appendix: Test Automation and LDX-E Integration for details.
Ports & Appliances	Opens the Ports & Appliances dialog from which you can view test port assignments and add, remove, ping an Appliance, get port information as well as update Appliance Firmware and reboot appliances

Window Dropdown Menu

The Window drop-down menu enables you to change how the various windows are displayed on the GUI.

Option	Description
Tabbed MDI	Organizes all open windows (resources, projects, results, etc) into a tabbed frame. Switch tabs by clicking the name of the window at the top of the frame or open a different file using the Resources or New File menus.
Cascade	Organizes all open windows diagonally across the screen from top-left to bottom-right.
Tile Horizontal	Organizes all open windows by stretching them to fit the screen horizontally.
Tile Vertical	Organizes all open windows by stretching them to fit the screen vertically.
Close All	Closes all open windows .
Restore Default Window Layout	Loads the default window layout (described as the Home Page at the beginning of this section) which is optimized for test development

Help Dropdown Menu

The Help dropdown menu provides access to this help file via the Contents selection and, via the About selection, the Version and Build of the TDE.

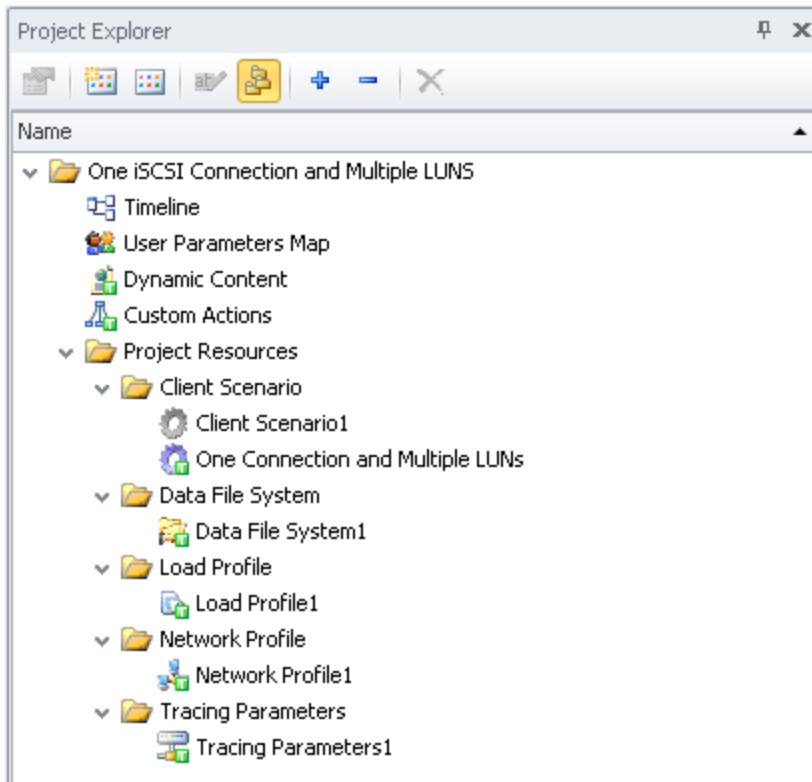
Option	Description
Help	Open online Help
Help (PDF)	Open the PDF version of online Help
3108/5102/5108T/5108S Quick Start Guide (PDF)	Open the Load DynamiX 1G Series 3108 and 10G Series 5102 5108T 5108S Quick Start Guide PDF file
6202/6204/6208 Quick Start Guide (PDF)	Open the Load DynamiX FC Series 6202E 6202 6204 6208 and Unified Series U1022 U1044 Quick Start Guide PDF file
Virtual Appliance Quick Start Guide (PDF)	Open the Load DynamiX Virtual Appliance Quick Start Guide PDF file
More Quick Start Guides...	Open the folder that contains all Load DynamiX end user documentation
About...	Get Version and Build information for this TDE

Project Explorer

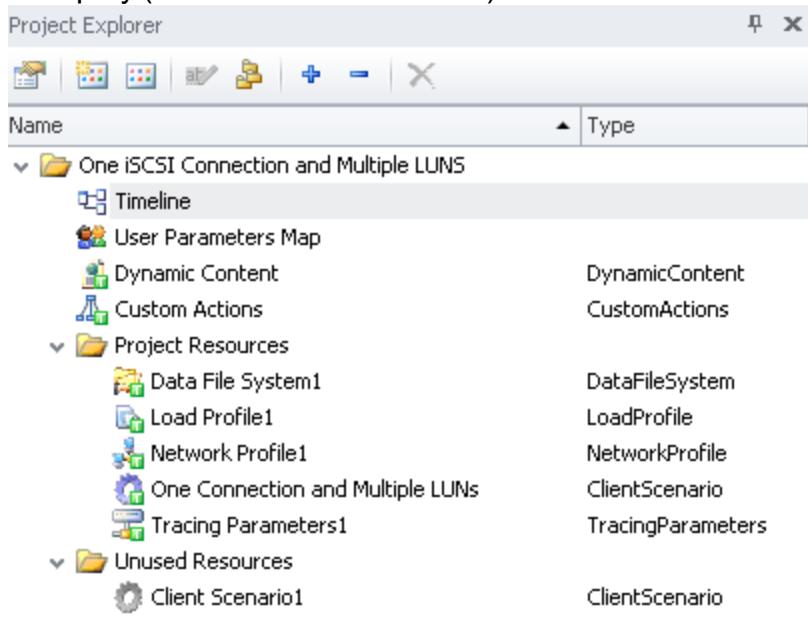
The Project Explorer window operates on the current Project. The current Project appears in the Project Explorer window when when a new Project is created, an existing Project is opened, or a copy of the current Project is created. Toolbar options available from the window are:

Button	Description
Open/Properties	Opens the Settings window for the currently selected Project Resource
Add New Item	Opens the Add New Item dialog, from which you can add a new Project Resource (you can also add a new item by right-clicking on the Project Resources or Unused Resources folders in the Project Explore window for an active Project)
Import Existing Item	Opens a browser in the current Project folder so that you can import a Resource from that Project or some other Project
Rename	Rename the currently selected Project Resource
Delete	Deletes the selected Resource
Group by/UnGroup by	Shows Resources as a single collection or Grouped by type
Expand All / Collapse All	Expand or Collapse folders within the Project Explorer window

Group By (enabled) - allows the Resources to be organized by Resource type in the Project Explorer window. In grouped mode, note the distinction between the icons for used resources (full color and have a green square on the bottom right of the icon) and unused (no color at all). In un-grouped mode, unused Resources appear in their own folder.



Group by (disabled - default mode)



Resources can be added to a Project's Resources folder in the following ways:

Function	Options
Add New Item	Project Explorer Add New Item button, main toolbar Project > Add New Item, right click any Resource > Add New Item)
Import Existing Item	Project Explorer Import Existing Item button, main toolbar Project > Import Existing Item, right click any Resource > Import Existing Item
	Drag-and-drop a default item from the Resources Library window

To add a new item to the Project folder:

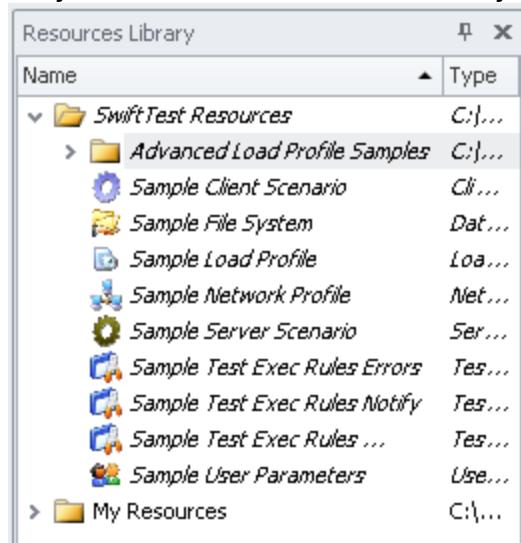
- Select Project > Add New Item or right click Resources and select New Item. The Add New Item dialog displays.
- Select the type of item to add.
- Enter the item name in the Name field. A default name displays in the field based on the item type and the number of items previously created.
- Click Add. The dialog for adding the item displays. The name also displays in the Resource folder of the Project Explorer window.
- Fill out the fields as needed.

To import an existing item to the Project folder

- Select Project > Import Existing Item or right click Resources and select Import Existing Item. A browser displays showing the most recently-accessed Project folder.
- Browse to the item you wish to import and select it.
- Click Open.

Resource Explorer

The Resources Explorer contains default Resource templates that can be dragged-and-dropped to a Project folder. Once added to a Project, the templates can be modified to suit test needs.



Add Resources from the Resources Explorer window

- Open the Resources Explorer window.
- Expand the folders (if they are not already expanded).
- Select the item and drag-and-drop it into the Resources folder in the Project Explorer window.

Add Resources to the Resources Explorer from an open Project

- Open the Resources Explorer window.
- Expand the folders (if they are not already expanded).
- In the Project Explorer window, select the resource and drag-and-drop it into the My Resources folder in the Resources Explorer window.

To use a Resource created in one Project in another, first open the Project that contains this resource then copy the Resource to be shared into the My Resources folder. Now open the Project that needs the Resource and then copy from My Resources into the Project.

Results Explorer

The Results Explorer contains a time-stamped folder for each test run. Each Results folder contains a set of files that show, graphically, the statistical outcome of the test run based on the test Actions contained in the Scenarios. Individual graphs can be exported to a CSV file for further analysis. Results folders also contain log files that contain test execution messages and statistics details. Results folders may be deleted when no longer needed. Results folders contain either Client and Server Statistics or just Client Statistics.

Server Statistics

The Server Statistics window provides a graphical view of data rate statistics for the Server Scenario of an executed Project. Standard graphical representations of Load Status, Network Status are available for each test run. Other graphs for commands, Actions, response times, etc are created based on the Actions executed in the Scenarios. Different views of the output can be obtained by selecting from the available tabs:

- Per Second – Displays count per second information
- Total – Displays overall count totals

Client Statistics

The Client Statistics window provides the same graphical capability as the Server Statistics window, but for a Client-view of the executed Project.

Custom Graph Templates

Statistics graph templates that allow the Tester to create custom graphs combining statistics from specific Project Results. See [Executing Tests and Assessing Results](#) for more details on Custom Graphs.

Project Summary

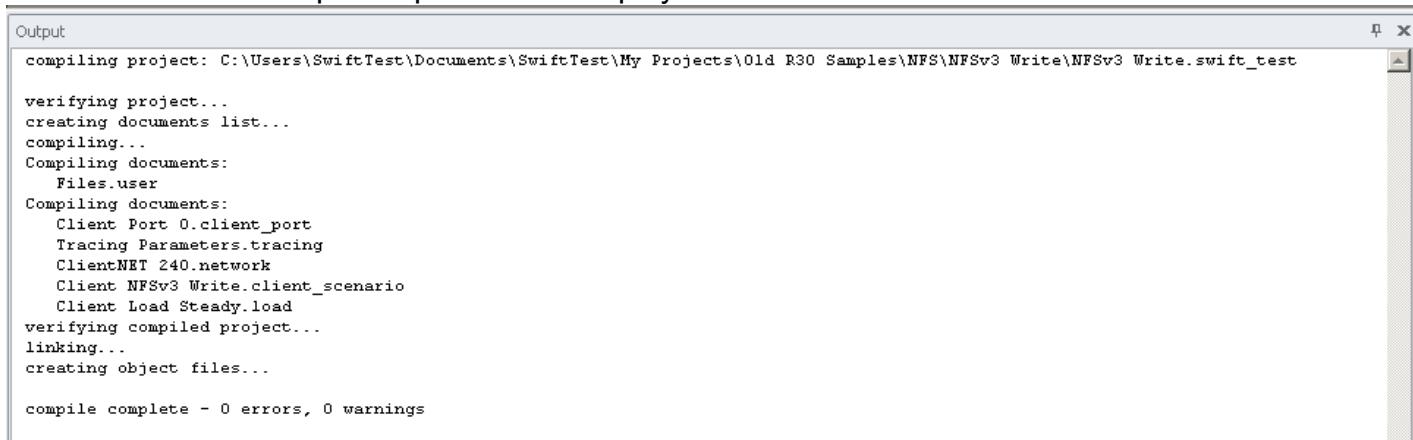
The Project Summary window shows Timeline and Load information for an executed Project:

- Timeline – Shows the test Resources that make up the Timeline
- Load – Shows the load on the port based on the metrics configured in the Load Profile

Output

The Output window displays the output when compiling and executing a Project. In general, Error and Warning messages seen in the Output window relate to either a compilation issue or a major issue with the Appliance software (e.g. a segmentation fault or lack of connectivity on a physical port, etc.). It does not report the results of specific Actions when applied to a file server or file service. The files contained in the Results folder are the primary sources for troubleshooting a Project and assessing how it is performing.

Shown below is a sample Output window display.



```
Output
compiling project: C:\Users\SwiftTest\Documents\SwiftTest\My Projects\Old R30 Samples\NFS\NFSv3 Write\NFSv3 Write.swift_test

verifying project...
creating documents list...
compiling...
Compiling documents:
  Files.user
Compiling documents:
  Client Port 0.client_port
  Tracing Parameters.tracing
  ClientNET 240.network
  Client NFSv3 Write.client_scenario
  Client Load Steady.load
verifying compiled project...
linking...
creating object files...

compile complete - 0 errors, 0 warnings
```

Toolbox

The Toolbox displays a set of predefined Actions that correspond to HTTP, CIFS-SMB, NFS and iSCSI protocol commands, Kerberos commands, and Load DynamiX Scenario Control Actions. You select from these Actions when creating Client and Server Scenarios. Actions only display if there is an open Client or Server Scenario. Detailed lists of supported commands can be found in the Reference sections. In an empty Project or a project with no Scenario open, the Client and Server Logical Ports display in the Toolbox.

Autofill

Certain kinds of parameters can be dynamically generated either at the time the Action is executed or when the parameter input is being defined. Tabular data such as User Parameter files support the AutoFill dialog. See [Advanced Concepts: User Parameters](#) for Autofill details.

Functions and Formula

Functions and Formula provide the ability to create input for Action input fields during the execution of a Test. Inputs may take the form of Strings (e.g. file name, IP address, Fibre Channel target, etc.) or Integers (e.g. Loop counter, Block Size, Offset, etc.). Functions generally produce Strings as output and their output is interpreted as a String in most cases. Some exceptions occur such as the output of a Function being used as a Loop counter. Action Input fields that support Functions will have the



Function Editor button enabled when focus is put on that field. Functions DO NOT support mathematical operations. A Function that has input "10 + 20" does not produce "30". It produces the string "10 + 20". Formula can be used to produce mathematical output.

Formula is an Action that computes a Tester-defined algorithm and produces a 64bit unsigned integer output. The output of the Formula Action can be added to input fields using the Function Editor button



For a complete discussion of Functions and Formula, see [Appendix: Functions and Formula](#). For a complete discussion of Variables, see [Advanced Concepts: Variables and Aliases](#).

Test Creation

Test Creation

This chapter provides information about how to create test Projects to verify networked storage devices and services using the Load DynamiX TDE. Topics include:

- The key components of a Load DynamiX Project
- Creating test Projects using the Load DynamiX Project Wizard or manually
- Authenticating access to servers in Load DynamiX Projects
- Reading and Writing Data in Load DynamiX Projects
- Controlling Project execution behavior using Test Execution Rules
- Tracing Project results using Tracing Parameters
- Using Text files to store Project notes

Key Components of a Load DynamiX Project

A Load DynamiX Project is composed of:

- A Timeline that determines how long the Project runs and its ramp up and down periods
- A Logical Port (or Ports) assigned to a Physical Port on the Appliance that are used to execute the Project
- One or more Network Profiles that define the network(s) that the Project uses
- One or more Load Profiles that define the load of the Client and/or Server
- One or more Scenarios that contain the Actions that are executed when the Project is run.
- Files that are used as input to the Project or configuration information (e.g. User Parameters, Tracing Parameters, Data File Systems, etc.)

For a list of the kinds of information typically required to design a Load DynamiX Project, see the Information Typically Required for Project Design section in the [Introduction chapter](#).

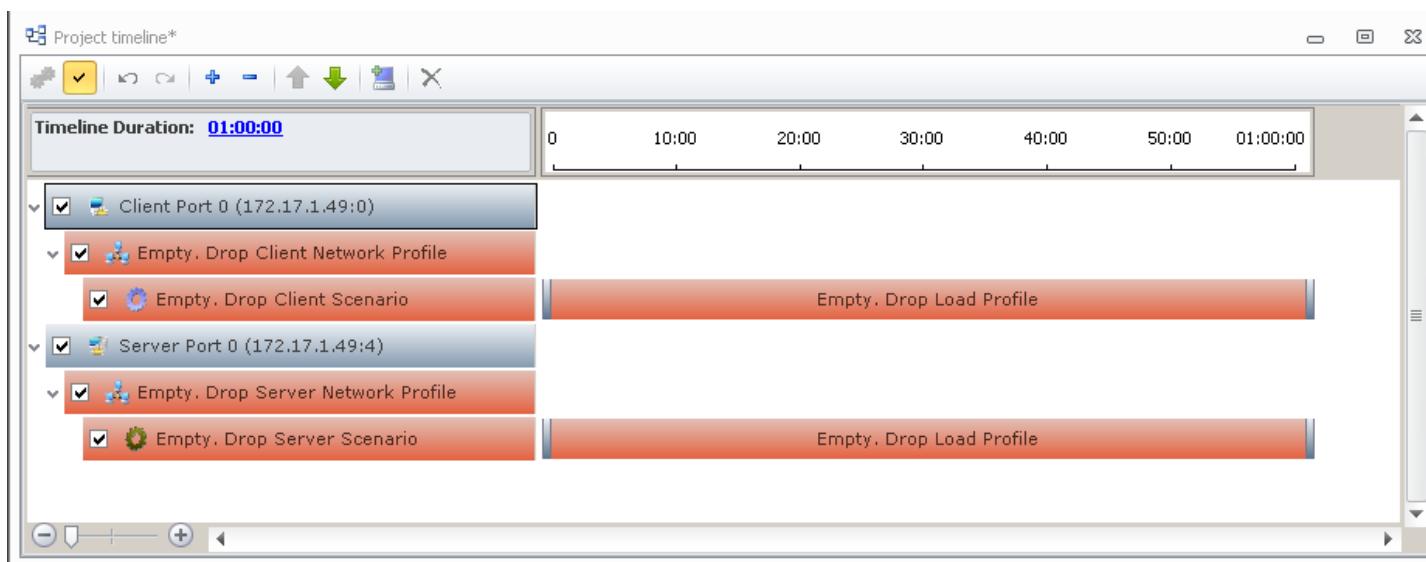
The Timeline

This section discusses how to construct the Timeline.

A Timeline requires a minimum set of Resources be present. For the Project to compile properly, the Timeline must contain a Logical Port (assigned to a Physical Port), a Network Profile, a Client and/or Server Scenario, and a Load Profile. The TDE requires that the test components be added in the order described below, and will not allow you to add components in any other order.

To construct a Project Timeline

- Starting with an empty Project, click on the Add Logical Port button  and select a Client Ports and the virtual Client Port desired. Repeat for the Server Logical Port. Remember that they must be linked to a Physical Port through the Ports & Appliances window before the Project can be executed. A specific Logical Port can only be included in a Timeline once (i.e., one Client Port 0, one Server Port 0, etc.). Your blank Timeline could look like this:



- Select a Network Profile from the Resources Library window and drag-and-drop it to a Logical Port in the Timeline window. More than one Network Profile can be associated with a Logical Port.
- Select a Client or Server Scenario from the Resources Library window and drag-and-drop it to a Network Profile in the Timeline window. If the Scenario was dragged from the Load DynamiX Resources folder within the Resources Library it will be empty. The empty Scenario must have the desired Actions added to it before it can be executed.

A Scenario template must be of the same type as the Logical Port – e.g. you cannot drop a Client Scenario template onto a Server Port. More than one Scenario can be associated with a Network Profile.

- Select a Load Profile from the Project Explorer window and drag-and-drop it into the Timeline window (in the area to the right of the Scenario).

Double-clicking on the Load Profile bar opens a menu from which you can change the duration and other attributes. You can also change the duration of the entire test by clicking the time field of the Timeline Duration: at the top left of the Timeline window. Changing the duration of the test will cause the load profiles to automatically fill the new time frame.

Once Scenario Actions are supplied, the Timeline Duration set and Physical Ports are assigned to Logical Ports, the Project is ready to compile and execute (see [Executing Tests and Assessing Results](#) for details).

Some additional information when working in the Timeline window:

- When you add a Load Profile to a Scenario it will automatically fill the given Duration and contain a RampUp and RampDown zone on either end of the timeline. This determines how quickly the designated load will be applied. You can adjust the time allowed for RampUp and RampDown by clicking on the edge of the zone and stretching it to the desired length, or you can double click the zone and set a specific value.
- Once a Resource has been added to the Timeline, you can double-click on the item to view or edit its properties.
- When setting up a Timeline that has both a Client and Server Scenario, the Duration input of the Load Profile of the Server Scenario must be sufficient to handle (must be greater than or equal to in length) the Duration input of the Load Profile assigned to the Client Scenario.
- Duration is automatically determined by the Load Profile parameters. You can modify the test

start and duration by:

- Moving your mouse to the start or end of the Load Profile graphical display and dragging to the right or left.
- Double-clicking a Load Profile and changing the Start time and Duration fields.
- Maximum Project/Test Duration is 1000 Hours.
- Dragging and dropping a network profile, Scenario, or load profile will copy it to the new location. Dragging while holding the "control" key will move the selected item to a new location.
- The Timeline Toolbar
 - Open Scenario - Opens the file that is linked to the selected component of the timeline.
 - Enable/Disable Resources - Displays and toggles the status of the selected component. This is also displayed as a check-box on each of the components. Enabling/Disabling affects anything attached to or below the selected component.
 - Create Duplicate Load Spec - Splits the selected load profile into two equal sections. Can be repeated as needed. Once split, new (i.e different) Load Profiles can be dragged onto the newly created Load Profile sections.
 - Undo/Redo - Undo or Redo the previously performed Action.
 - Expand All/Collapse All - Expand every tree of the Timeline to show all current components or collapse all trees down to show just the current Logical Ports.
 - Move Up/Move Down - Move the selected resource or port up or down in the sequence. Will not move objects outside of their current level. (i.e. Client Scenarios cannot be moved to a different network profile using this method).
 - Delete - Remove a component from the Timeline.

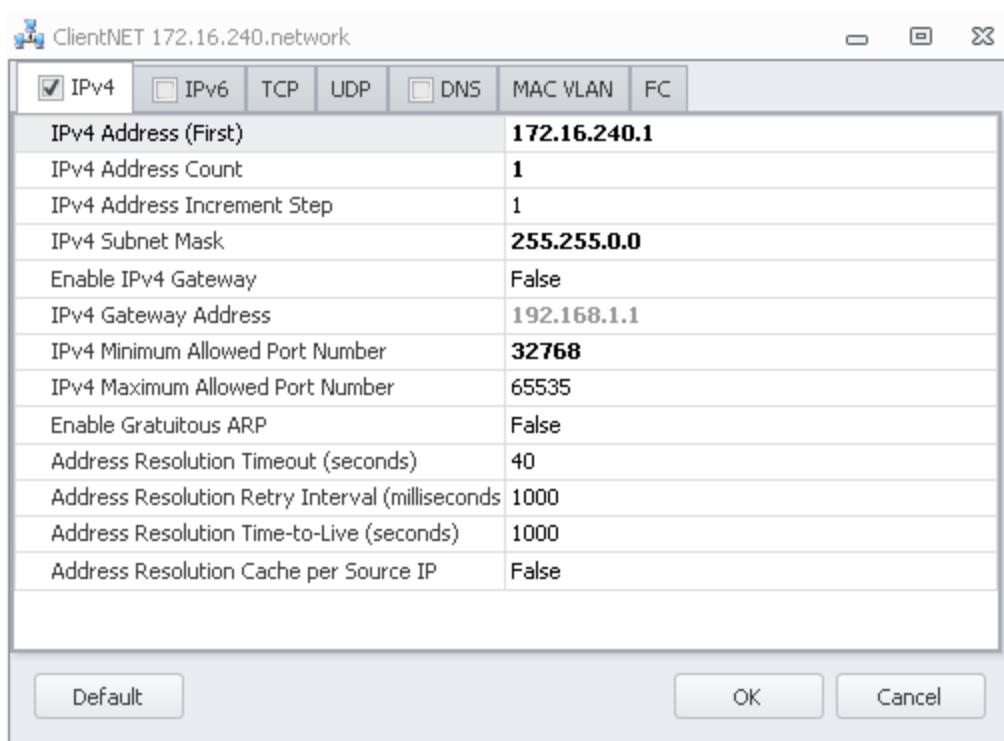
Network Profile

The Network Profile specifies the settings for a test network, either IPv4 or IPv6 or both, as well as certain TCP-specific settings and whether the network should specify its own MAC addresses and VLAN tags.

- The network can contain multiple IPv4 or IPv6 addresses (as generated by the Count and Increment settings)
- Port numbers are generated within a defined range. The default values range from 2000 to 65535, this property needs to be adjusted when accessing third party servers that only allow access from a particular range of ports. Port numbers for IPv4 and IPv6 networks are independent of each other and are defined in the appropriate tab.
- TCP tab controls operational behaviors of the Load DynamiX Appliance's TCP stack.
- UDP tab controls the UDP Inactivity Timeout value.
- MAC/VLAN tab controls operational behaviors of MAC address emulation and VLAN emulation.
- IPv4/IPv6 tabs control address ranges, gateway requirements, port numbers and ARP/NDP protocol behaviors.
- DNS tab controls DNS protocol behavior for such items as DNS server IP address, Time to Live, DNS Query Retry Interval, DNS Resolution timeout, Cache locality and Resource Record Set Order.
- FC tab controls NPIV parameters such as Enabled (True/False), NPIV count and the base NPIV WWPN.

The Network Profile resource is organized into 6 tabs: IPv4 parameters , IPv6 parameters , TCP parameters, DNS parameters, MAC and VLAN parameters and Fibre Channel (FC) parameters.

The various tab parameters are shown below



IPv4 Field	Value
IPv4 Address (First)	Base IPv4 address for this network
IPv4 Count	The number of IPv4 addresses available to this network
IPv4 Address Increment Step	Value added to the current IPv4 address to create the next address
IPv4 Subnet Mask	Subnet mask for the IPv4 network
Enable IPv4 Gateway	True/False - use the IPv4 Gateway address specified
IPv4 Gateway Address	IPv4 address of Gateway device if Enable IPv4 Gateway is True
IPv4 Minimum Allowed Port Number	IPv4 port range definition - low end of range
IPv4 Maximum Allowed Port Number	IPv4 port range definition - high end of range
Enable Gratuitous ARP	See below
Address Resolution Timeout (seconds)	The maximum time in seconds for the Address Resolution cycle to complete (the ARP protocol process to resolve the target IPv4 address of an Open Connection Action)
Address Resolution Retry Interval (milliseconds)	The interval in milliseconds between attempts to resolve an IPv4 address using the ARP protocol. The maximum time to wait for an ARP response
Address Resolution Time-to-Live (seconds)	The period in seconds in which ARP responses are kept in the cache

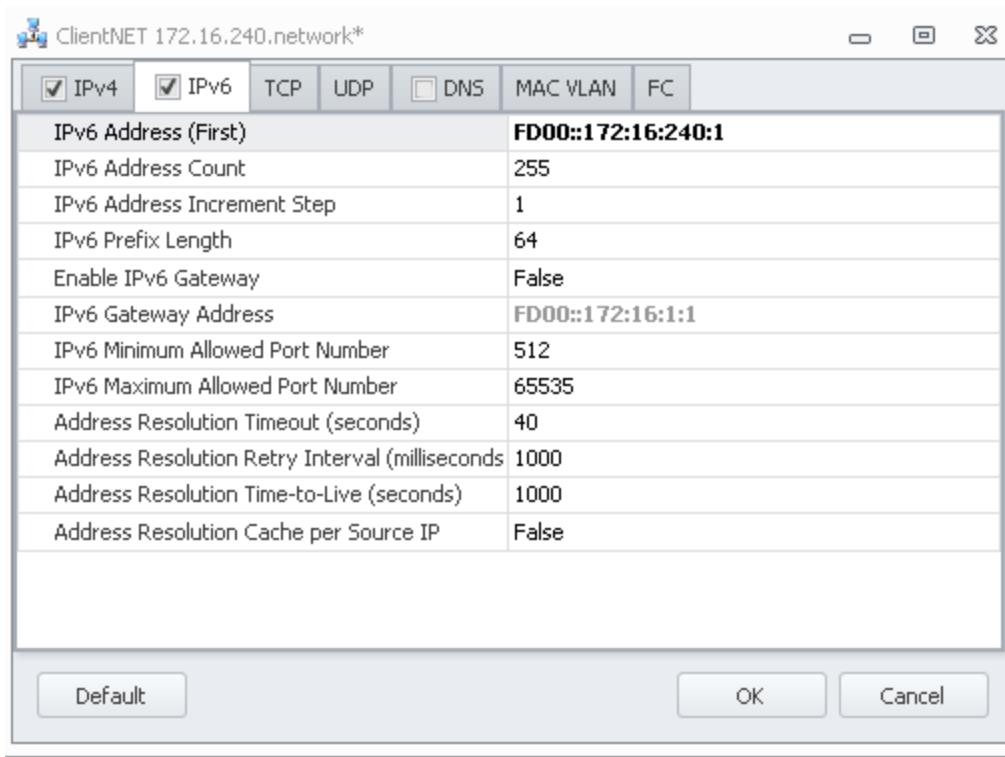
Address Resolution Cache-per-Source IP (True/False)	If set to False (default), there is one ARP cache entry for each unique destination IP address. If set to True, there will be an ARP cache entry for each unique destination IP - source IP address pair. If set to False, the ARP cache is shared by all scenarios using a given port, and if set to True – by all scenarios using a given source IP address.
---	--

Note: Gratuitous ARP

If the use of gratuitous ARP packets is enabled, then before the first use of every new association between IP address and MAC address, the new association is broadcast to the network via ARP, with intention to notify listening ARP caches of the need to refresh its content with regard to the specific association.

By default the use of gratuitous APR packets is disabled, and the use of gratuitous ARP is not required for any Load DynamiX operations. It is relatively common for 3rd party servers to not handle ARP caching correctly. The most common issue is that after a 3rd party server's ARP cache has associated the particular IP address "A" with a particular MAC address "B", then the re-association of IP address "A" with different MAC address "C" is often not noticed by the server ARP cache until it is refreshed, which may take a relatively long time.

IPv6 Parameters are shown below



IPv6 Field	Value
IPv6 Address (First)	Base IPv6 address for this network
IPv6 Count	The number of IPv6 addresses available to this network
IPv6 Address Increment Step	Value added to the current IPv6 address to create the next address
IPv6 Prefix Length	The length of the routing Prefix for all IPv6 addresses
Enable IPv6 Gateway	True/False - use the IPv6 Gateway address specified

IPv6 Gateway Address	IPv4 address of Gateway device if Enable IPv6 Gateway is True
IPv6 Minimum Allowed Port Number	IPv6 port range definition - low end of range
IPv6 Maximum Allowed Port Number	IPv6 port range definition - high end of range
Address Resolution Timeout (seconds)	The maximum time in seconds for the Address Resolution cycle to complete (the NDP protocol process to resolve the target IPv6 address of an Open Connection Action)
Address Resolution Retry Interval (milliseconds)	The interval in milliseconds between attempts to resolve an IPv6 address using the NDP protocol. The maximum time to wait for an NDP response.
Address Resolution Time-to-Live (seconds)	The period in seconds in which NDP responses are kept in the cache
Address Resolution Cache-per-Source IP (True/False)	If set to False (default), there is one NDP cache entry for each unique destination IP address. If set to True, there will be an NDP cache entry for each unique destination IP - source IP address pair. If set to False, the NDP cache is shared by all scenarios using a given port, and if set to True – by all scenarios using a given source IP address.

NOTES:

Calculating the number of IPv4 addresses that a Network Profile will provide is done by the following algorithm.

Step 1. Determine the Maximum Number of IPv4 Addresses (MNAv4) possible using the IPv4 Subnet Mask.

Step 2. Compare MNAv4 with the IPv4 Count input. The smaller of the two values is now the MNAv4.

Step 3. Divide MNAv4 by IPv4 Address Increment Step to determine the Calculated Address Count (CACv4).

CACv4 is the maximum number of IPv4 addresses for a given IPv4 Network Profile

Calculating the number of IPv6 addresses that a Network Profile will provide is done by the following algorithm.

Step 1. Determine the Maximum Number of IPv6 Addresses (MNAv6) possible using the IPv6 Prefix Length.

Step 2. Compare MNAv6 with the IPv6 Count input. The smaller of the two values is now the MNAv6.

Step 3. Divide MNAv6 by IPv6 Address Increment Step to determine the Calculated Address Count (CACv6).

CACv6 is the maximum number of IP addresses for a given IPv6 Network Profile

IPv4 Example:

Netmask 255.255.255.0 = 256 Max possible IPv4 addresses;

IPv4 Count = 1000;

IPv4 Step = 2;

Smaller of (256,1000); MNAv4 = 256;

MNAv4/IPv4 Step = 128 IPv4 addresses will be generated by this Network Profile.

Address Range.

The Address Range generated by a Network Profile is determined by the CACv4 or CACv6 calculations, Step and the Netmask or Prefix values. IPv4 examples:

IPv4 Base Address: 192.168.1.0

Netmask: 255.255.255.0

CACv4: 128

Step: 2

Range: 192.168.1.0, 192.168.1.2, 192.168.1.4, 192.168.1.6, ..., 192.168.1.252, 192.168.1.254

IPv4 Base Address: 192.168.1.250

Netmask: 255.255.255.0

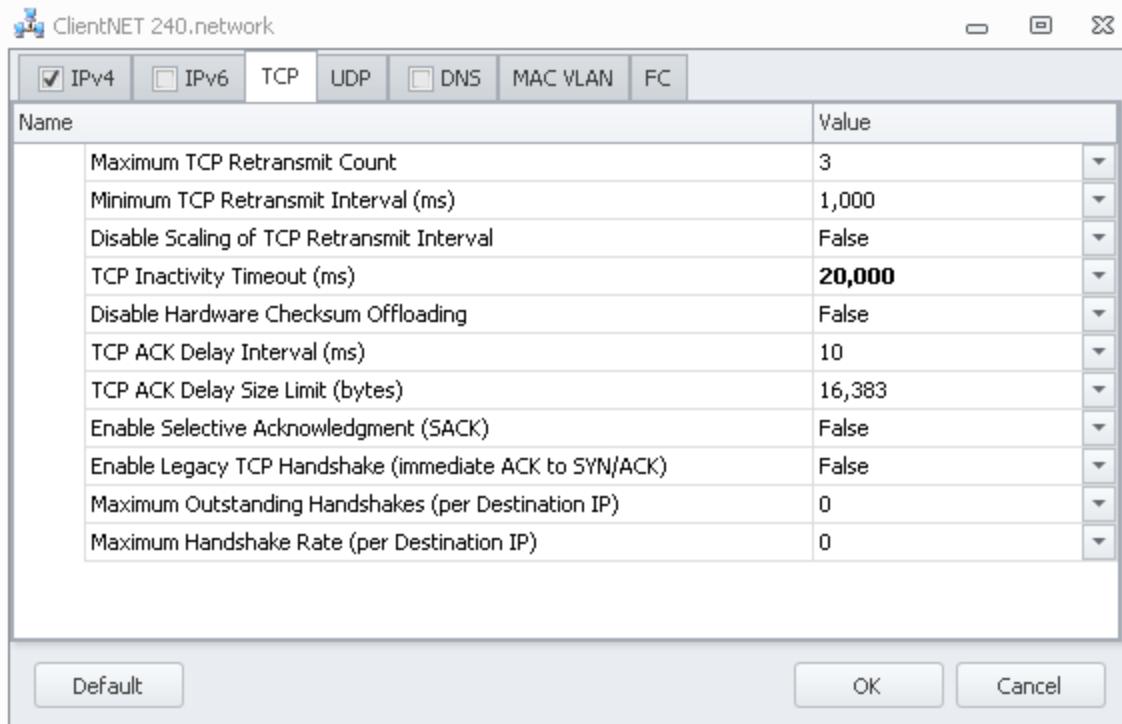
CACv4: 12

Step: 1

Range: 192.168.1.250, 192.168.1.251,

192.168.1.252,...,192.168.1.255,192.168.1.0,192.168.1.1,...192.168.1.5

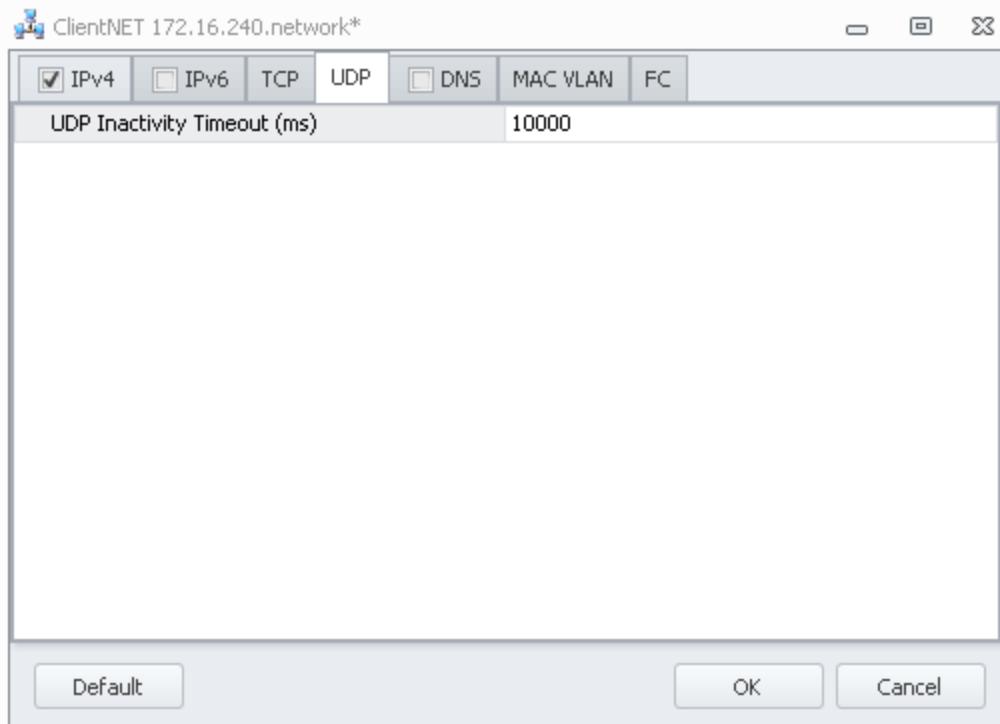
TCP Settings



TCP Setting	Value
Maximum TCP Retransmit Count	Maximum number of times TCP packets will be retransmitted
Minimum TCP Retransmit Interval (ms)	Interval between TCP retransmits
Disable Scaling of TCP Retransmit Interval	True/False - disables the TCP adaptive retransmission algorithm
TCP Inactivity Timeout (ms)	TCP inactivity period timeout interval in milliseconds
Disable Hardware Checksum Offloading	True/False - Disables hardware generated checksum generation

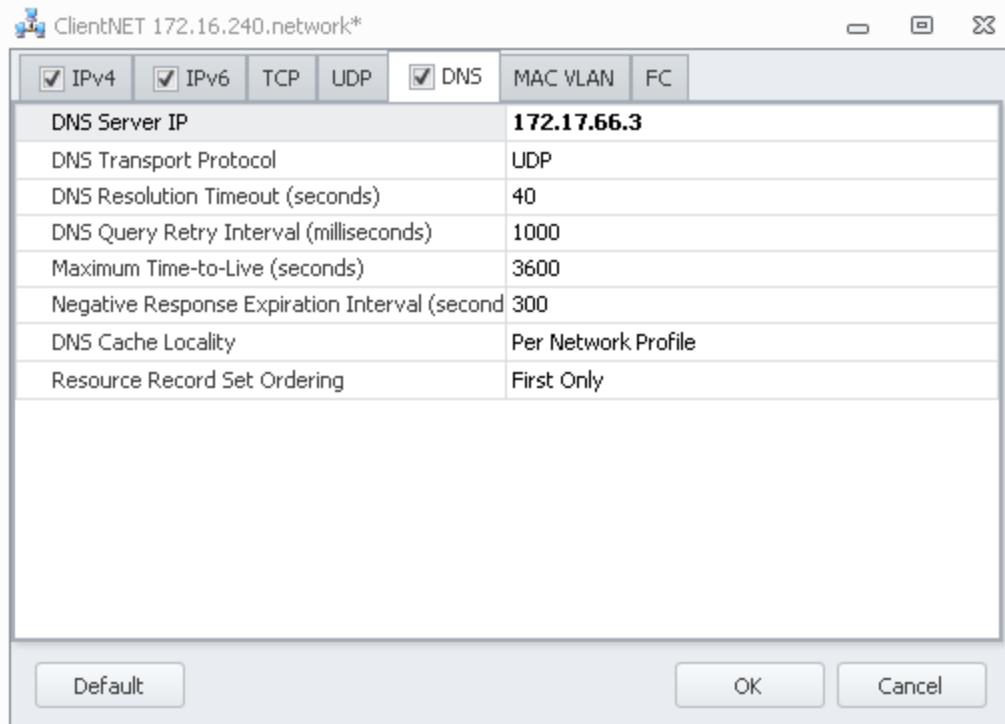
TCP ACK Delay Interval (ms)	The delay in ms between TCP ACKs (see Appendix: Jumbo Frames and Delayed ACK)
TCP ACK Delay Size Limit (bytes)	The number of unacknowledged bytes of data before a delayed ACK will be sent (see TCP ACK Behavior in Appendix: Jumbo Frames and Delayed ACK)
Enable Selective Acknowledgment	True/False - enables TCP SACK processing of out of order/missing packets
Enable Legacy TCP Handshake	TCP stack sends an ACK message immediately after receiving a SYN/ACK message
Maximum Outstanding Handshakes (per Destination IP)	This input helps control TCP reconnect behaviors when a TCP connection closes unexpectedly in the middle of Project execution. Default value of 0 means unlimited and ensures unchanged behavior from prior releases. An integer value > 0 establishes the maximum number of handshake operations that will be attempted on a given IP address. Used in conjunction with Maximum Handshake Rate (per Destination IP).
Maximum Handshake Rate (per Destination IP)	This input helps control TCP reconnect behaviors when a TCP connection closes unexpectedly in the middle of Project execution. Default value of 0 means unlimited and ensures unchanged behavior from prior releases. An integer value > 0 establishes the maximum rate (attempts per second) of handshake operations that will be attempted on an IP address. Used in conjunction with Maximum Outstanding Handshakes (per Destination IP).

UDP Settings



UDP Setting	Value
UDP Inactivity Timeout (ms)	UDP inactivity period timeout interval in milliseconds

DNS Settings

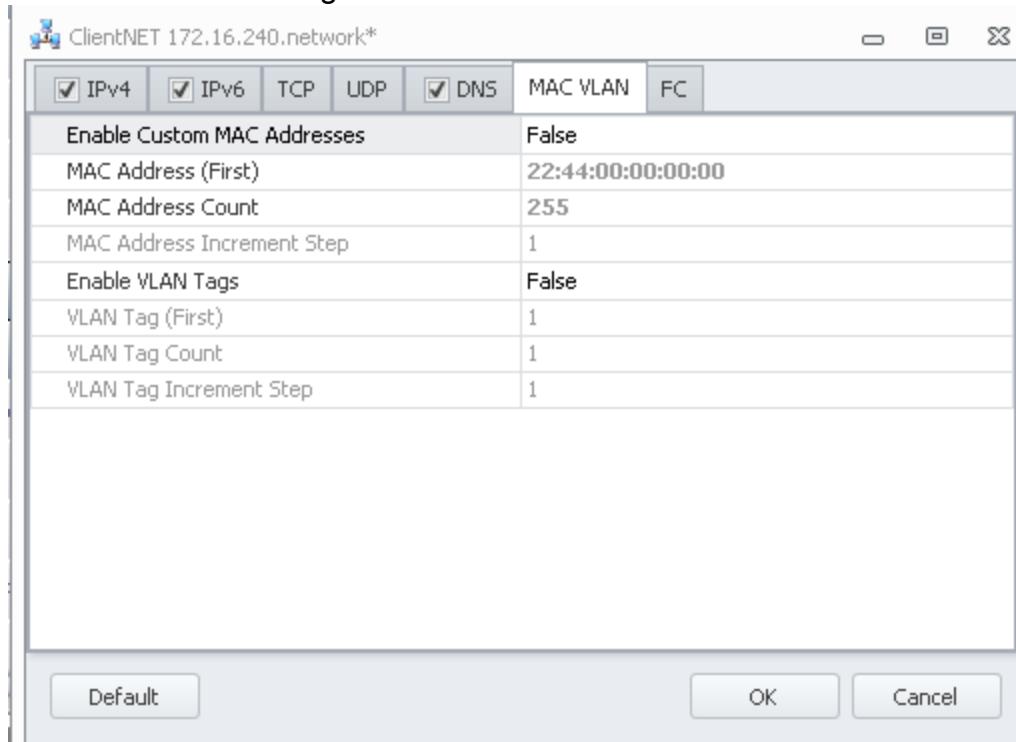


DNS Parameter	Meaning	Options
DNS Server IP	IP address of the DNS server	IPv4 or IPv6 format
DNS Transport Protocol	Protocol used to send/receive DNS packets	UDP or TCP
DNS Resolution Timeout (seconds)	Timeout period in seconds for DNS requests	0 to 100 seconds
DNS Query Retry Interval (milliseconds)	The retry interval in milliseconds for DNS packets	0 to 100000 milliseconds
Maximum Time to Live (seconds)	The maximum time DNS responses are kept in the cache	0 to 360000 seconds
Negative Response Expiration Interval (seconds)	The length of time in seconds that a Negative Response is held in the Client's cache	0 to 360000 seconds
DNS Cache Locality	The locality of the DNS cache entries	Network Profile, Port, Source
Resource Record Set Ordering	How multiple responses are handled	First Only, Round Robin, Random

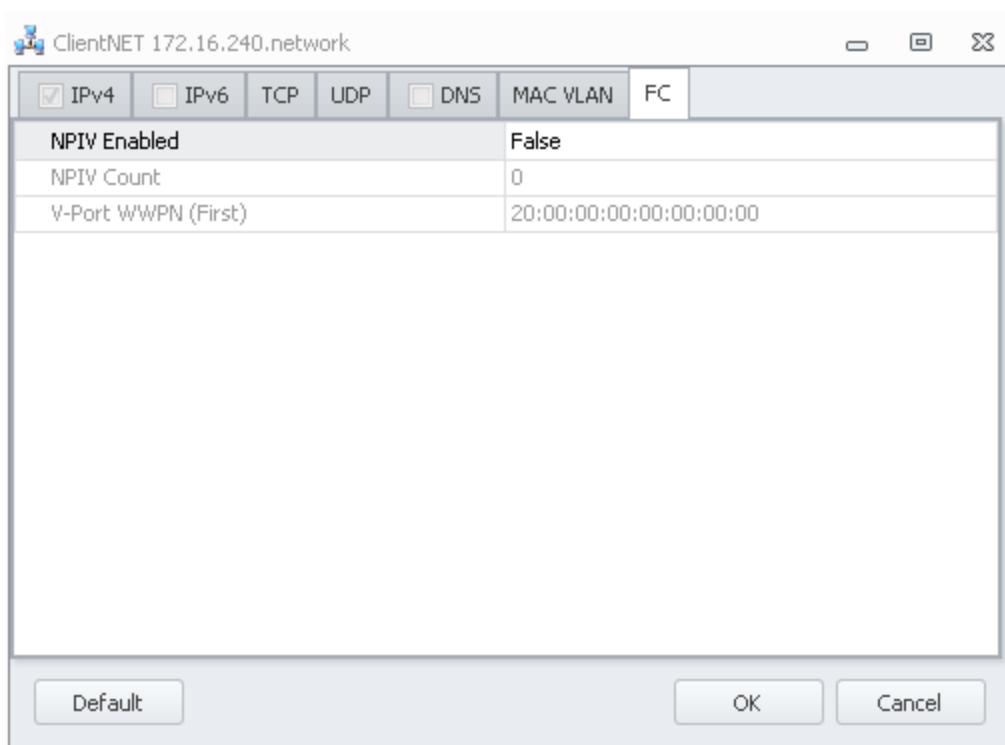
The typical Load DynamiX Client Scenario starts with an Open Connection Action for the Device Under Test and concludes with Close Connection to that device. The Device Under Test can be specified as an IP Address or as a name that can be resolved by DNS. If DNS name lookup is to be used, the DNS tab of the Network Profile of that Scenario must be enabled (box checked) and the IP Address of a DNS server that will resolve the Device Under Test's name filled in. Otherwise an IP Address is expected as input to the Open Connection Action.

See [Appendix: DNS](#) for more details on DNS support.

MAC and VLAN Settings



MAC and VLAN Settings	Value
Enable Custom MAC Addresses	True/False - Use the MAC addresses specified below in this network
MAC Address (First)	Base MAC Address
MAC Address Count	Count of MAC Addresses to use
MAC Address Increment Step	Increment between simulated MAC addresses
Enable VLAN Tags	True/False - use VLAN tags specified below
VLAN	Base VLAN number
VLAN Count	Number of VLANs to simulate
VLAN Step	Increment between simulated VLANs

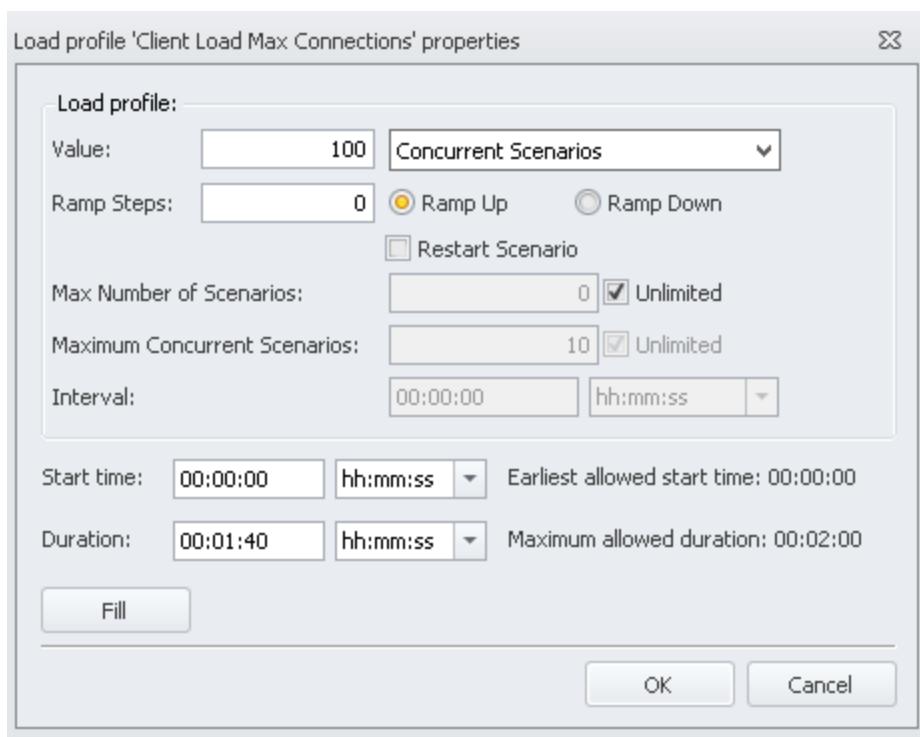


FC Settings	Value
NPIV Enabled	True/False - Use a Virtual WWPN as initiator WWPN in the scenarios assigned to this Network Profile
NPIV Count	The count of Virtual WWPN to be used
V-Port WWPN (First)	Base Virtual WWPN address

See [Reference: FC/iSCSI/SCSI Commands](#) chapter for more discussion on NPIV. NPIV settings made in the Network Profile take run-time precedence over any NPIV setting at the Port level.

Load Profile

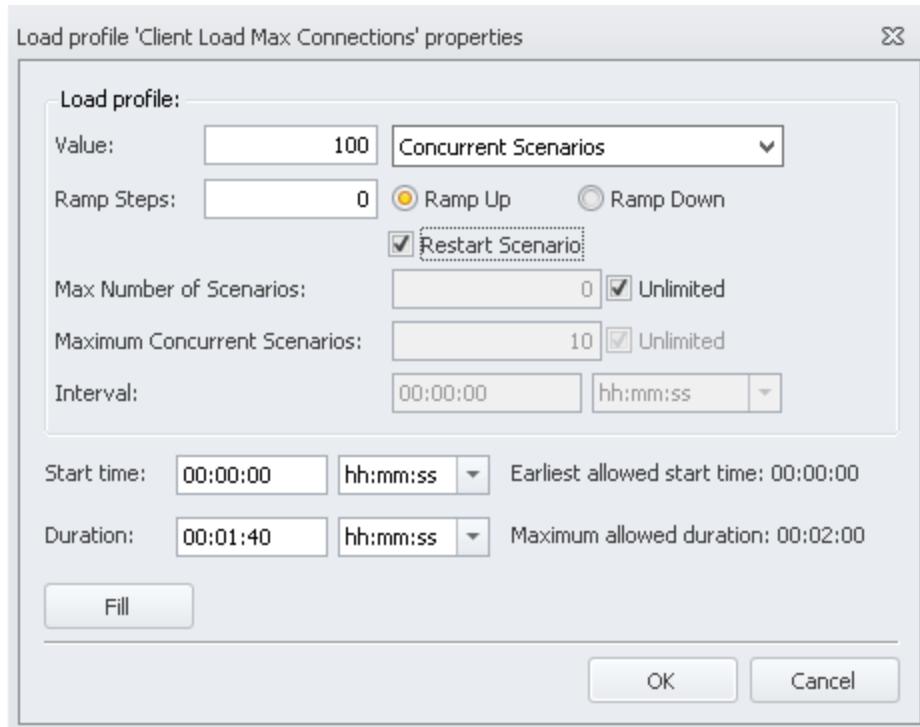
The Load Profile controls the load generated by a Client or Server Scenario.



Input Field	Impact on the Load generated
Type	<p>Load type (the units of the load specification):</p> <ul style="list-style-type: none"> · Concurrent Scenarios · New Scenarios Per Second · Concurrent Actions · New Actions Per Second · Concurrent Connections · New Connections Per Second · Bandwidth · Throughput · New Scenarios Per Interval
Ramp Steps	<p>Specifies the rate at which the test reaches peak level. For example, if Value=1000, Load Type = Concurrent Scenarios, and Ramp Steps=10, the first increment of time covers the first 100 Scenarios, the second time increment covers the second 100 Scenarios, the third time increment covers the third 100 Scenarios, and so on up to 1000 concurrent scenarios. The Duration of the Load Profile is divided into Steps number of time increments. Ramp Steps = 0 or 1 are the same and produce an immediate Ramp Steps number of the selected Load Type.</p>
Ramp Up / Ramp Down	<p>Ramp Down = The test starts with the peak performance and moves down (e.g., using the prior example, the first test would use 1000 Concurrent Scenarios and the 10th and final iteration would use 100 Actions/second). Ramp Up = low performance up to peak performance.</p>
Start time	<p>How many time units into the Timeline this Load Profile begins. Units = Seconds, Min:Seconds and Hrs:Min:Seconds.</p>
Duration	<p>How many time units into the Timeline this Load Profile executes for. Units = Seconds, Min:Seconds and Hrs:Min:Seconds</p>
Max Number of Scenarios	<p>Limit the Scenario execution to a Maximum number. When this number is completed the Scenario will stop executing</p>
Maximum Concurrent Scenarios	<p>Use this value in conjunction with other load types such as New Scenarios per Second to control the concurrently executing scenarios.</p>

Interval	Used in conjunction with the New Scenarios Per Interval load type to specify the time interval.
Restart Scenario	Forces a Scenario to restart with the same IP address (if applicable) and current User Parameters file settings however it completes (Success, Abort, Failure).

Restart Scenario



The Restart Scenario check box allows the Tester to specify that some elements of a Scenario are re-used whenever the Scenario is restarted following:

- Successful execution
- Aborted Execution
- Failed Execution

The elements that are re-used are the IP address and the User Parameter file state. Normally when a Scenario exits, the IP Address is incremented, the Port is incremented and the User Parameter file pointers are incremented. If the Restart Scenario check box is set then the User Parameter pointers are left as is and the IP Address is left as is. The Port number the Scenario used is incremented.

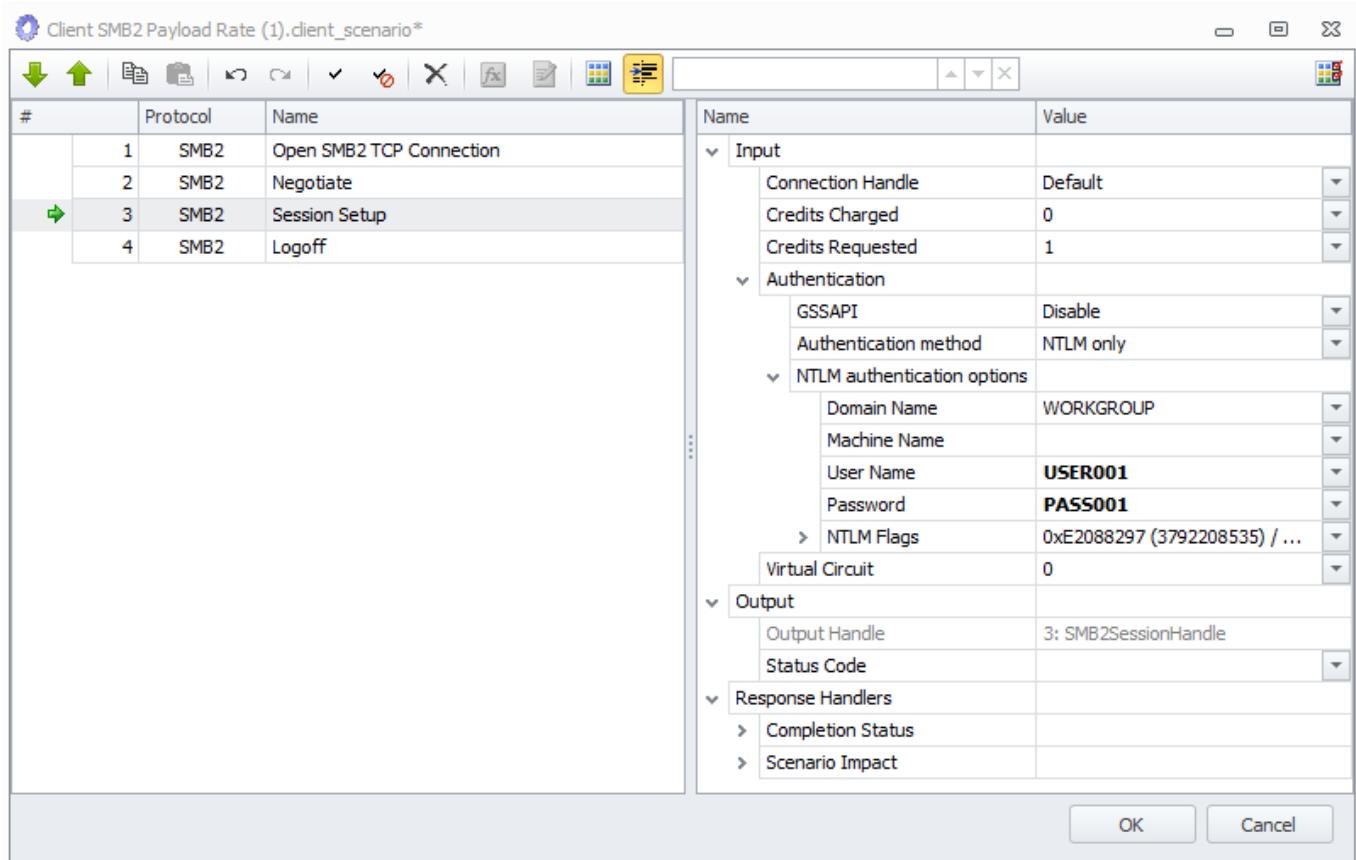
The use of Advance User Parameters or Reset User Parameters within a Scenario that has Restart Scenario configured in the Load Specification will impact the User Parameter settings that are used at the start of the next instance of the Scenario. If it is necessary to use Advance User Parameters and Reset User Parameters in a Scenario with Restart Scenario set then the Tester is advised to have a Reset User Parameters with Scope == Scenario as the very first Action in the Scenario.

Use of Restart Scenario on a Fibre Channel Scenario will only impact User Parameter settings because Fibre Channel Scenarios do not use IP addresses or Ports.

The Restart Scenario feature in a Load Specification for a Server Scenario is disabled.

Load Profile Examples

Using a simple Scenario such as the one below we can use the Load Profile to determine certain kinds of behavior



If a Stair Step of running scenarios or connections was required (an increasing number of scenarios executing or connections open for a given period of time), the following Load Profile could be used

Load profile 'Client Load Max Connections' properties

Load profile:

Value:	1,000	Concurrent Scenarios
Ramp Steps:	5	<input checked="" type="radio"/> Ramp Up <input type="radio"/> Ramp Down
<input type="checkbox"/> Restart Scenario		
Max Number of Scenarios:	0	<input checked="" type="checkbox"/> Unlimited
Maximum Concurrent Scenarios:	10	<input checked="" type="checkbox"/> Unlimited
Interval:	00:00:00	hh:mm:ss

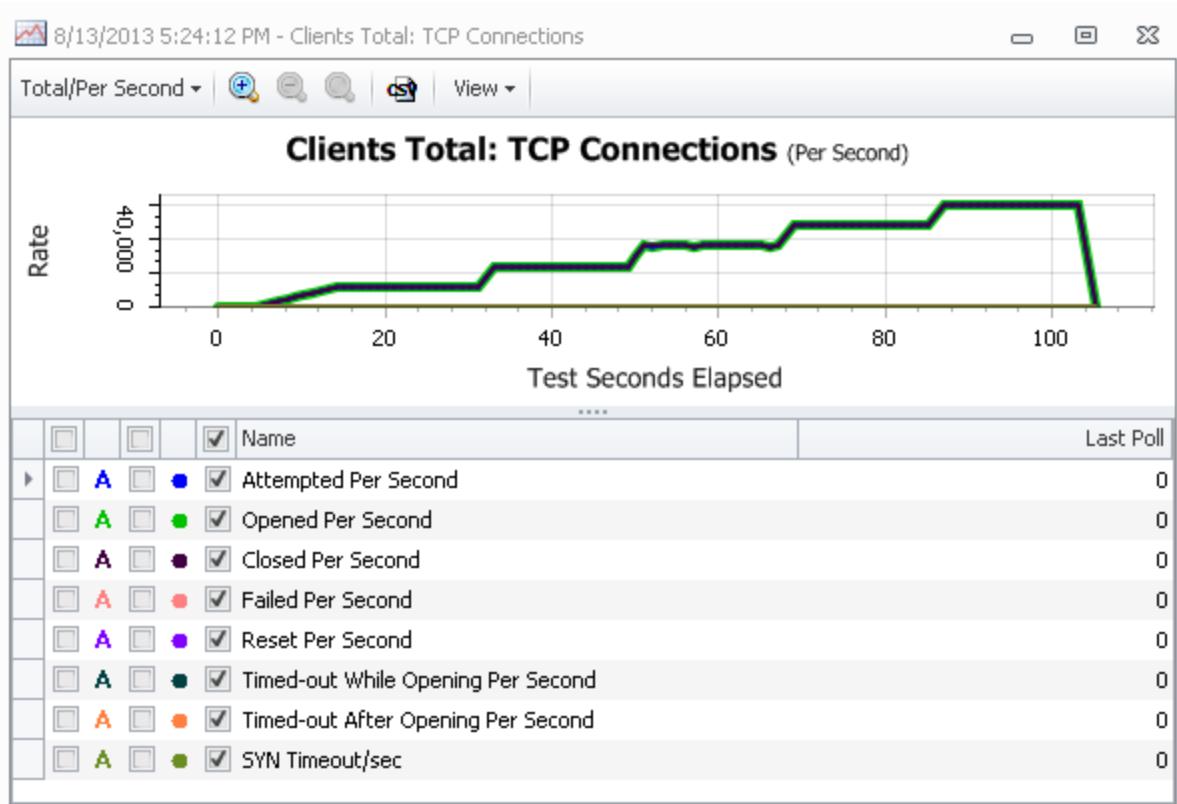
Start time: 00:00:10 hh:mm:ss Earliest allowed start time: 00:00:00

Duration: 00:01:40 hh:mm:ss Maximum allowed duration: 00:02:00

Fill

OK **Cancel**

It will produce a TCP Connection graph of



A new set of 200 connections every approximately 12 seconds. The spikes at the beginning of each 12 second time segment is the Appliance Client software initiating the additional 200 connections and then settling in at the requested rate.

If TCP connections that mimic a sawtooth is desired, the following Scenario and Load Profile would generate the sawtooth graph.

Client SMB2 logon logoff.client_scenario*

#	Protocol	Name
1	SMB2	Open SMB2 TCP Connection
2	SMB2	Negotiate
3	SMB2	Session Setup
4	SMB2	Logoff
5	SWT	Delay Execution
6	SMB2	Close SMB2 TCP Connection

Name	Value
Input	Delay Interval
	5,000

Load profile 'Client Load Max Connections' properties

Load profile:

Value: 1,000 Concurrent Scenarios

Ramp Steps: 1 Ramp Up Ramp Down

Restart Scenario

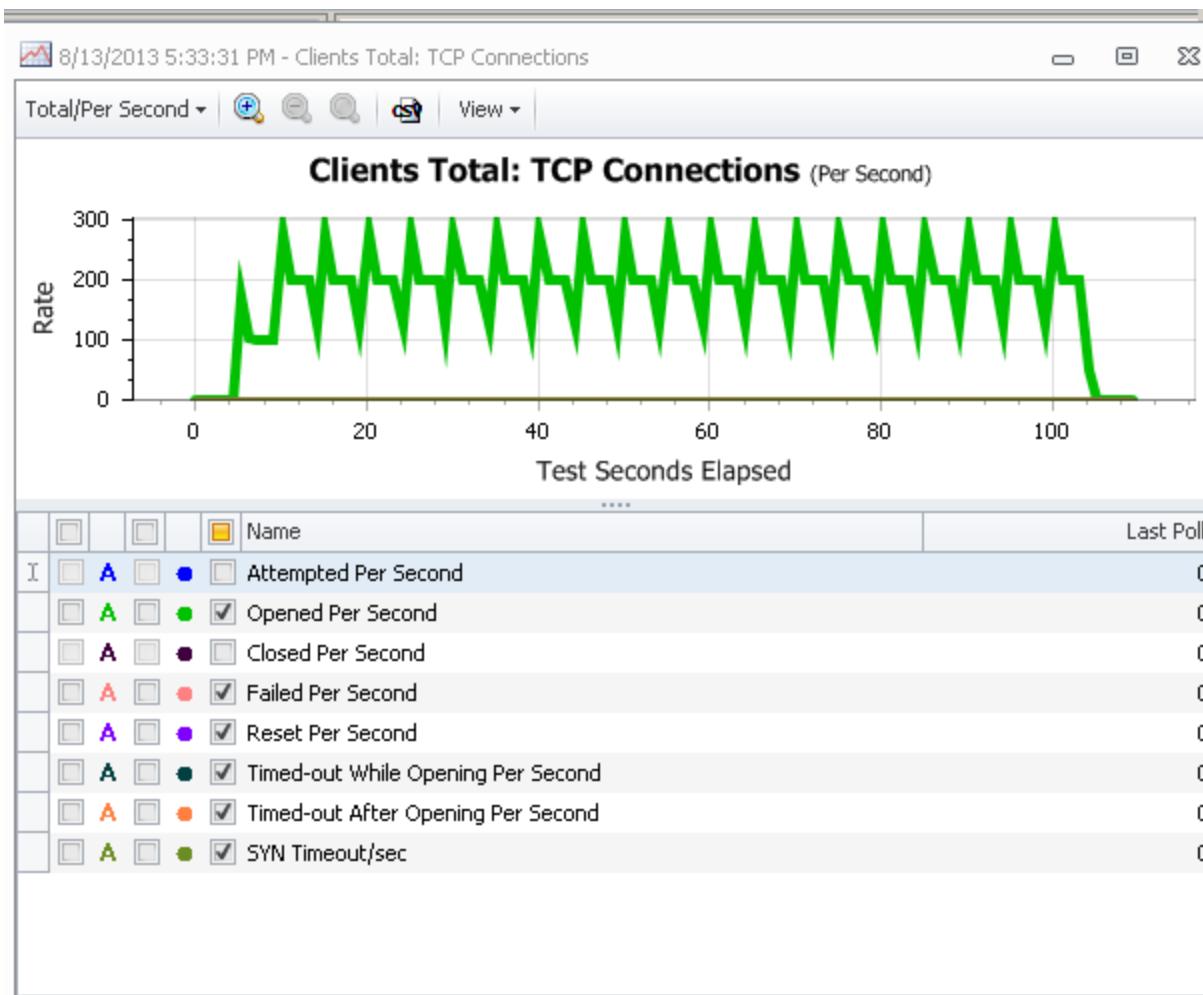
Max Number of Scenarios: 0 Unlimited

Maximum Concurrent Scenarios: 10 Unlimited

Interval: 00:00:00 hh:mm:ss

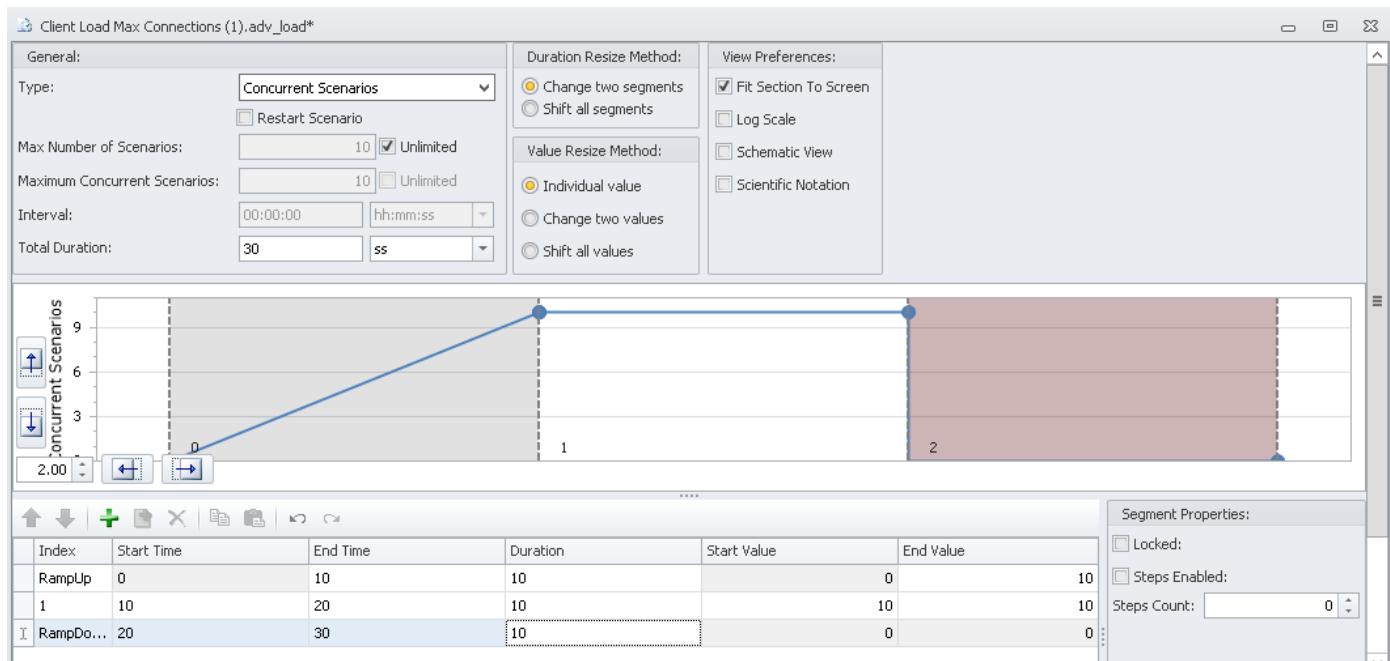
Start time: 00:00:00 hh:mm:ss Earliest allowed start time: 00:00:00

Duration: 00:01:40 hh:mm:ss Maximum allowed duration: 00:02:00



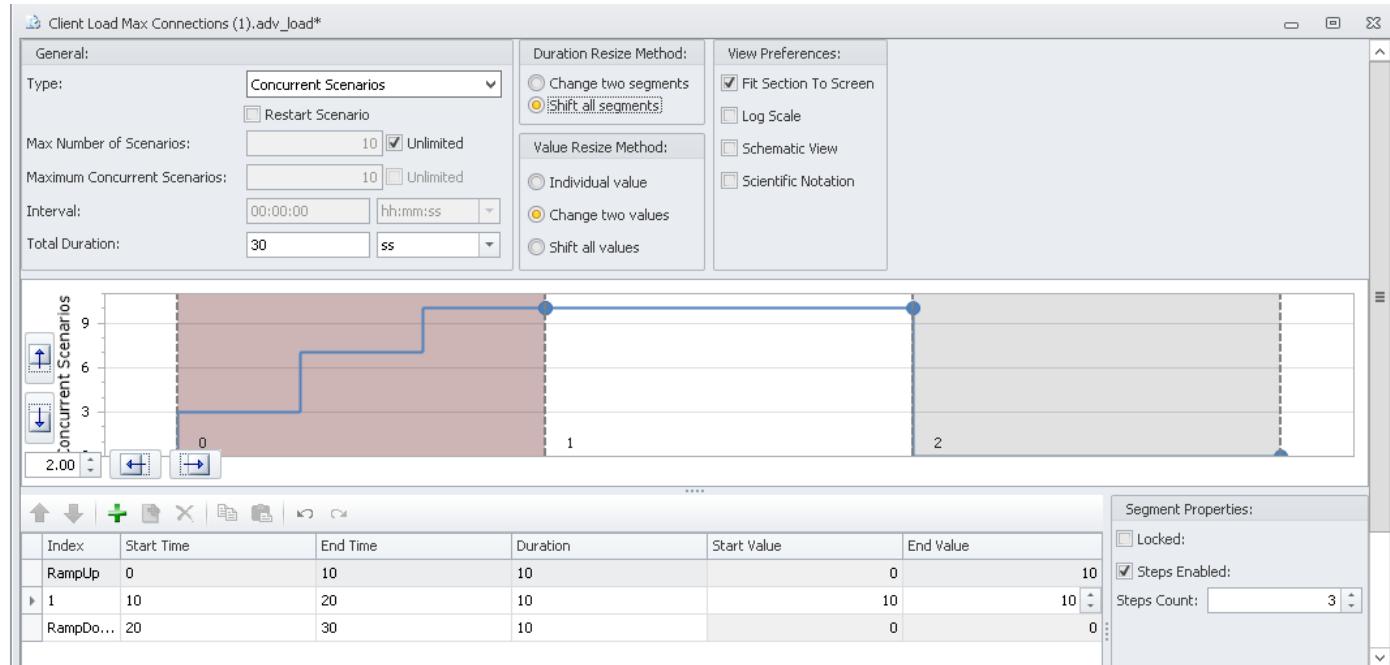
Advanced Load Profile

Load Profiles may also be created using the Advanced Load Profile template. Normally a Load Profile describes a single execution segment of the Project. It could be Ramp Up or Ramp Down or a Load Profile describing one of the main execution segments of the Project. These Load Profiles are all defined separately and dropped on to the Timeline where needed by the Tester. Advanced Load Profiles contain multiple Load Profile segments including Ramp Up and Ramp Down and may contain one or more main execution Load Profile segments. Below is the default Advanced Load Profile dialog:

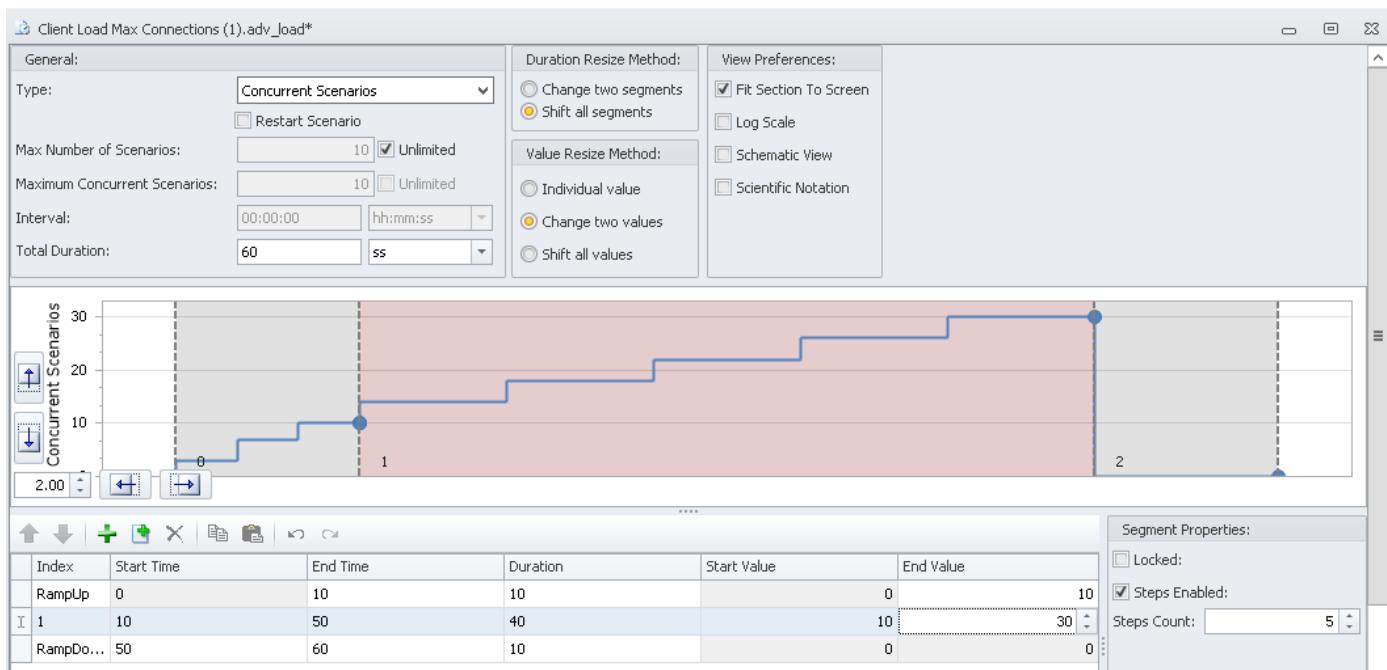


If the desired Advanced Load Profile were to contain a Ramp Up with 3 steps to 10 Concurrent Scenarios and then have those 10 Concurrent Scenarios step up to 30 in 5 steps with an overall execution time of the main segment to be 40 seconds, the following sets of changes would be made to this Advanced Load Profile.

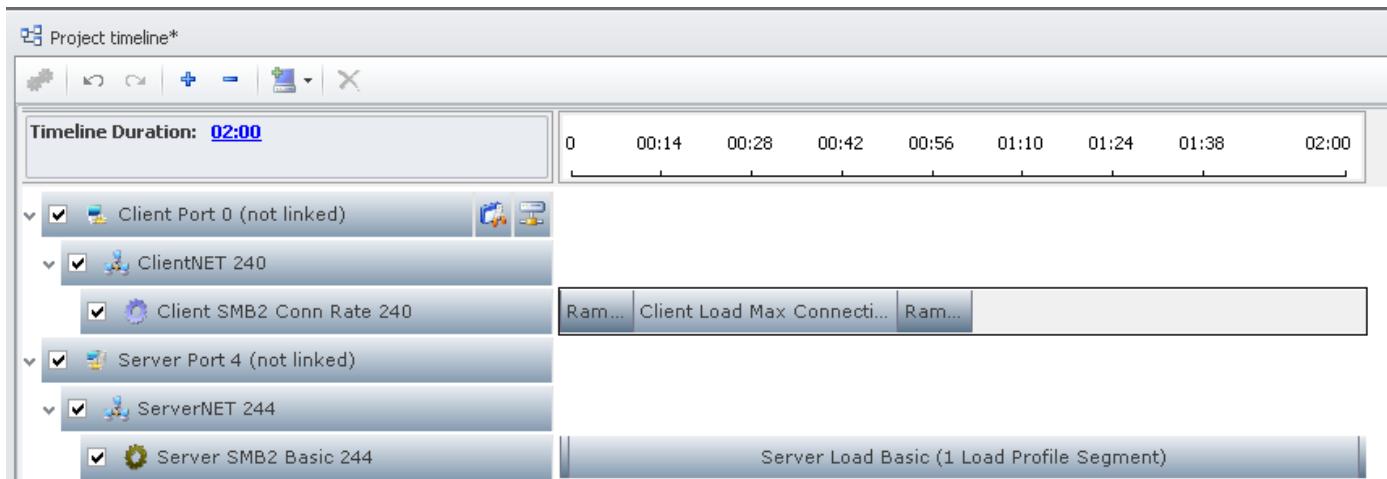
First - Create the Steps in the Ramp Up segment by highlighting the segment (click on the segment in the middle portion of the dialog or click on the segments row in the lower portion of the dialog); click on Steps Enabled on the lower right and increment the Step Count to 3 as shown below.



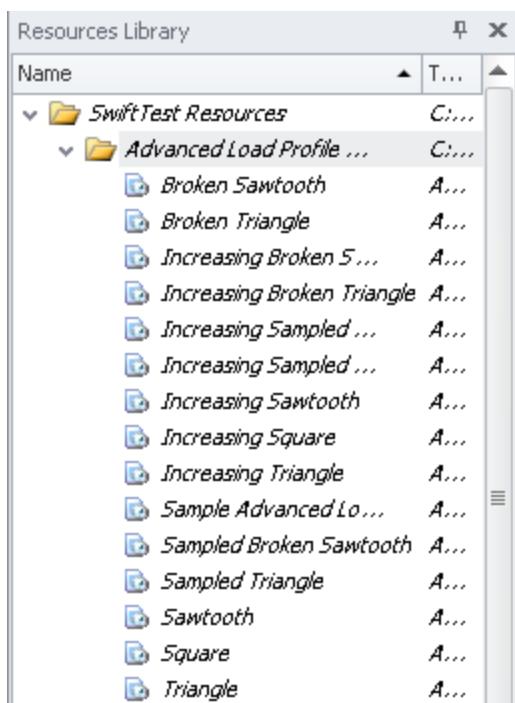
Second - To change the duration, ramp up the main execution section of the Advanced Load Profile to 30 Concurrent Scenarios in 5 steps, highlight the middle section by clicking on the middle row in the lower portion of the dialog and change the Duration field to 40, change End Value to 30, and in the lower right portion, click Steps Enabled and increase Step Count to 5.



Now to use this Advanced Load Profile in a Project, drop it onto the Load Profile section of the Timeline for a Scenario.



In the Load DynamiX Resources folder in the Resources Explorer is a folder containing a collection of 14 pre-defined Advanced Load Profiles that follow certain patterns and are ready to deploy.



Advanced Load Profile Name	Pattern
Sawtooth	Sawtooth Wave, 10 concurrent scenarios at the peak, 0 steps
Increasing Sawtooth	Sawtooth Wave, 10,20,30,40,50 concurrent scenarios at the peak, 0 steps
Broken Sawtooth	Sawtooth Wave, 10 concurrent scenarios at the peak, 0 steps, 30 sec pause between peaks
Increasing Broken Sawtooth	Sawtooth Wave, 10,20,30,40,50 concurrent scenarios at the peak, 0 steps, 30 sec pause between peaks
Sampled Broken Sawtooth	Sawtooth Wave, 10 concurrent scenarios at the peak, 5 steps up to peak, 30 sec pause between peaks
Increasing Sampled Broken Sawtooth	Sawtooth Wave, 10,20,30,40,50 concurrent scenarios at the peak, 5 steps
Triangle	Triangle Wave, 10 concurrent scenarios at the peak
Broken Triangle	Triangle Wave, 10 concurrent scenarios at the peak, 30 sec pause between peaks
Sampled Triangle	Triangle Wave, 10 concurrent scenarios at the peak, 5 steps up to/down from peak
Increasing Triangle	Triangle Wave, 10, 20, 30, 40,50 concurrent scenarios at the peak, 0 steps
Increasing Broken Triangle	Triangle Wave, 10, 20, 30, 40,50 concurrent scenarios at the peak, 0 steps, 30 sec pause between peaks
Increasing Sampled Triangle	Triangle Wave, 10, 20, 30, 40,50 concurrent scenarios at the peak, 5 steps
Square	Square Wave 10 concurrent scenarios at the peak, 0 steps
Increasing Square	Square Wave 120, 240, 360, 480 concurrent scenarios, 0 steps

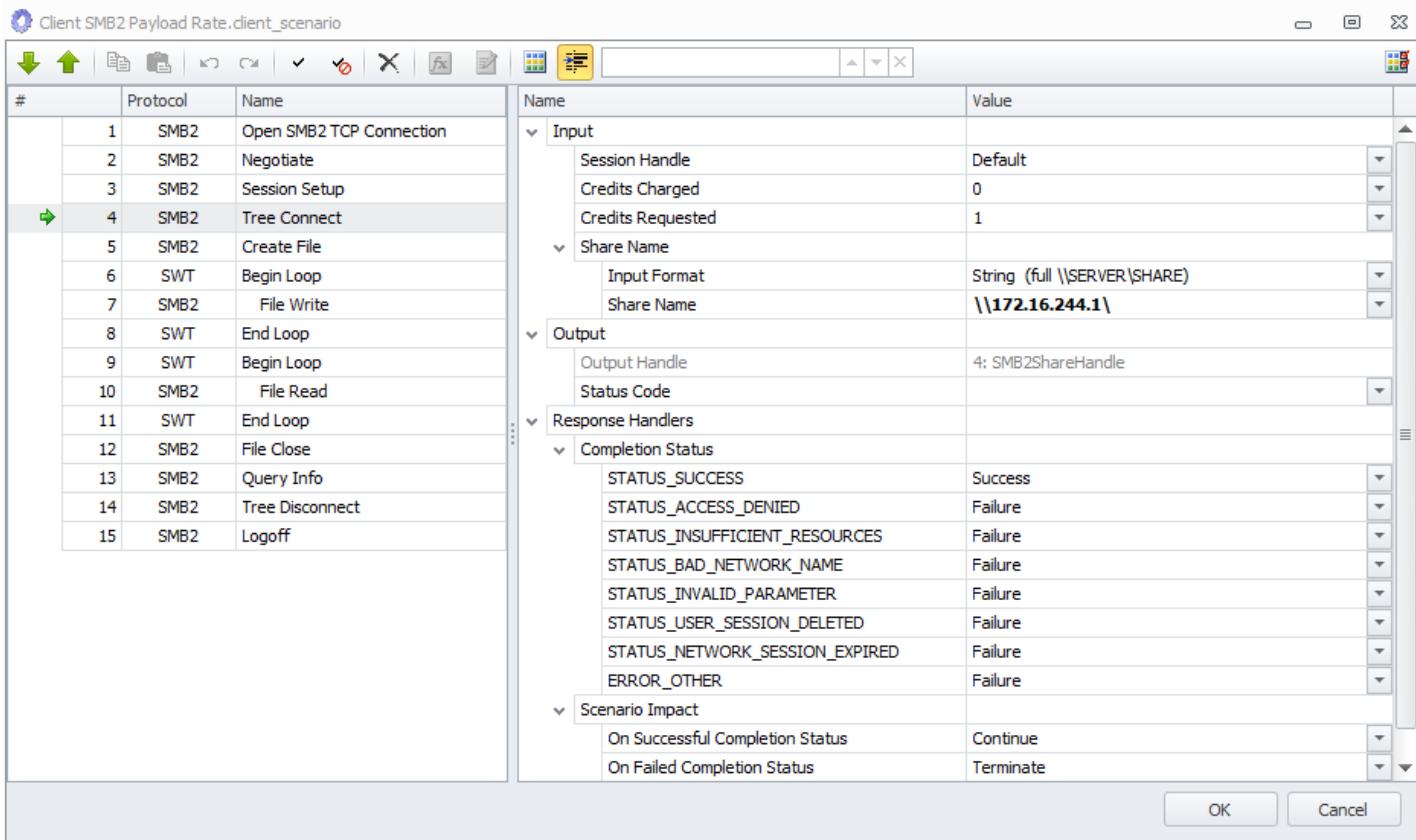
There is one additional Advanced Load Profile in this folder named Sample Advanced Load Profile - it is an empty Advanced Load Profile.

Client/Server Scenario

The first part of this chapter was devoted to describing the key components of a Load DynamiX Project. Now we will discuss how to pull these components together into a Project.

The process for creating Client and Server Scenarios is the same. Drag a Client or Server Scenario template from the Resources Library into the Project Explorer or Timeline window and double click it. The Toolbox contains Actions that are appropriate for the kind of template (Client or Server). Drag and drop Actions from the Toolbox onto the Scenario to create the executable portion of a Project.

The Client Scenario below contains a series of Actions that open a connection to an SMB2 server, create a file, write data to that file in a loop, close the file and then disconnect from the SMB2 server. Notice the Response Handlers properties on the right hand side of this Scenario. Response Handlers allow the Tester to determine what the response to the return code for the execution of each Action should be. Response Handlers settings allow an Action to react to specific error return codes and determine whether the Scenario, in which this Action is executing, should Continue executing or should Terminate (stop) executing. In the example below, the SMB2 Tree Connect Action specifies that if the return code is STATUS_SUCCESS that the Scenario should Continue executing and for all other return codes, the Scenario should Terminate. See [Advanced Concepts: Response Handling](#) for more details.



Scenario template fields.

Column Name	Contents
#	Sequential command identifier (number)

Protocol	<p>Action protocol:</p> <p>Client:</p> <ul style="list-style-type: none"> • SMB (CIFS-SMB) • SMB2, SMB2/SMB3, SMB2/MSRPC • NFSv2 • NFSv3 • NFSv4, NFSv4.1 • iSCSI • Kerberos • HTTP • SCSI • FC • TCP Echo • OpST-Sw ift, CDMI, Amazon S3 • SWT (Load DynamiX Scenario Control) <p>Server:</p> <ul style="list-style-type: none"> • SMB (CIFS-SMB) • SMB2 • NFSv3 • iSCSI • HTTP • TCP Echo
Name	Action Name.
Name	Action Input field Name.
Value	Action Input field value. From the drop-down list, you have the option of selecting either a standard value or specifying a custom value such as a User Parameter (see Advanced Concepts: User Parameters) or the output of a Function (See Dynamic Parameter Generation below).

Actions Inputs

The Inputs required by an Action depend on the specific Action. Inputs might be a File Name String as required in an Action that creates a File. Inputs might be an Integer for Actions that write a certain number of Bytes to a File. Inputs might be a hexadecimal string when providing Flags in a Authentication Action. The type of input required (string, integer, hexadecimal string, etc.) should be obvious based on the kind of input field that is being filled. For more information on Integer inputs and Integer Shorthand see [Reference: Action Input Shorthand](#). For more information on Strings and other functions as well as Formula that can provide input to Actions, see [Appendix: Functions and Formula](#). For more information on User Parameter Files as inputs to Actions, see [Advanced Concepts: User Parameter Files](#). For more information on Variables as input to Actions, see [Advanced Concepts: Variables and Aliases](#).

Actions and Handles

One of the common elements of most Load DynamiX Actions is that they will take some form of Handle as input and will output some kind of Handle. Handles are the element that binds one Load DynamiX Action to Another. In the screenshot above, there is a CIFS-SMB **Tree Connect** Action highlighted. This Action takes as input a Session Handle (the output of the CIFS-SMB **Session Setup** Action) and outputs an SMBShareHandle which is input to (among others) the CIFS-SMB **Create or Open File** Action. When there are multiple instances of Actions generating a Handle that is possible input to another Action, the input field that expects the Handle will show all possible inputs in

the drop down menu. The default value that is provided for Handle inputs to most Actions is the word Default. If there is a single possible Handle for input, Default == that Handle. If there are multiple Handles available, Default == the last Handle in the list (i.e. the most recent Handle of the type required).

Using Default for all Handle input fields can cause Project issues if there are more than one Handle available and what is desired is not the last Handle in the list. It is recommended in multiple Handle input situations, to explicitly pick which Handle is to be used.

[Executing Tests and Assessing Results](#) describes how to compile, run, and assess the output generated by executing a Project.

User Parameters

The User Parameter template enables you to define parameters that will be provided to Scenario Action input fields during test execution. See [Advanced Concepts: User Parameters](#) for a detailed discussion.

Data File System

Using Data File Systems and Data Verification you can create file systems content and structure on Load DynamiX servers as well as specify the data content to be transmitted by all CIFS-SMB and NFS Write I/O operations as well as verify the data content received by CIFS-SMB and NFS Read I/O operations. See [Advanced Concepts: Data File Systems and Data Verification](#) for more details.

Creating Load DynamiX Projects

The most expedient way to construct a new Project is to use one of the sample tests from the Load DynamiX Projects folder. The Load DynamiX Projects are delivered Read-Only, so you must open it and save it before it can be used. Simply open one of these Projects and use File → Save Project As with a new name before you edit it. There are two other ways to create Projects - using the Project Wizard or creating them manually.

Project Wizard

The Load DynamiX TDE also provides a means of creating test Projects from scratch. To construct a basic test Project you can use the Project Wizard.

To create a new test

- Select File → New Project.
- In the New Project dialog select Wizards on the left hand side, and either New Client Test Project or New Client and Server Test Project. Give the Project a meaningful name in the Name field and then click OK.
- The Load Configuration window will open. Specify the number of client ports to use and some load parameters: Test Duration, Ramp Up/Down Duration, Load Profile type, and Load Profile value.
- The next step is Network Configuration. For each Client Port specify the base IP address and number (Count) to use, a subnet mask, whether a Gateway is required by specifying its address, whether you want to virtualize MAC addresses during Project execution by specifying a base address and the number of addresses to use (Count).
- Next select a Scenario Type. These Scenarios consist of a pre-defined set of Actions that are required for a basic test.
- The next step is the Scenario Configuration. Customize the most commonly modified parameters that relate to the Scenario Type selected in the last step. Modifications are made per port.
- The last step of the Wizard is a Configuration Summary. Click Finish to generate the Project

or to modify the configuration, click Back.

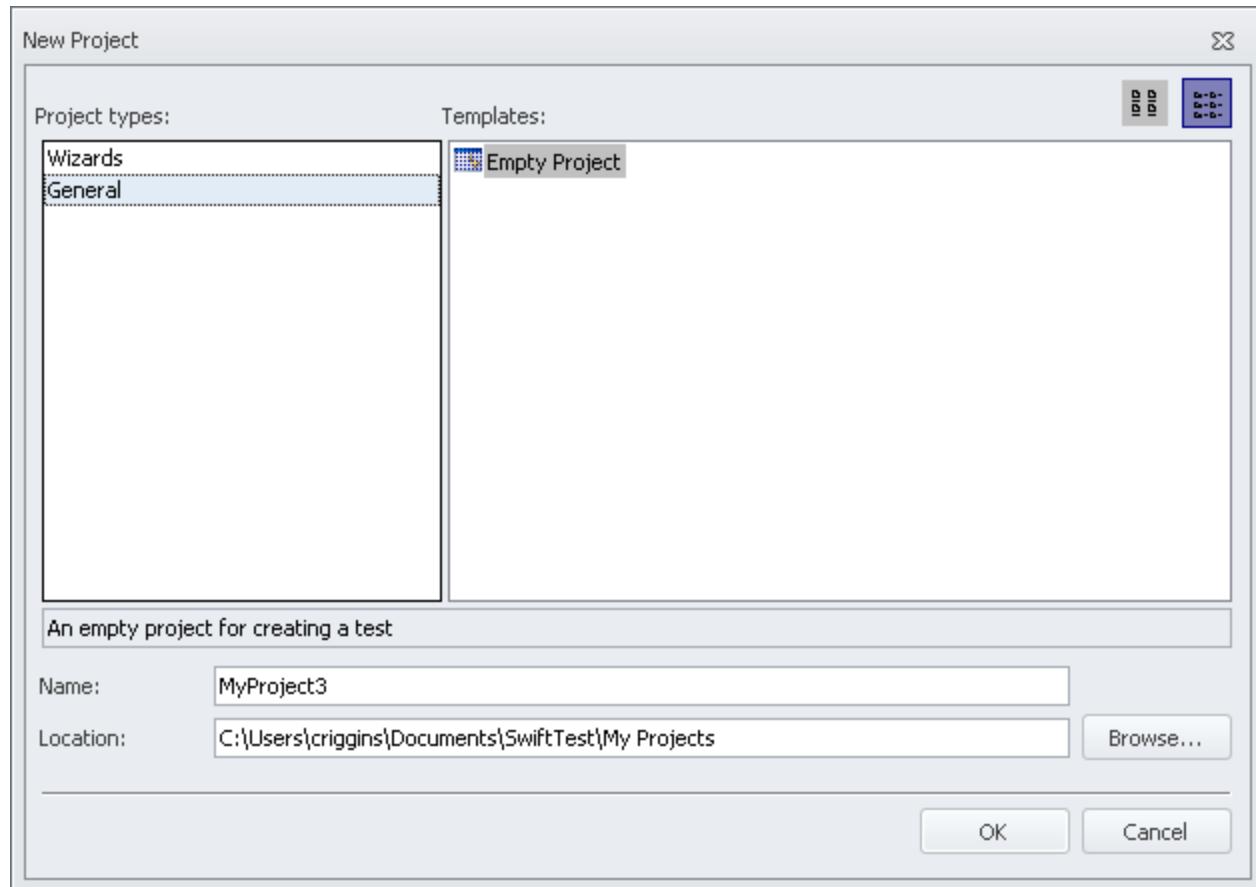
Once Physical Ports are assigned to Logical Ports, the Project is ready to run.

Create a Project Manually

Projects are created by assigning test components (Logical Ports and Resources Library items) to a Project Timeline.

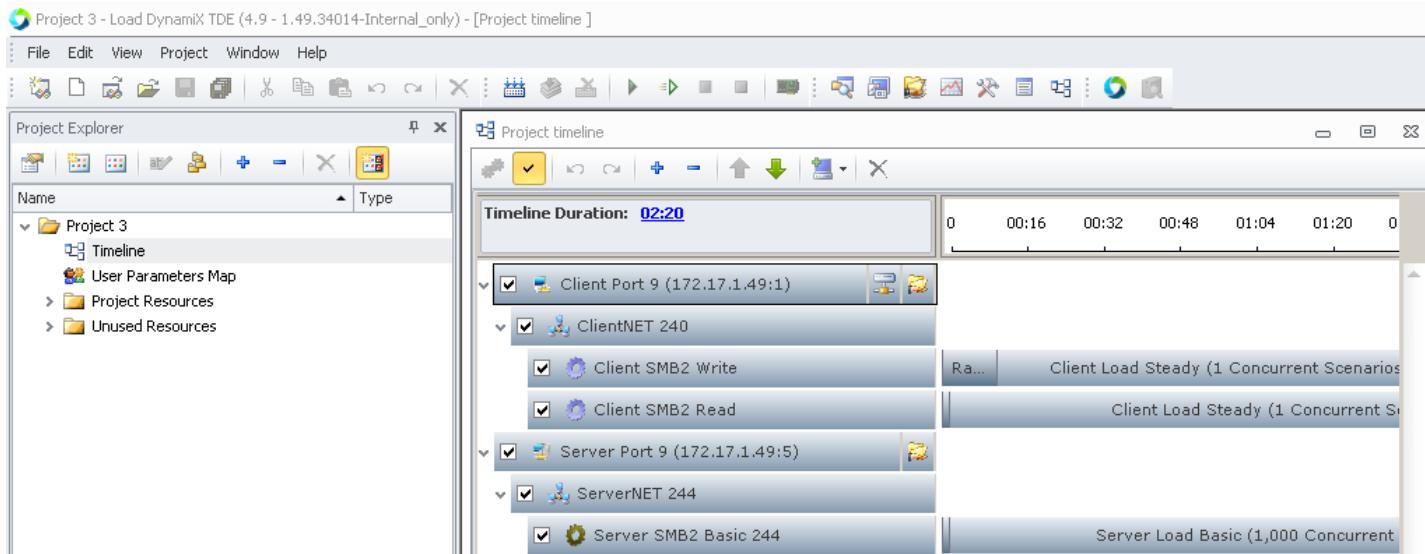
To create a new test

- Select File → New Project. The New Project dialog displays.



- Highlight General
- Enter a meaningful name in the Name field
- Click OK.

The Project Explorer window shows the newly created Project.



- Drag test components from the Resources Library into the Project Explorer window. Initially these Resources will be stored in the Unused Resources folder. Move them to the Timeline and they move into the Project Resources folder. Delete a Resource from the Timeline and they move from the Project Resources folder to the Unused Resources folder.
- All Projects must include a Network Profile, Load Profile, and Client and/or Server Scenario. The other test components are optional. These Resources are dragged (associated with) a Logical Port. All Projects must have at least one Logical Port which will be associated with a Physical Port on the Load DynamiX Appliance so that the Project can be executed. The Add Resources section later in this chapter provides details on adding individual Resources.
- Double-click Timeline to construct the Project Timeline. The Timeline window is where you drag-and-drop test components from the Resources folder into the Project Timeline. The order and content of the Timeline governs how the test executes.

Reading and Writing Data

The typical Load DynamiX Project will read and/or write data to or perform meta-data operations (logins, searches, etc) on a networked storage device or service. Reading and writing data with the Load DynamiX Appliances is accomplished through the use of Actions within a Scenario. A Scenario consists of one or more CIFS-SMB, NFS, iSCSI, HTTP, HTTPS, Kerberos, CDMI, OpenStack Swift, Amazon S3 or Scenario Control Actions for execution by the Appliance Client or Server software during a test. An instance of a Client Scenario emulates Actions taken by an end user application. Load DynamiX internal SCSI servers (the result of executing a Start SCSI Server Action) do not actually write the data written to them to disk.

Actions require input to perform their task (e.g. a file open Action requires a file name). While Action Input fields can be assigned explicitly as constants by the test developer, it is often more useful to have this information provided dynamically during run time so as to be able to vary the values used (e.g. one file name for all Scenario executions versus a unique file name for each Scenario execution). User Parameters provide that capability in the Load DynamiX TDE. References to User Parameter files can be assigned to Action input fields instead of constant values. See [Advanced Concepts: User Parameters](#) for a more detailed discussion of how and where to use User Parameters. See [Advanced Concepts: Data File Systems and Data Verification](#) for details of how to use Data File Systems and ::DataContent files in reads and writes.

Load DynamiX Action input fields that take Strings as input (e.g. file name, share name, folder name, user id, password, path name, events, variables, etc) accept the UTF8 character encoding so that the strings input to these fields can contain a broader set of characters. This is also true for Load DynamiX Resources such as Data File Systems , User Parameter files, Test Execution Rule

messages, etc. Two examples of the use of UTF8 strings:

SMB File Name



Variable content



Only a small subset of the Load DynamiX Actions are described in this section. The remainder of this Reading and Writing Data section of this chapter uses CIFS-SMB and Kerberos Actions as examples. The Load DynamiX Appliances also support SMB2, NFS v2, v3, v4, v4.1, pNFS, HTTP/HTTPS, iSCSI, CDMI and OpenStack Swift protocol-related Actions.

Authenticating Access to Servers

Before any test is able to open/create files or read/write data, the test must establish its credentials (security identity, permissions, etc) with the storage server it is interacting with. This process varies depending on which of the storage protocol is being used.

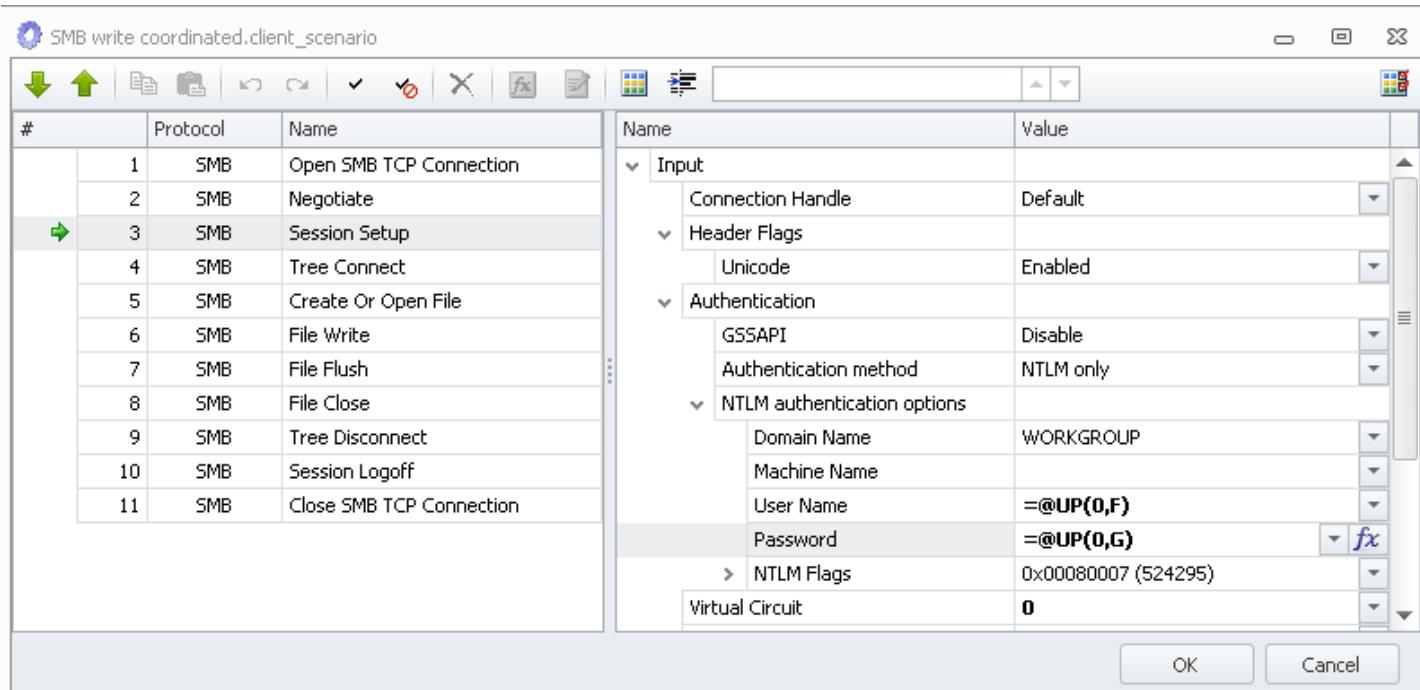
CIFS-SMB Authentication (links to protocol reference materials are provided in the [References and Terminology section](#).)

For CIFS-SMB Projects, the authentication process and user credentials are established using the **Negotiate** and **Session Setup** Actions. The **Negotiate** Action establishes whether NTLM Extended Security will be used.

In the screen shot above, NTLM Extended Security has been enabled. This means that the Scenario that contains this **Negotiate** Action will communicate credential information with the target storage server using NTLM (v2) Extended Security processes which is cryptographically stronger than NTLMv1 security. If this flag was Disabled, then the normal NTLMv1 security processes would be used. These processes determine how credential information is passed between Client and Server and how the information is to be processed.

The CIFS-SMB protocol also provides support for eliminating Man-In-The-Middle security breaches via the SMB Signing feature. The SMB signing feature adds digital signatures into CIFS-SMB packets to strengthen CIFS-SMB authentication. Load DynamiX TDE and Appliance software support CIFS-SMB Signing in the **Negotiate** Action Flags input field. The Packet Signing property Enables or Disables the SMB Signing feature. In the screen shot above, CIFS-SMB Signing (Packet singing) is Disabled. For more information about CIFS-SMB signing go to this site: <http://support.microsoft.com/kb/887429>

The **Session Setup** Action establishes the credentials. In this Action the User Identity and Password are delivered. In this Action, it is also determined if the Kerberos Authentication process is to be used. In the simplest case (no Kerberos), the **Session Setup** Action specifies NTLM only authentication and provides the User Identity and Password in User Name and Password fields.



There are many NTLM Authentication flags in the NTLM Authentication input field. This example uses the default settings. Once this Action is executed successfully, the CIFS-SMB server has accepted the credentials for the test and the storage-related permissions have been established.

If the test requires Kerberos Authentication, the following additional Actions would be required.

Kerberos Authentication

The Load DynamiX TDE and Appliance Client software support Kerberos authentication for CIFS-SMB, SMB2 and NFS (the Load DynamiX Start Server Actions (CIFS-SMB, SMB2, NFSv3, iSCSI, HTTP, HTTPS) do not support Kerberos). Kerberos provides authenticated communications between trusted hosts (clients and servers) that operate over insecure networks. Using Kerberos, every packet between the Client and Server contains a signature known only to the Client, Server and a Kerberos process called the Key Distribution Controller (KDC). The benefit is that only trusted hosts are allowed to communicate with each other, and that communications between the hosts cannot be corrupted by a Man-in-the-Middle attack.

See [Appendix: NFSv4 Notes: Locking, Kerberos & Delegation](#) and [Reference: Kerberos v5 Command List](#) for more details.

Without Kerberos, an administrator establishes user names, passwords and shares on a server or

domain. Once user names and passwords are configured, no other authentication is required for clients accessing shared information. Kerberos adds a third party, the KDC, to manage host authentication. The KDC manages communications within a realm, which typically is a domain (e.g. Load DynamiX.COM). Using this model, each host obtains a key that enables it to communicate with another specific host or application on that host. The KDC distributes keys to hosts. In a Windows environment, the KDC typically is a Windows Domain Controller (DC). It uses a database identifying all the hosts for which it has responsibility. Note that although the KDC and DC typically reside on the same server, Kerberos processing is independent of a user domain login.

The KDC has two server processes; the Authentication Server (AS) service and a Ticket Granting Server (TGS) service. The AS grants access to the Ticket Granting Server, via a Ticket Granting Ticket. A host then obtains a ticket for a specific application server from the Ticket Granting Service.

The Kerberos processing flow is as follows:

1. A client establishes a TCP connection to the KDC on TCP port 88.
2. The client authenticates to the KDC, and then obtains a Ticket Granting Ticket from the AS authorizing the client to obtain a Kerberos Ticket from the TGS.
3. The client then obtains a Kerberos Ticket from the TGS enabling it to communicate with a specific server.
4. The client then connects to the target server using the Kerberos Ticket issued in Step 3 above. No user name or password is involved in the request. Instead, the ticket authenticates the user on the target server.
5. The target server contacts the KDC to validate the ticket. If the ticket is valid, and the User has permission to access the service on the target server, the client connection request is granted.
6. The client then attempts to use the application on the target server. The user still must have access rights to the data on the target server. If not, access to the server is allowed but access to data is rejected.
7. The Kerberos ticket is valid for the time duration specified in the ticket, similar to a ticket on a metro bus where a rider can show the same ticket for a day. The User can authenticate against the server for the duration of the ticket lifetime.

The Load DynamiX appliance supports four client-side Kerberos Actions:

Open Kerberos Connection

AS-REQ

TGS-REQ

Close Kerberos Connection

NOTE: Windows servers generally require domain names to be in Upper Case only. Always use Upper Case when specifying domain names.

These four operations must complete before a user can begin protocol-level processing. Please refer to the sample test “CIFS-SMB Payload – W2K3 Server Kerberos” for an example of Kerberos processing flow.

Configuring Kerberos

Establishing Kerberos authentication involves several steps in the Load DynamiX TDE. Note that a KDC must be available to the test bed and that Kerberos client information must have already been established before a Load DynamiX client can communicate using Kerberos.

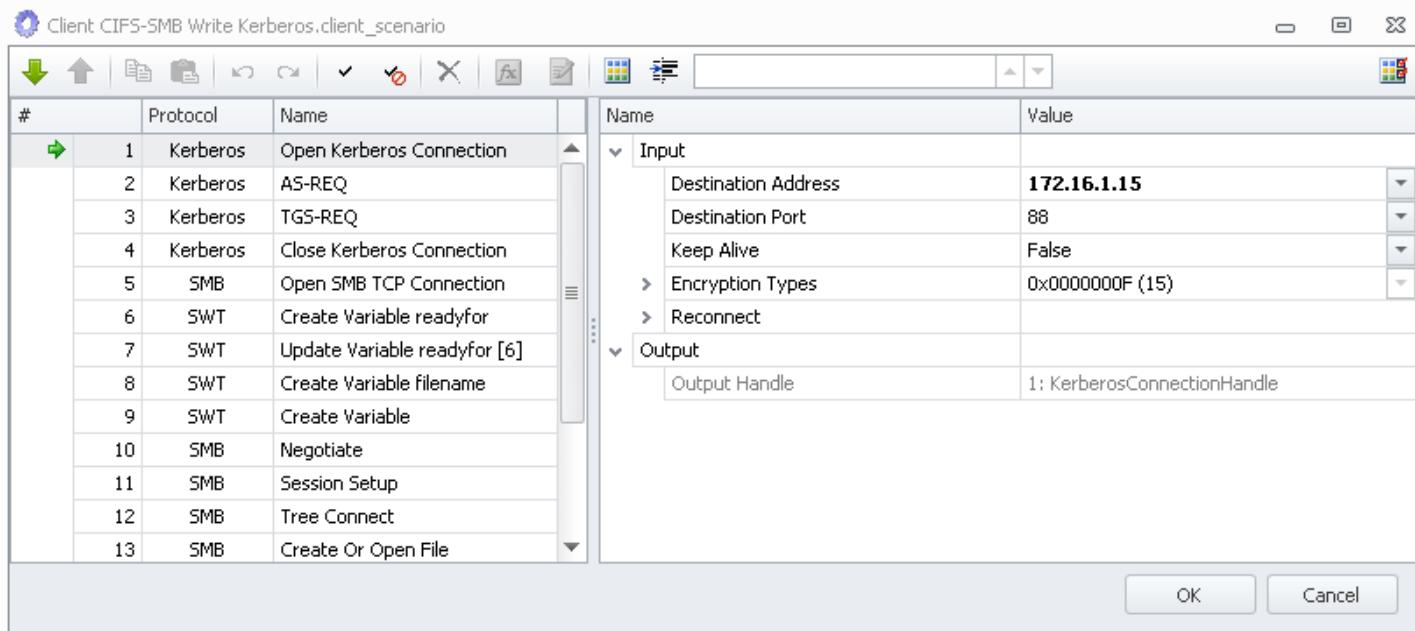
CAUTION: The appliance and server time settings must be within the maximum allowable time difference or Kerberos authentication will fail. By default, the maximum allowable time difference on a Windows Server is 5 minutes. To change the Time and Date on the Load DynamiX Appliance, see Setting Time and Date chapter in the Appliance-specific Quick Start Guide which is accessible in the TDE Help dropdown menu.

CIFS-SMB Actions

The Load DynamiX Actions used with Kerberos processing for CIFS-SMB are shown below.

The steps specify for the user to open a Kerberos connection, obtain an AS Ticket from the KDC, obtain a TGT ticket from the KDC and then use the ticket in the Session Setup Action. The steps required to configure Kerberos are as follows:

The **Open Kerberos Connection** Action opens a TCP connection to the Kerberos server (typically, the Domain Controller). Enter the IP address of the KDC. All other parameters are optional.



The **AS-REQ** Action (AS=Authentication Server) requests a Principal Ticket, which grants the right for the client to request privileges from the KDC. Required parameters include:

Principal: The user name or the host name and domain, e.g. USER001@STORAGE.QAD)

Password: The user's password

Service Principal: The ticket for the Kerberos Ticket Granting Service, e.g. krbtgt/STORAGE.QAD@STORAGE.QAD

NOTE: The following optional parameters are also specified in the command, but leave all these parameters set to False in this version unless you are doing conformance testing without requiring the client to take the specified Action. In the current release, the Load DynamiX appliance cannot automatically request ticket renewal or a new ticket unless the Actions are inserted to specifically request new tickets within a single Scenario. Note, however, that a ticket that expires during the execution of a Scenario does not cause a Scenario failure by itself. Even if a Kerberos ticket expires,

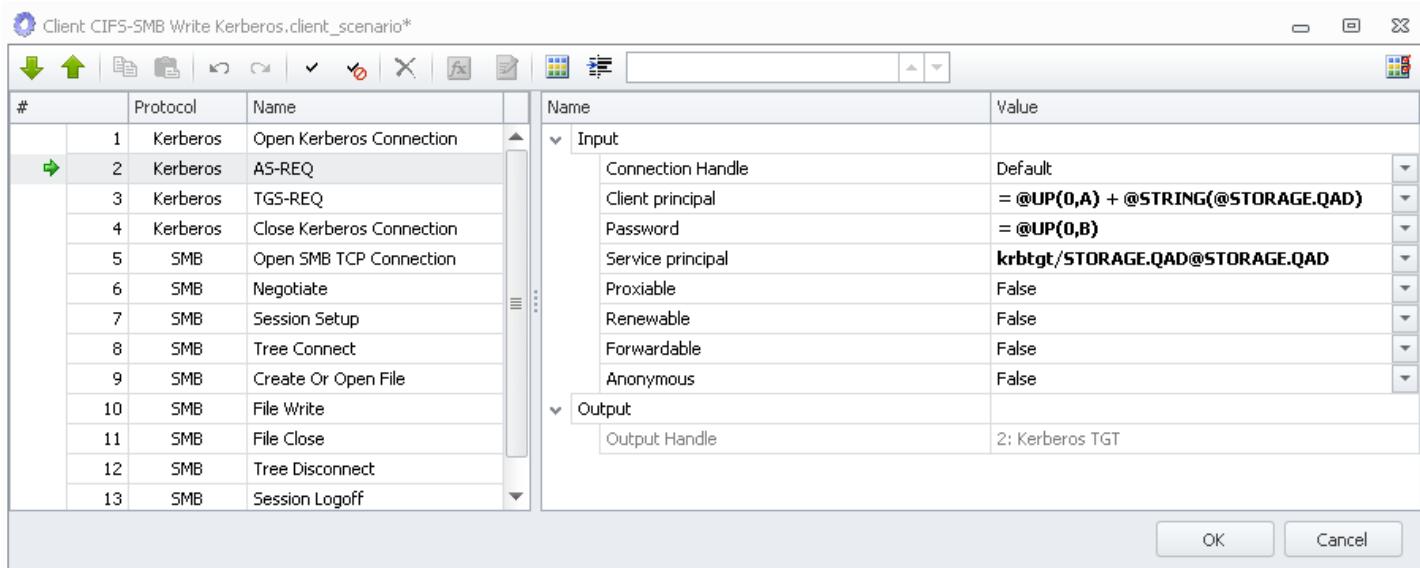
an open connection remains open and continues to operate until it closes. A ticket is used only to establish a connection and is not consulted during the time the connection is open.

The optional parameters are:

Proxiable: The specified user can delegate the ticket to other users.

Renewable: The specified user can renew the ticket when it expires without resupplying credentials (User Name and Password).

Forwardable: The specified user can use the same ticket to request access from another client system. If set to false, the ticket contains the specified host's source IP address and cannot be used on another host.

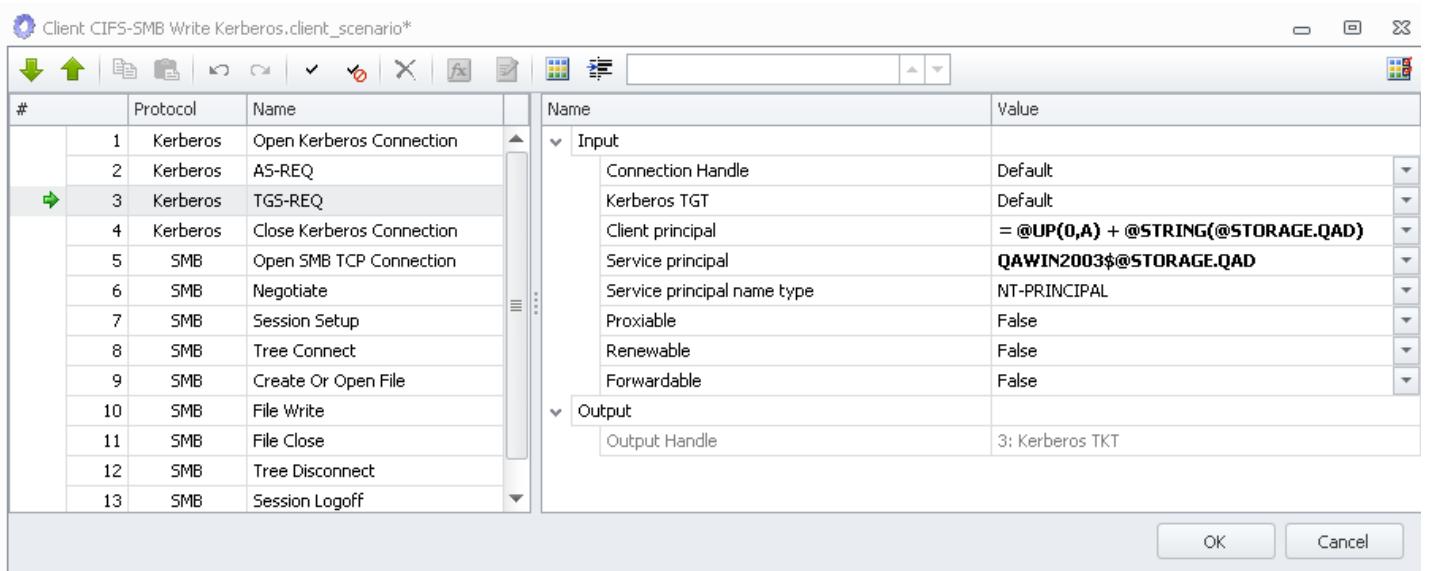


Anonymous – The request may be passed with no user name or password. If no user name or password is supplied, the request is evaluated strictly for anonymous access. If the user name and password are supplied, the request is passed and evaluated with login credentials. It is possible that one user may be allowed anonymous access, while another user may not. Anonymous can be used in lieu of all other parameters in this Action.

The **TGS-REQ** Action (TGS=Ticket Granting Server) requests a Kerberos Ticket (TKT) from the KDC.

Required parameters include:

Principal: The user name or the host name and domain, e.g. USER001@STORAGE.QAD)

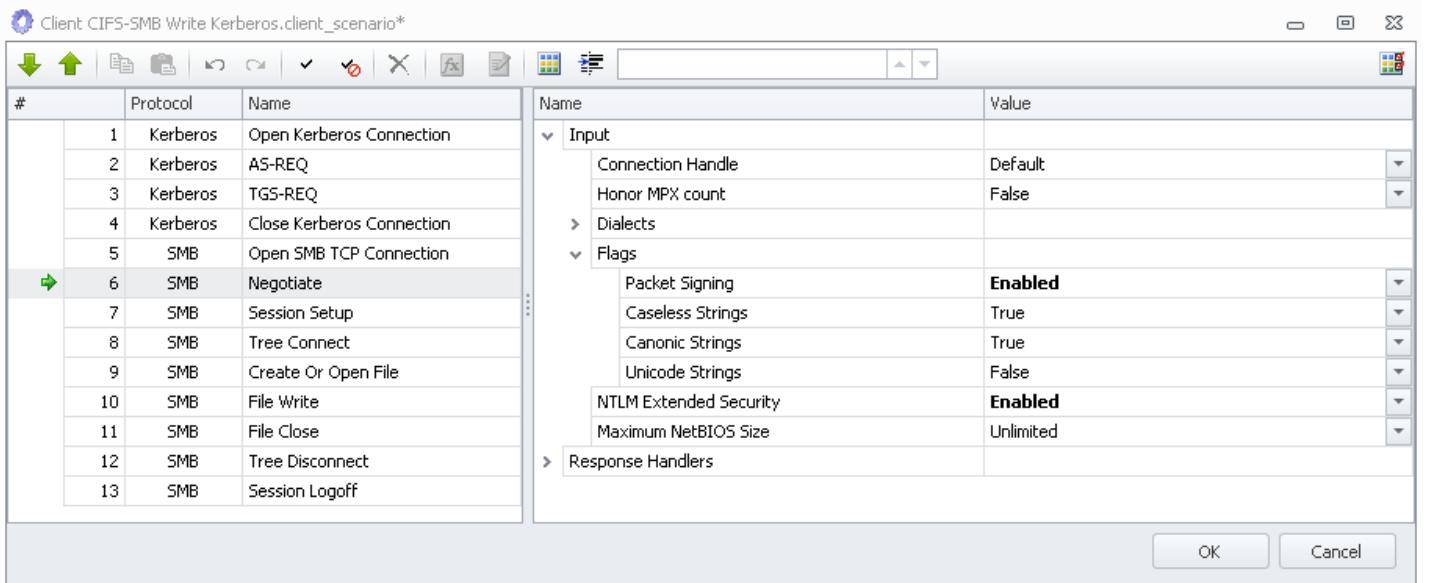


The Server_principal (in this case, the CIFS-SMB service on the server): The Kerberos Ticket Granting Ticket service, e.g. qa-adc1\$@STORAGE.QAD.

Note: The format for the server principal name is service-specific. Windows uses the format `servername$@REALM`. The format specified in the Kerberos standard is `{service}/{servername}@REALM`, where `{service}` is the service (e.g. SMB or NFS) running on the target server and `{servername}` is the name of the target server.

The **Close Kerberos Connection** Action and the Open SMB TCP/NetBIOS Connection Action has no Kerberos-specific parameters.

The Negotiate Action requires NTLM Extended Security to be set to Enabled. All other parameters may be set to default.



The **Session Setup** Action requests the type of authentication to be used. Required parameters include:

Use GSSAPI: Selects the authentication method:

If Disable is chosen, NTLM authentication must be used. Disable specifies that no GSSAPI encapsulation is to be performed, which renders Kerberos authentication unusable as it can operate only with GSSAPI.

If Enable is selected, an additional parameter, Authentication Method, displays below Use GSSAPI.

Authentication Method: Selects the order in which Authentication information is placed:

NTLM only: Selects and sends only NTLM authentication.

NTLM then Kerberos: Sends NTLM authentication followed by Kerberos in the same packet.

Kerberos then NTLM: Sends Kerberos authentication followed by NTLM in the same packet.

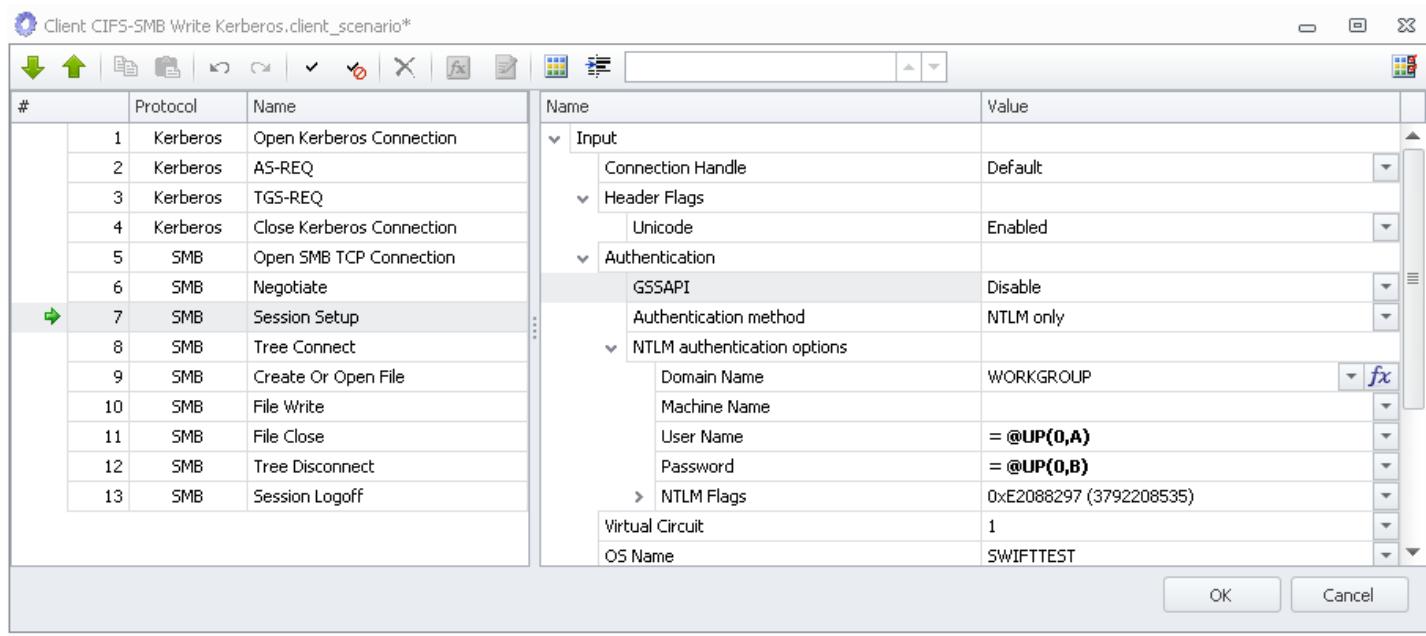
Kerberos only: Only Kerberos authentication is sent.

User Name: The user requesting the session

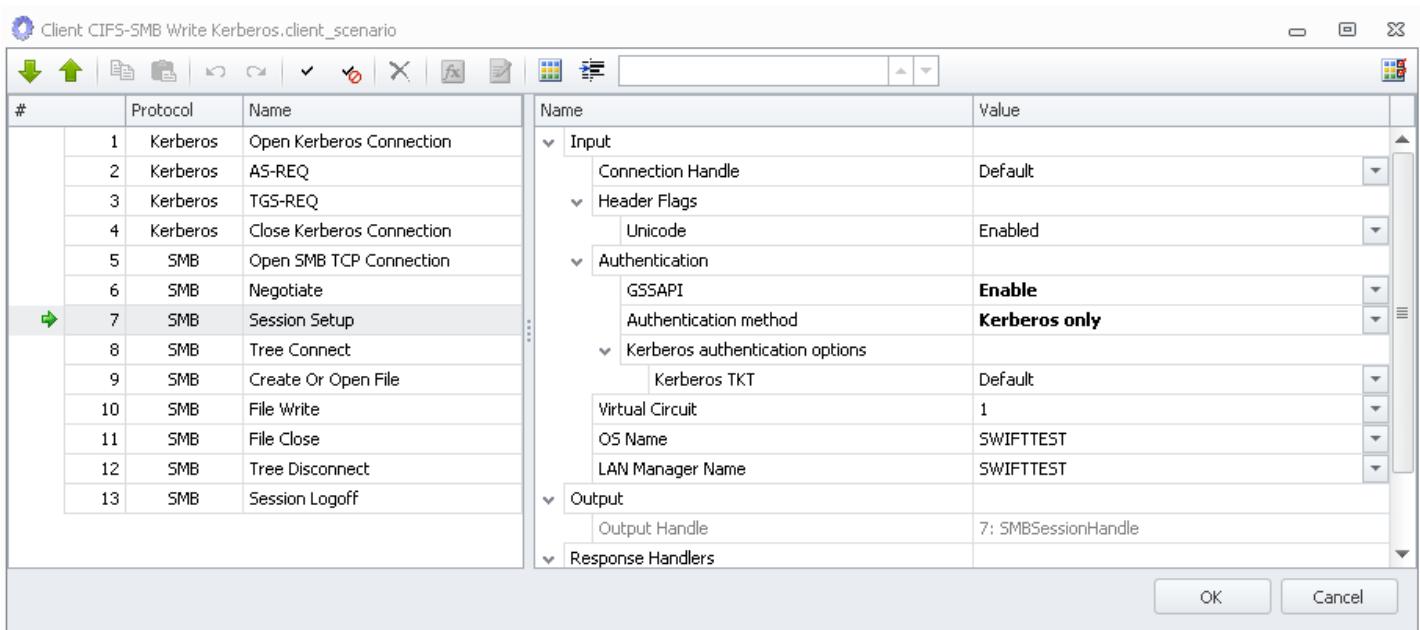
Password: The user's password

All other parameters are optional.

Session Setup without GSSAPI:



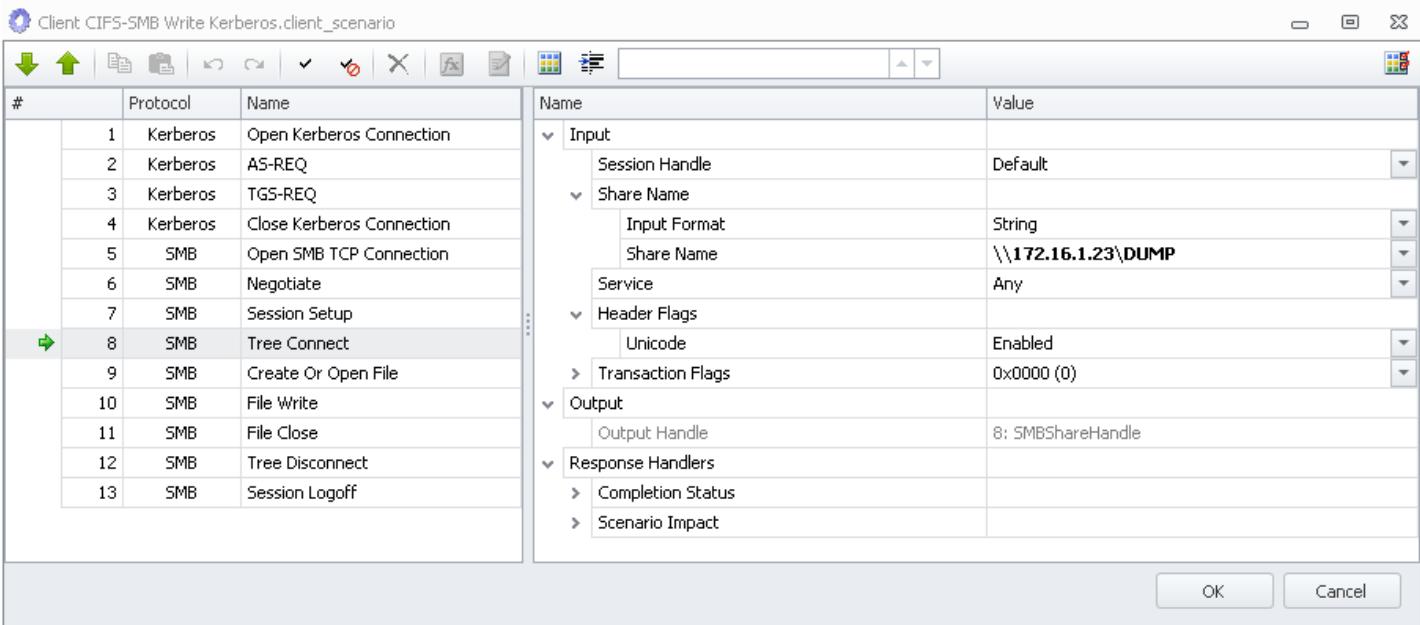
Session Setup with GSSAPI:



Writing Data

The CIFS-SMB **Create or Open File**, **Write File** and **Read File** Actions enable the Tester to create or select files for use during a Scenario and to Read from and Write to that file.

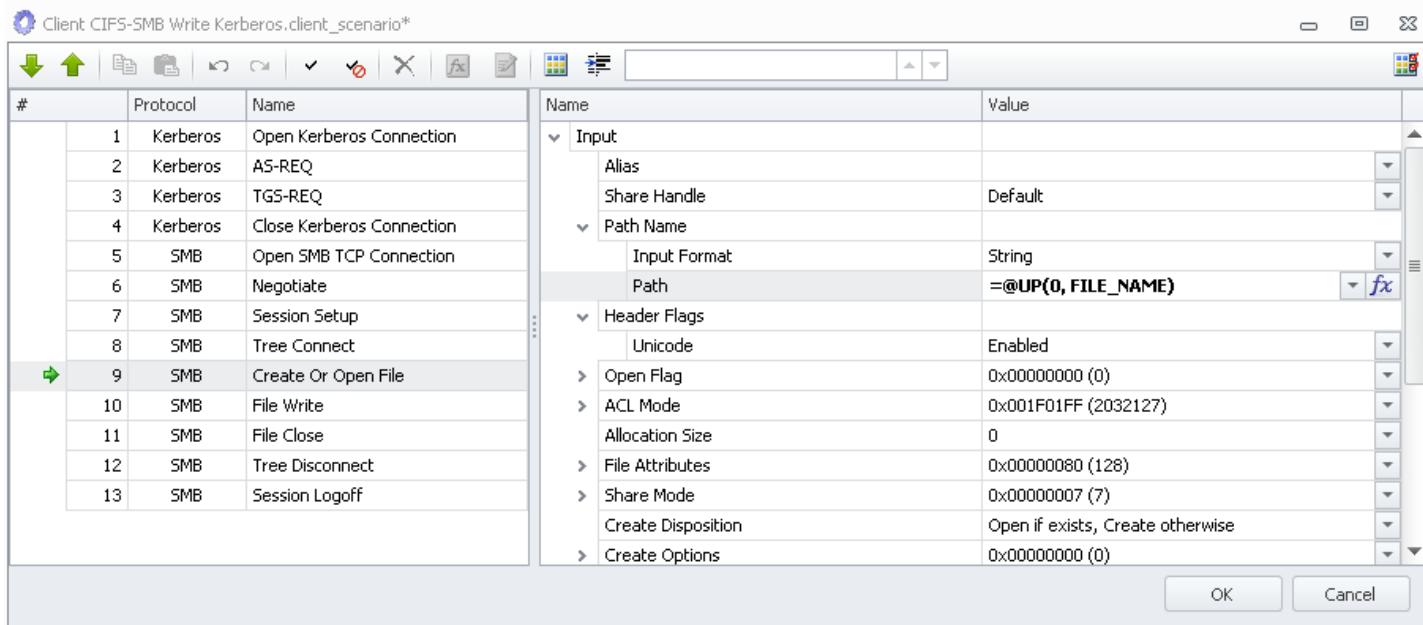
This Scenario shows the Actions within a Scenario whose purpose is to open and read a file.



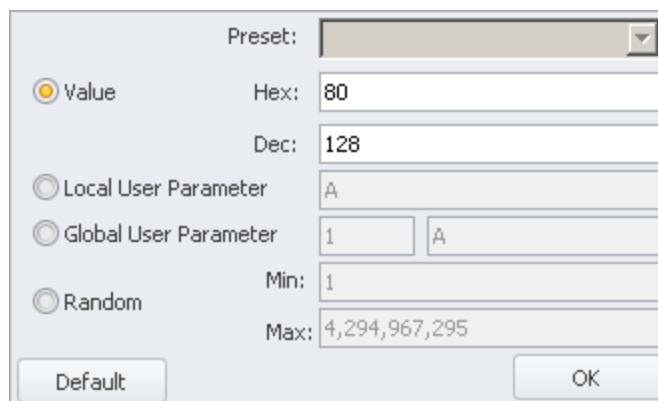
The first four Actions in this Scenario establish the Kerberos credentials which will be used later in the Session Setup to authenticate the user. The next three Actions (**Open SMB TCP Connection**, **Negotiate** and **Session Setup**) establish a connection with the CIFS-SMB server and log in as the user defined in the **Session Setup** Action. The next Action, **Tree Connect**, connects the test to a directory, in this case \\172.16.1.23\Dump. The final three Actions (**File Close**, **Tree Disconnect** and **Session Logoff**) disconnect the test from the CIFS-SMB server.

Creating a File

The CIFS-SMB **Create or Open File** Action enables you to specify a filename to be created. Action Input fields control what happens if a specified file already exists (e.g. Open the file if it exists or Create it if not, etc.). The screenshot below shows an example CIFS-SMB file creation using file name from a (Global) User Parameter file. See the [Advanced Concepts: User Parameters](#) section for more detail on the use of User Parameters.



Note in the above example, the presence of the Hexadecimal values showing next to several of the Create or Open File input fields. These fields are flags passed to the CIFS-SMB server as the Hexadecimal value shown when the Action is executed. These flag fields tend to have many selections with True or False values. To make the definition of these flag fields easier, the Load DynamiX TDE allows the flag settings to be specified as a Hexadecimal value directly on the line associated with the flag name (e.g. ACL Mode, Share Mode, File Attributes, etc). The Hexadecimal value of the input can be changed by manipulating the various components of the flag. Clicking the ">" to the left of a field opens that field and shows its component values. The Hexadecimal value can also be set directly by clicking on the value itself. When this is done, the Hex Editor feature is opened and looks like this



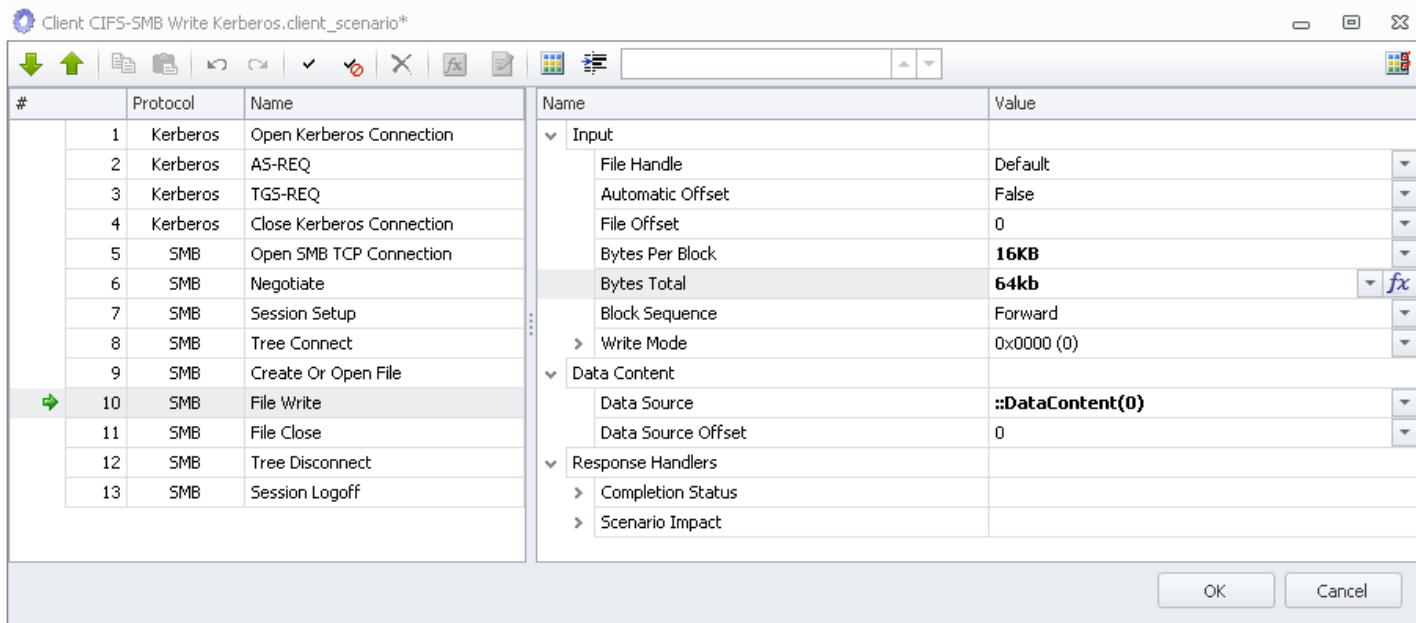
The Value of the field can be manipulated either as Hexadecimal or Decimal input. The input can also come from Local or Global User parameters or by creating a Random integer and using that.

In all cases, it is the responsibility of the Tester to understand the impact of any value inserted into an Action input field by the Hex Editor.

You can specify a full path or only a file name. If you specify only a file name and no directory as shown

above, the system assumes the current directory. When executed, the Action creates the specified file. Subsequent CIFS-SMB **File Write** Actions then write data to the file (see the screenshot below). The File Offset input field determines the byte position in the file where the Write operation begins (byte positions in a file range from 0 to N-1 for an N byte file). Automatic Offset allows the Tester to tell the Load DynamiX Appliance to manage the Offset values for each Write (or Read operation). Automatic Offset == False means that the Tester will manage the Offset value.

The **File Write** Action writes user or Load DynamiX-generated data to the file specified by the handle in the File Handle field. Notice the entry in the Data Source input field. It is a Resource from a Data File System that can contain Sequential data, Random data or user-specified data (e.g. a user-specified file that has been imported into the Data File System). For more details on the use of Data File Systems, see [Advanced Concepts: Data File Systems & Data Verification](#).



The File Offset, Bytes Per Block and Bytes Total fields in this Action (and many other Action input fields) can be specified in a shorthand format:

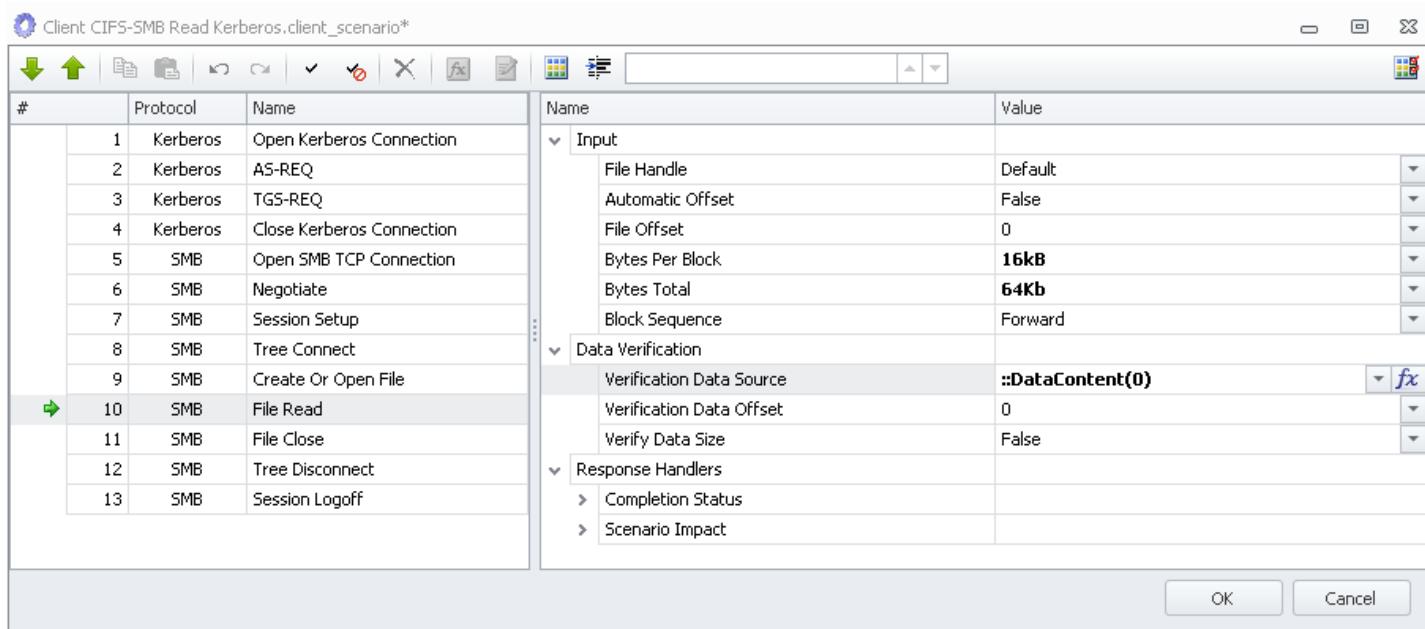
Shorthand	Value	Example
KB, kb, Kb, kB	1024	20kB = 20*1024
mb, MB, Mb, mB	1048576	10MB = 10 * 1048576
GB, gb, Gb, gB	1073741824	5gb = 5 * 1073741824
TB, tb, Tb, tB	1099511627776	700Tb = 700 * 1099511627776
K, k	1000	3K = 3*1000
M, m	10^6 (1000000)	13M = 13*1000000
G, g	10^9 (10000000000)	4G = 4* 10000000000
T, t	10^12 (100000000000000)	9t = 9*100000000000000

See the [Reference: Action Input Shorthand](#) for more details on the shorthand and also what happens when the shorthand is not allowed in a field.

Reading Data

The previous section described how to create a file and write data to it using CIFS-SMB Actions. To Read a file follows the same process (**Negotiate**, **Session Setup**, **Tree Connect**, **Create or Open File**) to get to the point of being ready to Read the file as well as the steps required to close down the Scenario (**File Close**, **Tree Disconnect** and **Session Logoff**).

This screenshot shows the **File Read** Action within a Scenario.



Once the file is open, you can read data from it. If the file exists on real storage server, the test Project will read the data from that file. If an internal Load DynamiX Server is used for the test, the server sends Load DynamiX-created data unless the file was created with user defined data earlier in the test.

See the [Advanced Concepts: Data File Systems and Data Verification section](#) for details about how to create entire file systems on an internal Load DynamiX server.

See the Reference sections for lists supported SMB, NFS, iSCSI, Kerberos, HTTP protocol Actions and Load DynamiX-defined Scenario Control Actions.

Test Execution Rules

Test Execution Rules allows a Tester to specify conditions to be monitored during the execution of Scenarios. If a condition of "Error" type is matched during the Scenario execution, the entire Project execution is immediately terminated, with a user-specified message. If a condition of "Warning" type is matched during the Scenario execution, then Scenario execution continues after logging a user-specified message into the Output window (or to STDOUT if the test is being executed using LdxCmd.exe). In the Load DynamiX environment, Projects continue to execute even if a Scenario fails for some reason. If it is desired to stop a Project immediately once any Scenario fails then Test Execution Rules are the means to do that. A Test Execution Rule of type "Error" and a condition of "Scenarios Failed >= 1" would cause the test to stop executing as soon as the first Scenario fails.

Test Execution rules take effect by being associated with (dragged and dropped onto) a Timeline resource. The Test Execution Rules and Conditions are evaluated for condition matching during Scenario execution via the TDE (see [Executing Tests and Assessing Results](#)) and Automation (see [Appendix: Load DynamiX Test Automation](#)) monitor for condition matching in the same way.

See [Advanced Concepts: Test Execution Rules](#) for a complete discussion of how to define and use Test Execution Rules.

Tracing Parameters

Tracing Parameters can be associated with Project execution by dragging them into the Project's Timeline and dropping them over preferred Port, Network, or Scenario components currently in

Timeline. Multiple instances of the same or different Tracing Parameters profile can be enabled in this way for different Timeline components. However each Timeline component may not contain more than one Tracing Parameters. When Tracing Parameters are dropped onto a Timeline component, a Trace Results file is created during execution and Tracing Parameters property values are used to how much trace information is captured for the component they were dropped onto. The Trace Results File (PCAP or CAP) is downloaded into the Results Explorer upon completion of the Project's execution or

when the Stop button  is pressed. Tracing Parameters files can be added to a Project by clicking on the Add New Item  icon and selecting the Tracing Parameters item.

The Trace (PCAP or CAP format) Results File contains the trace of all network packets that were sent or received via all Timeline components that have Trace Configuration Profile enabled, subject to Trace Configuration parameters as noted below. Displaying the Trace (PCAP) Results File requires the user to install Wireshark or another external viewer capable of parsing and displaying files in PCAP format. Upon installation of the external viewer, the Trace (PCAP) Results Files can be opened from within Load DynamiX GUI by double-clicking on them in the Results Explorer. Displaying the Trace (CAP) Results File requires the user to install Microsoft's NetMon or Message Analyzer applications. Tracing Results files with the .cap extension should be configured to be opened by either NetMon or Message Analyzer.

Tracing Parameters will capture UP TO the maximum length specified. If the Project run ends before the maximum length is reached, only those bytes are captured in the PCAP file.

Capturing and decoding SMB3.x.x encrypted packets

The encryption capabilities introduced into the SMB3 protocol by Microsoft has made packet capture and display by .PCAP based network capture tools like Wireshark unusable. To address the need to capture and visualize network traffic when using SMB3 protocol support on the Load DynamiX Appliances, support for the Microsoft .CAP format network tracing file has been added to the Tracing Resource. Selecting *.cap in the Tracing Options input field to generate a .cap format output file that can be interpreted by either NetMon or Message Analyzer. [Load DynamiX Support](#) can provide the details of how NetMon or Message Analyzer must be configured to interpret the SMB3 encrypted packets.

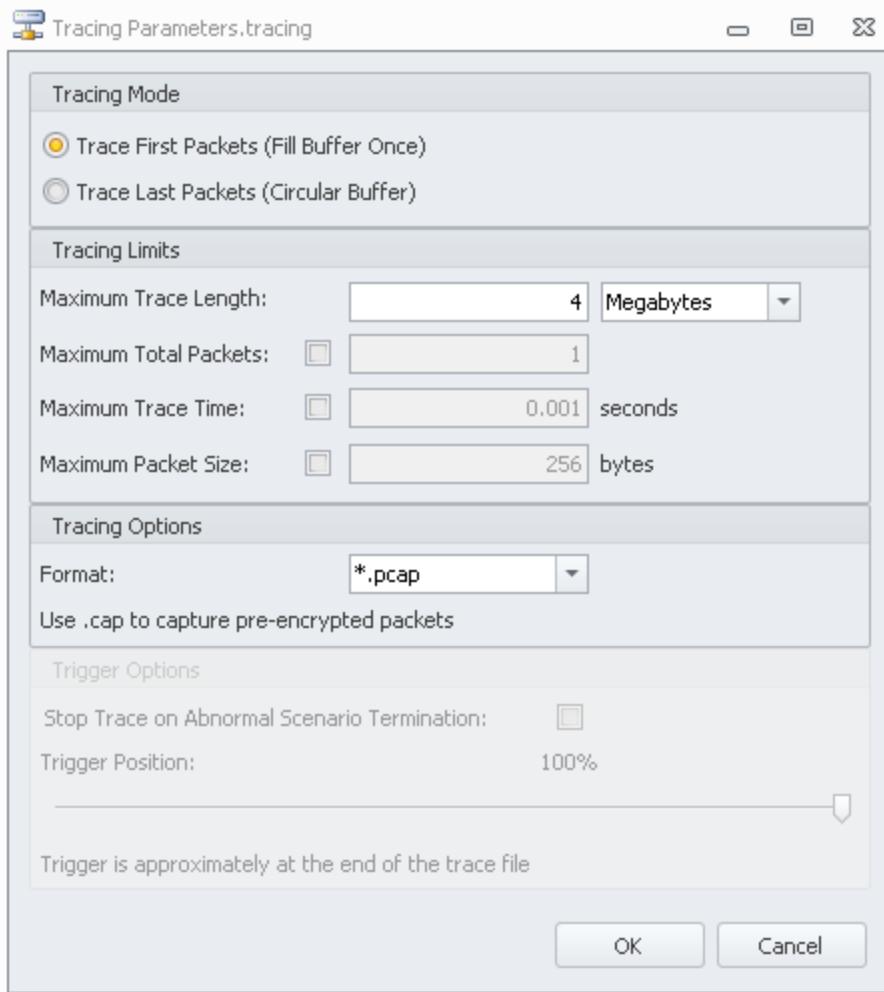
Tracing Resource

The Load DynamiX Appliance client and/or server software captures packets sent/received during execution in a trace buffer. The properties below determine how the packets are captured, how many will be captured, the format of the resulting file, etc.

Trace Configuration properties:

- Tracing Mode - The tracing buffer can be filled in one of two ways
 - Trace First Packets (default behavior) - the tracing buffer is filled starting at the beginning of the Project run until the buffer is full.
 - Trace Last Packets - a circular tracing buffer is implemented so that the packets at end of the Project run are captured.
- Tracing Limits - the properties that define how much to capture.
 - Trace Length - the size or length of the trace buffer (the maximum size of the Trace Results file). Max Tracing Length is 1GB in Trace First Packets mode, 256MB in Trace Last Packets mode.
 - Maximum Total Packets - constrains the capture buffer contents by setting the maximum number of packets that will be captured

- Maximum Trace Time - constrains the capture buffer contents by setting the maximum time for the capture interval.
- Maximum Packet Size - specifies how many bytes of each packet captured are saved in the trace buffer.
- Tracing Options - .PCAP or .CAP format output Trace Results file
 - *.pcap for PCAP format output file (Wireshark compatible)
 - *.cap for CAP format output file (MS NetMon or Message Analyzer compatible)
- Trigger Options (only applicable to Trace Last Packets mode) - Properties that determine how packet capture behaves when Scenarios abort).
 - Stop Trace on Abnormal Scenario Termination - stops packet capture when Scenarios terminate abnormally (captures the packets that were involved in the Scenario termination).
 - Trigger Position (only enabled when Stop Trace on Abnormal Scenario Termination is enabled) - allows the tester to specify approximately how many bytes of packet data are in the trace buffer following the first Action that causes a Scenario to terminate. A 50% setting allows for up to 50% of the maximum length bytes of packet capture after the terminating Action. A 1% setting allows for up to 99% of the maximum length bytes of packet capture after the terminating Action. A 100% setting places the packet at the end of the trace buffer (0% packet capture after the terminating Action). If the maximum length of the tracing buffer has been filled before the terminating Action occurs then the % settings indicate approximately where in the Trace Results file the terminating Action will be found. If the maximum length of the tracing buffer has NOT been filled before the terminating Action occurs then there is no guarantee of where in the Trace Results file the terminating Action will be found. In the example below, the maximum length is 100KB and the Trigger Position is 75%. If less than 100KB of packet data is captured and a Scenario terminating Action occurs, the PCAP file will have at most 25KB of packet capture after the terminating Action.



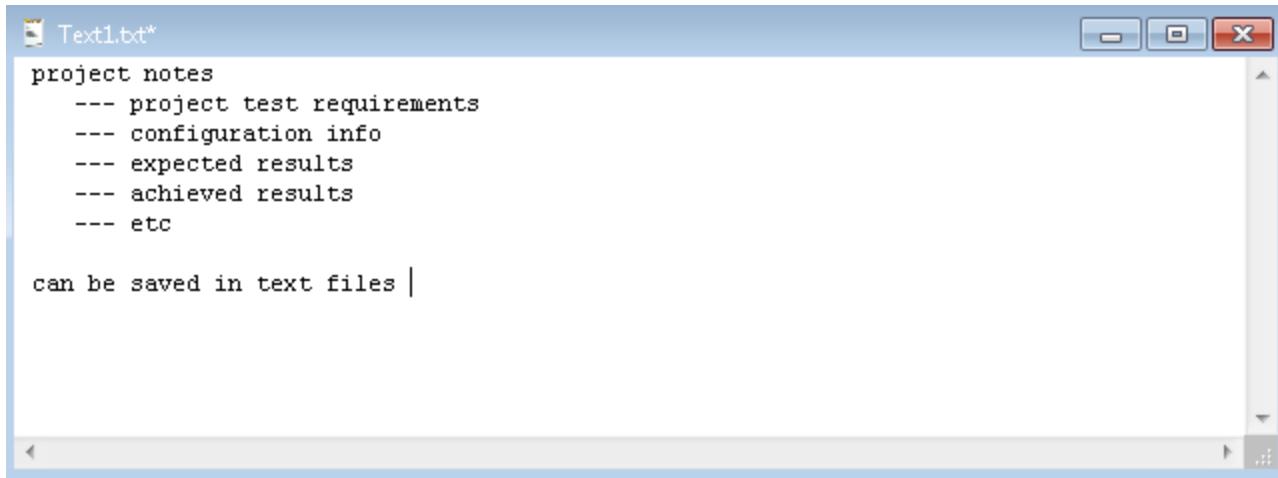
NOTES:

1. Tracing Parameters are not a performance tool. Using Tracing Parameters to capture PCAP data will reduce performance in the Load DynamiX Client and Server software.
2. Because packets sent by the Load DynamiX Appliance are stored in the Trace Results file before they are given to the network hardware to transmit and the Appliance lets the network hardware calculate packet checksums, these packets will show up in PCAP viewers such as Wireshark as having invalid checksums. This issue can be ignored.
3. Dropping Tracing Resource files that contain different Tracing Options settings (e.g. one with .cap selected and one with .pcap selected) onto the same Project will generate an error message.
4. Trying to open a .cap format Trace Results file with Wireshark or other PCAP based network tracing tools is likely to result in an error.

Add Project Notes

The application provides you the ability to create a text file containing notes about the Project. When you choose to add a text file, a window opens for you to enter the text. The file is not used for any actual

processing. Project Notes files can be added to a Project by clicking on the Add New Item  icon and selecting the Text item.



```
project notes
--- project test requirements
--- configuration info
--- expected results
--- achieved results
--- etc

can be saved in text files |
```

Resource Explorer

The Resource Explorer provides the Tester with a folder, named Load DynamiX Resources, of Resource templates that can be used to facilitate Project development.

Resources Library	
Name	Type
> My Resources	
< Sample Resources	
> Advanced Load Profile Samples	
Sample Client Scenario	Client Scenario
Sample File System	Data File System
Sample Load Profile	Load Profile
Sample Network Profile	Network Profile
Sample Server Scenario	Server Scenario
Sample Test Exec Rules Errors	Test Execution Rules
Sample Test Exec Rules Notify	Test Execution Rules
Sample Test Exec Rules Warning	Test Execution Rules
Sample User Parameters	User Parameters

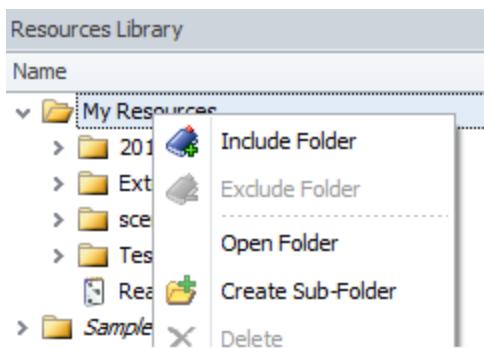
Many of these templates are shells that need Tester input to be of any value to a Project. For example, the Sample Network Profile needs IPv4 or v6 address information or TCP information. The Sample Client Scenario requires protocol-specific Actions to be useful. A collection of Advanced Load Profile Resources is provided in the folder named Advanced Load Profile Samples. These Resources produce specific load patterns such as a Square Wave or a Sawtooth Wave. These Resources are more ready to use, requiring minimal Tester input to be Project ready.

Name	Type
> My Resources	
< Sample Resources	
< Advanced Load Profile Samples	
Broken Sawtooth	Advanced Load Profile
Broken Triangle	Advanced Load Profile
Increasing Broken Sawtooth	Advanced Load Profile
Increasing Broken Triangle	Advanced Load Profile
Increasing Sampled Broken Sawtooth	Advanced Load Profile
Increasing Sampled Triangle	Advanced Load Profile
Increasing Sawtooth	Advanced Load Profile
Increasing Square	Advanced Load Profile
Increasing Triangle	Advanced Load Profile
Sample Advanced Load Profile	Advanced Load Profile
Sampled Broken Sawtooth	Advanced Load Profile
Sampled Triangle	Advanced Load Profile
Sawtooth	Advanced Load Profile
Square	Advanced Load Profile
Triangle	Advanced Load Profile

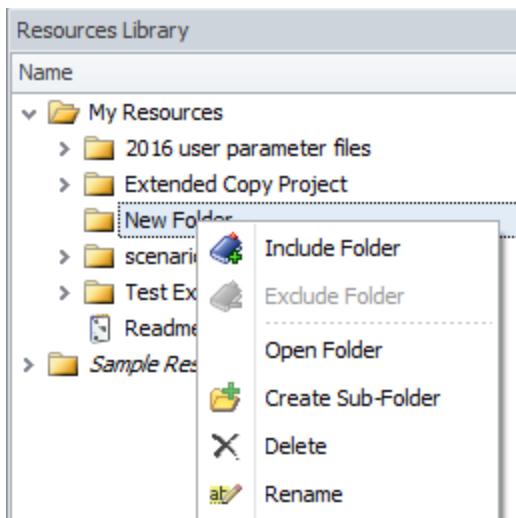
The Resource Explorer also provides a folder, named My Resources, into which the Tester can save resources that are to be shared amongst multiple Projects. This folder support Tester-added sub-folders so that the Tester can organize any useful Resource files in this folder by Resource type or Protocol type or Project name or whatever categorizing methodology the Tester prefers.

Name	Type
< My Resources	
< 2016 user parameter files	
server IP 2016	User Parameters
Server password 2016	User Parameters
< Extended Copy Project	
Initiator I Load	Load Profile
iscsi Extended Copy	Network Profile
iscsi VAAI Extended Copy (1)	Client Scenario
< scenarios	
Client SMB2 Conn Rate 260	Client Scenario
Server SMB2 Basic 264	Server Scenario
< Test Execution Rules	
Sample Test Exec Rules Notify extra	Test Execution Rules

Simply drag the Resource to be shared into the My Resources folder or into a Tester-defined sub-folder and then drag it from there into any Project that requires it. My Resource sub-folders are created by right-clicking on My Resources and selecting the Create Sub-Folder menu item.



The default name of the new sub-folder is New Folder. It can be renamed during the create process by highlighting New Folder and typing in a new name, or it can be renamed later by selecting New Folder in the sub-folder list, right clicking and selecting the Rename menu item.



Executing Tests and Assessing Results

Executing Tests and Assessing Results

This chapter describes how to execute a test and assess the results.

Assigning Physical Ports to Logical Ports

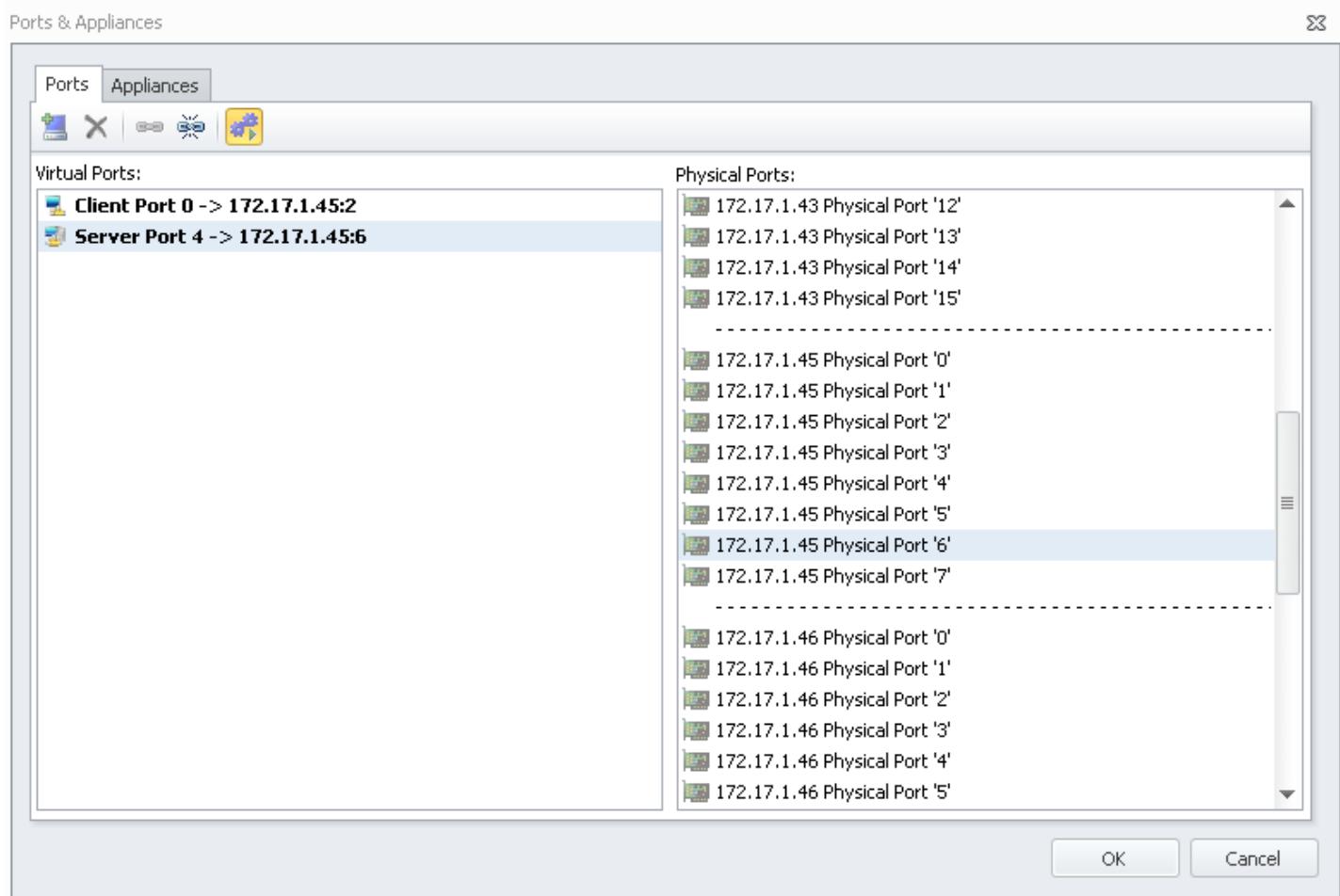
Before a Project can be compiled and executed, its Logical Ports must be assigned Physical Ports to use. This is done using the Ports and Appliances button on the main toolbar. Click the Ports &

Appliances button:



The Ports & Appliances window allows you to manage the linkage between the Logical Ports in the software and the physical ports on the appliance and to manage various aspects of the Appliance. The Ports tab is where Logical Port-to-Physical Port assignment is performed. Dragging a Physical Port from the right portion of the tab onto the Logical Port on the left side of the tab accomplishes the necessary assignment.

Ports tab



In addition, this window allows you to manage the Physical and Logical Ports in your configuration.

Logical Ports can be added or deleted : 

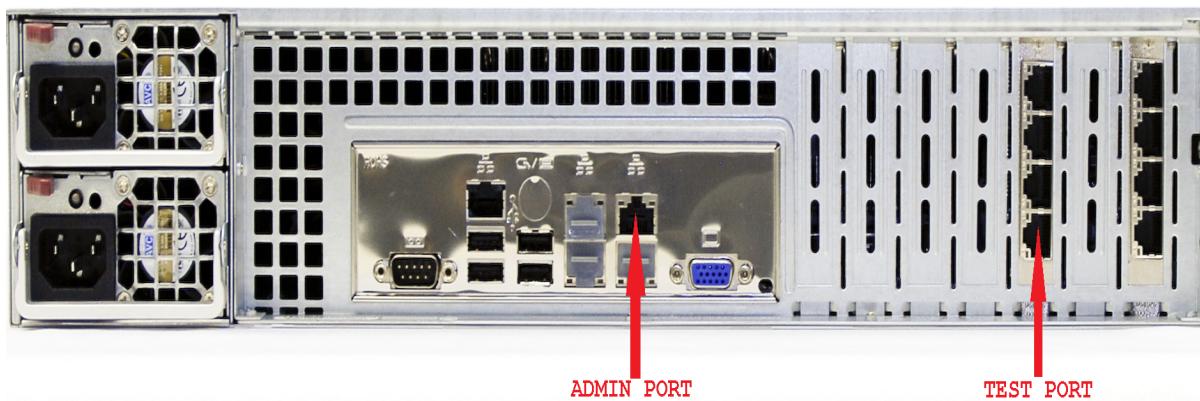
The links between Real and Logical Ports can be created or broken : 

The Project Ports Only button  allows you to toggle between showing just the Logical Ports that are applicable to the current Project and all Logical Ports.

A right-click on any of the Client or Server Logical Ports shown in this window provides the same set operations as the toolbar (Add/Delete Logical Port, Link Port/Clear Link and Project Ports Only On/Off) plus the ability to Rename  a Logical Port name to be something more relevant to its use.

The Logical Ports on the left hand side are the available ports for test development, any port that you have in your Timeline will be highlighted in bold. The Physical Test Ports on the right hand side are the ports that are available on the appliance.

If the Appliance used in a test were a Load DynamiX 1G Series Model 3000 or 3108, the Physical Ports that would be available would be as shown below:



Appliances tab

Ports & Appliances

Ports Appliances

Ping Stop Port Ports Status Link Status Clear Content Update Reboot

Port Details Rediscover Targets Clear NPIV Licenses

Physical FC Port 1	Link Speed	16 Gbps
172.17.74.10	BB Credit	5
Physical FC Port 0	Initiators	1
Physical FC Port 1	Initiator WWPN	21:00:00:0e:1e:14:b2:81
Physical FC Port 2	FCID	0x10600
Physical FC Port 3	Targets	8
Physical Port 4	Targets	8
Physical Port 5	Target WWPN	21:00:00:0e:1e:07:02:f8
Physical Port 6	FCID	0x10300
Physical Port 7	LUN(s)	0-4
172.17.74.11	Lun	0
Physical FC Port 0	Size	8192.00 MB
Physical FC Port 1	Lun	1
Physical FC Port 2	Size	10240.00 MB
Physical FC Port 3	Lun	2
Physical FC Port 4	Size	10240.00 MB
Physical FC Port 5	Lun	3
Physical FC Port 6	Size	10240.00 MB
Physical FC Port 7	Lun	4
	Size	100.00 MB

OK Cancel

Note: in the Appliances tab window above, the physical ports on a Load DynamiX 1G Series Model 3000/3108 Appliance are the green icons and the physical ports on a Load DynamiX 10G Series Model 5000/5102/5108S/5108T Appliance are the blue icons . Physical Fibre Channel Ports of the 6202/6204/6208 Appliances look like . Icons for the Unified Series appliances are a mixture of the FC Series and the 10G Series . Fibre Channel over Ethernet ports (model 6202E) use the icon.

This window allows the user to manage your appliances. Here you can:

Add and Remove Appliances that are available to you :

Refresh Appliance Firmware version information on the screen :

Ping the Appliance :

Stop a running Port :

Check the port status (Ports idle or in use) :

Test Link status (Ports connected) :

Clear Content (remove any filesystem content downloaded to the Appliance) :

Update the Appliance's Firmware or System :

Reboot the Appliance :

Port Details (display Fibre Channel port details):

Rediscover Targets (rediscover this Appliance's Fibre Channel targets):

Clear NPIV (remove all NPIV initiator and target WWPNs from a Fibre Channel port) :

Licenses: Activate or Add Licenses to an Appliance:

All operations (other than Ping) require HTTP access between the TDE and the Appliance. If operations from this window other than Ping do not work, a simple test of HTTP access is to open a browser window on the WorkStation system and type the IP address of the Appliance into the browser's address bar. If the browser window displays the text "Load DynamiX" then the HTTP connection is operating correctly and some other issue may be impacting access to the Appliance.

Selecting Ping Appliance, Reboot Appliance, Port Status, Test Link(s), or Update will open a new dialog box. This dialog box will have a status window, run, cancel, and close buttons as they relate to the tool. The Update tool also has an address bar for defining the location of the new Firmware file (*.tgz). Ping, Port Status and Test Link(s) run immediately on selection. Update Firmware and Reboot commands require additional user interaction before they execute.

The Port Details, Rediscover Targets and Clear NPIV functions only operate on Fibre Channel Ports. Port Details provides information regarding Initiator and Target WWPNs and more LUN details such as vendor, size and logical number (0-n). Rediscover Targets forces the Appliance Fibre Channel Firmware to re-discover all fibre channel targets connected to the Port. Clear NPIV forces the Appliance Fibre Channel Firmware to remove all NPIV Initiators and Targets associated with the Port. See [Reference FC/SCSI/iSCSI Commands and Behaviors](#) for more details.

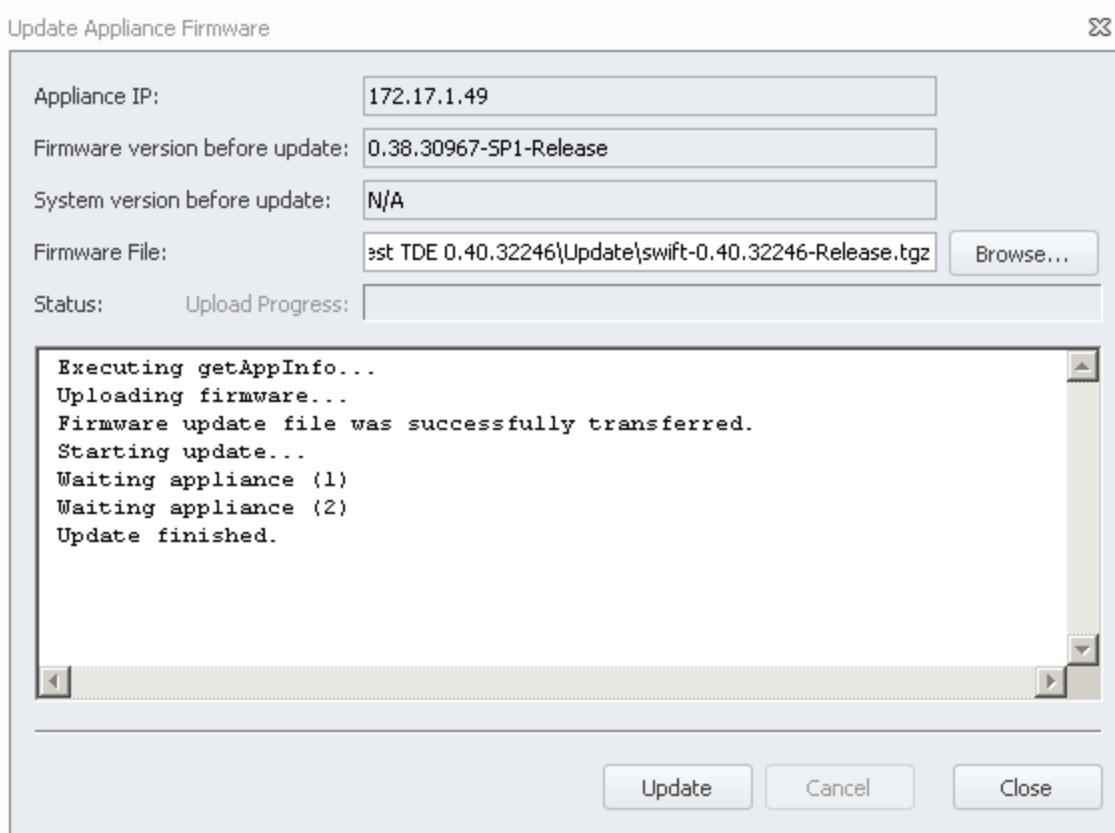
The Test Link(s) button gives specific information regarding one or more ports on an Appliance. This interface will tell you if the port is connected to a network and the port speed, and communications mode. To select more than one port at a time hold Ctrl while you select the ports.

Each release of the Test Development Environment includes the most current version of the Appliance Firmware. This Firmware often includes new features and enhancements.

To Update the Firmware running on an Appliance to the Firmware delivered with the Test Development Environment, follow this process:

- Highlight the Appliance to be updated
- Click the Update Appliance Firmware button 
- The Appliance Firmware (swift.tgz) file associated with this release should appear in the Firmware File window
- Click Update and the new Firmware will be downloaded to the selected Appliance
- When the Update Appliance Firmware process is complete, click the Close button
- Click the Reboot Appliance button: 
- Click Reboot and when the Reboot Appliance process completes, click Close

The Appliance is now ready for use.



Appliance System Update

In the rare case that the underlying software system needs to be updated on the LDX Appliance, customers will be informed by Load DynamiX Support, other LDX representatives or authorized 3rd-party representatives, that a System Update is required.

The System Update provides fixes, improvements and/or updates to components that are part of the LDX Appliance underlying operating system, but are not directly responsible for LDX Appliance features. Therefore, updating these components requires an Update process that is different from the normal Firmware Update that a user would typically perform on each new release of the LDX Software.

To perform System Update, use the Firmware Update button in the Ports & Appliances window. When prompted, select the System Update file, instead of the normal Firmware file, and click Update to begin the System Update process.

Once the System Update process completes, a reboot may be required. After that, the LDX Appliance is ready to use.

System Update is functionality added in v4.0 TDE and Appliance Firmware, and it supports the following Appliances:

- LDX 1G Series 31xx
- LDX 10G Series 51xx
- LDX FC Series 62xx
- LDX FCoE Series 62xxE
- LDX Unified Series U10xx
- LDX Enterprise Series E10xx (on the load generating sub-appliance only)
- LDX Virtual Series V1000

Prior to v4.0, a System Update file will not be recognized by the Firmware Update function.

Compilation



A constructed Project Timeline is compiled using the Compile Project button located on the toolbar or by pressing the F7 key. It is not necessary for the Timeline window to be open when compiling the test – compilation results will display in the Output window.

Shown below is a sample compilation output, including error messages. The compile does not stop on encountering the first error, but attempts to compile the entire test to alert you to all existing errors.

```
Output
compiling project: C:\Users\SwiftTest\Documents\SwiftTest\My Projects\R30 Samples\CIFS-SMB\CIFS-SMB FD Payload - W2K3 Server Kerberos\CIFS-SMB FD Payload - W2K3 Server Kerberos

verifying project...
creating documents list...
compiling...
Compiling documents:
 50 Users Passwords 24 Files.user
Compiling documents:
  Client Port 0.client_port
  Tracing Parameters.tracing
  24 Data Files.dfs
  Client.NET 240.network
  Client CIFS-SMB Write Kerberos.client_scenario
  Client Load Steady.load
  Client CIFS-SMB Read Kerberos.client_scenario
verifying compiled project...
linking...
creating object files...

compile complete - 0 errors, 0 warnings
```

You can stop a compilation by clicking the Cancel button (located to the right of the Compile Project button) or pressing the F6 key.

Compilation also occurs when the test is executed using the Start Test (F5) button (see next section).



You also have the option of compiling a single test component using the Compile Document button located next to the Compile Project button or by pressing the Ctrl + F7 keys. Selecting the test component you wish to compile and clicking Compile Document allows you to spot-check a test while

composing it. The Compile Document function is only enabled when the Scenario or Network Profile windows are open.

Execution

A Project is executed by pressing the Start Test (F5) button  located on the toolbar or by pressing the F5 key. The TDE compiles the Project and, if successful, executes it. During execution, the Project

logs information to the Output window and a Results Folder. The Stop  and Abort  buttons will Stop or Abort the Project, respectively. A Stopped Project retains all statistics accumulated during the execution, will generate log files, and, if Tracing Parameters are present, PCAP files. An Aborted Project retains any statistics collected up to the Abort but will not generate PCAP or log files. Project execution can be additionally controlled by Test Execution Rules described in the [Advanced Concepts: Test Execution Rules](#).

Load DynamiX Appliance Test Ports are only active during test execution so Pinging them prior to running a test will not produce a positive response. If it is necessary to Ping the Appliance Test Ports during test execution to verify that the Test Port is active, please limit the frequency and size of Ping requests as it can impact Test Port performance during test execution. Load DynamiX suggests setting the Ping packet size to less than 32 bytes (-s option on the linux/unix ping command or -l option on the Windows ping command).

When the TDE downloads a Project to the Appliance for execution and tells the Appliance to execute a Project, the Load DynamiX Appliance starts gathering statistics from the Project on a roughly 1/2 second interval. The TDE will request the Appliance to upload statistics updates approximately every 1 second. At the end of the execution of the Project, a final set of statistics will be uploaded to the TDE which also contains information that tells the TDE that the Project has completed. When the TDE receives the "end of project" indication, it downloads the log, pcap, data verification information and other files necessary to complete the execution.

Statistics updates are uploaded to the TDE using the HTTP protocol and if for some reason the TDE is not able to receive an update, the update is lost but because of the short update cycle, little if any information is lost. The most likely cause of this situation is because a PC running the TDE goes into Hibernation mode while a test was running. Although Windows hibernation is automatically disabled while a test is running, if the PC is sent to hibernation or powered off by the user, the TDE will not receive updates from the Appliance. If the TDE does not receive the statistics update that contains the "end of project" indication then the TDE will not know that the Project has terminated. If the Duration timer that the TDE keeps completes and the TDE has not received the "end of project" indication then it will wait for at least 5 minutes and then it will stop waiting and issue a "Test Expired" error message.

TDE, Project and Output Logs

The TDE captures log information in several files that may be useful in debugging TDE and/or Project behaviors. There are two log files generated during a Project compilation or execution

Output Log file: Output.log

Located in the Results folder. Only captured during Project execution. Records all of the time-stamped Project-related information concerning start and stop time, error messages, and results for a Project execution. Example Output.log

	Line	Type	Date / Time	Text	
▶	0	Status	1/28/2014 1:28:41 PM	Load DynamiX Framework [Version 5.35.27060-Internal-Private_MPIO_ALUA]	
	1	Status	1/28/2014 1:28:41 PM	(C) Copyright 2008-2014 Load DynamiX, Inc.	
ⓘ	2	Info	1/28/2014 1:28:41 PM	Preparing to execute test: APPL-2518-validation	
ⓘ	3	Info	1/28/2014 1:28:41 PM	Device [11]: Port type SwiftTest 16Gb Fibre Channel	
ⓘ	4	Info	1/28/2014 1:28:41 PM	Device [11]: Linked speed 16 Gbps Full Duplex.	
ⓘ	5	Info	1/28/2014 1:28:41 PM	Device [10]: Port type SwiftTest 16Gb Fibre Channel	
ⓘ	6	Info	1/28/2014 1:28:41 PM	Device [10]: Linked speed 16 Gbps Full Duplex.	
ⓘ	7	Info	1/28/2014 1:28:41 PM	Device [9]: Port type SwiftTest 16Gb Fibre Channel	
ⓘ	8	Info	1/28/2014 1:28:41 PM	Device [9]: Linked speed 16 Gbps Full Duplex.	
ⓘ	9	Info	1/28/2014 1:28:41 PM	Device [8]: Port type SwiftTest 16Gb Fibre Channel	
ⓘ	10	Info	1/28/2014 1:28:41 PM	Device [8]: Linked speed 16 Gbps Full Duplex.	
ⓘ	11	Info	1/28/2014 1:28:41 PM	Device [3]: Port type SwiftTest 16Gb Fibre Channel	
ⓘ	12	Info	1/28/2014 1:28:41 PM	Device [3]: Linked speed -1 Gbps Full Duplex.	
ⓘ	13	Info	1/28/2014 1:28:42 PM	Device [2]: Port type SwiftTest 16Gb Fibre Channel	
ⓘ	14	Info	1/28/2014 1:28:42 PM	Device [2]: Linked speed -1 Gbps Full Duplex.	
ⓘ	15	Info	1/28/2014 1:28:42 PM	Device [1]: Port type SwiftTest 16Gb Fibre Channel	
ⓘ	16	Info	1/28/2014 1:28:42 PM	Device [1]: Linked speed -1 Gbps Full Duplex.	
ⓘ	17	Info	1/28/2014 1:28:42 PM	Device [0]: Port type SwiftTest 16Gb Fibre Channel	
ⓘ	18	Info	1/28/2014 1:28:42 PM	Device [0]: Linked speed -1 Gbps Full Duplex.	
⚡	19	Debug	1/28/2014 1:28:42 PM	Device [11] subnet [0]: Initialized 255 IPv4 Addresses in range 192.169.1.2 - 192.169.1.0	
ⓘ	20	Info	1/28/2014 1:28:42 PM	Expected execution duration is up to 00:01:00	
ⓘ	21	Info	1/28/2014 1:28:42 PM	Client-side high-performance stack is entering active state...	
ⓘ	22	Info	1/28/2014 1:29:34 PM	All scenarios ramped-down. Exiting...	
ⓘ	23	Info	1/28/2014 1:29:34 PM	Client-side high-performance stack is exiting active state...	
ⓘ	24	Info	1/28/2014 1:29:34 PM	Client-side high-performance stack execution time 00:00:52.	

Project Log file: Project.log

Located in the Project Folder. Records Compilation log entries as well as the same information as the Output Log file but can be made to be cumulative so as to record this information over a number of executions. Example Project.log.

```
[02/25/2013 14:01:01,SwiftTest TDE,2296,FileInfo] Compiling project: SMB2 Full Duplex Payload.swift_test
[02/25/2013 14:01:01,SwiftTest TDE,2296,Info] Verifying project...
[02/25/2013 14:01:01,SwiftTest TDE,2296,Info] Creating documents list...
[02/25/2013 14:01:01,SwiftTest TDE,2296,Info] Compiling...
[02/25/2013 14:01:01,SwiftTest TDE,2296,FileInfo] 24 Users-Passwords-Files.user
[02/25/2013 14:01:01,SwiftTest TDE,2296,FileInfo] Client Port 0.client_port
[02/25/2013 14:01:02,SwiftTest TDE,2296,FileInfo] Sample Test Exec Rules.test_rules
[02/25/2013 14:01:02,SwiftTest TDE,2296,FileInfo] 24 Data Files.dfs
[02/25/2013 14:01:02,SwiftTest TDE,2296,FileInfo] ClientNET 240.network
[02/25/2013 14:01:02,SwiftTest TDE,2296,FileInfo] Client SMB2 Write.client_scenario
[02/25/2013 14:01:02,SwiftTest TDE,2296,FileInfo] Client Load Steady.load
[02/25/2013 14:01:02,SwiftTest TDE,2296,FileInfo] Client SMB2 Read.client_scenario
[02/25/2013 14:01:02,SwiftTest TDE,2296,FileInfo] Server Port 4.server_port
[02/25/2013 14:01:02,SwiftTest TDE,2296,FileInfo] ServerNET 244.network
[02/25/2013 14:01:02,SwiftTest TDE,2296,FileInfo] Server SMB2 Basic 244.server_scenario
[02/25/2013 14:01:02,SwiftTest TDE,2296,FileInfo] Server Load Basic.load
[02/25/2013 14:01:02,SwiftTest TDE,2296,Info] Verifying compiled project...
[02/25/2013 14:01:02,SwiftTest TDE,2296,Info] Linking...
[02/25/2013 14:01:02,SwiftTest TDE,2296,Info] Creating object files...
[02/25/2013 14:01:03,SwiftTest TDE,2296,Info] Compile complete - 0 errors, 0 warnings
[02/25/2013 14:03:26,SwiftTest TDE,2296,FileInfo] Compiling project: SMB2 Full Duplex Payload.swift_test
[02/25/2013 14:03:26,SwiftTest TDE,2296,Info] Verifying project...
[02/25/2013 14:03:26,SwiftTest TDE,2296,Info] Creating documents list...
[02/25/2013 14:03:26,SwiftTest TDE,2296,Info] Compiling...
[02/25/2013 14:03:26,SwiftTest TDE,2296,FileInfo] 24 Users-Passwords-Files.user
[02/25/2013 14:03:26,SwiftTest TDE,2296,FileInfo] Client Port 0.client_port
[02/25/2013 14:03:26,SwiftTest TDE,2296,FileInfo] Sample Test Exec Rules.test_rules
[02/25/2013 14:03:26,SwiftTest TDE,2296,FileInfo] 24 Data Files.dfs
[02/25/2013 14:03:26,SwiftTest TDE,2296,FileInfo] ClientNET 240.network
[02/25/2013 14:03:26,SwiftTest TDE,2296,FileInfo] Client SMB2 Write.client_scenario
[02/25/2013 14:03:26,SwiftTest TDE,2296,FileInfo] Client Load Steady.load
[02/25/2013 14:03:26,SwiftTest TDE,2296,FileInfo] Client SMB2 Read.client_scenario
[02/25/2013 14:03:26,SwiftTest TDE,2296,FileInfo] Server Port 4.server_port
[02/25/2013 14:03:26,SwiftTest TDE,2296,FileInfo] ServerNET 244.network
[02/25/2013 14:03:26,SwiftTest TDE,2296,FileInfo] Server SMB2 Basic 244.server_scenario
[02/25/2013 14:03:26,SwiftTest TDE,2296,FileInfo] Server Load Basic.load
[02/25/2013 14:03:26,SwiftTest TDE,2296,Info] Verifying compiled project...
[02/25/2013 14:03:26,SwiftTest TDE,2296,Info] Linking...
[02/25/2013 14:03:26,SwiftTest TDE,2296,Info] Creating object files...
[02/25/2013 14:03:26,SwiftTest TDE,2296,Info] Sorting ports...
[02/25/2013 14:03:26,SwiftTest TDE,2296,Info] Checking appliance version...
[02/25/2013 14:03:26,SwiftTest TDE,2296,Info] Appliance (172.17.1.49) verification completed.
[02/25/2013 14:03:26,SwiftTest TDE,2296,Error] 172.17.1.49 - software versions are equal (1.31.22451-Internal)
[02/25/2013 14:03:26,SwiftTest TDE,2296,Info] Checking port(s) status...
[02/25/2013 14:03:26,SwiftTest TDE,2296,Info] 172.17.1.49:5 - Idle
[02/25/2013 14:03:26,SwiftTest TDE,2296,Info] 172.17.1.49:1 - Idle
[02/25/2013 14:03:26,SwiftTest TDE,2296,Info] Uploading ports...
[02/25/2013 14:03:26,SwiftTest TDE,2296,Info] Uploading documents to port: 172.17.1.49:5...
```

and a log file that is updated by the TDE to capture Windows version information, exception back-traces, key folder locations, etc.

TDE Log file: SwiftGlobal.log

Located in the "Log" directory described in the [Product Installation chapter](#).

Successful, Failed, Aborted Actions and Scenarios

Load DynamiX Protocol (CIFS-SMB, SMB2, HTTP, etc) Actions

A Protocol Action is considered Successful if it meets the following criteria:

- (1) The connection for the requested Action is open
- (2) All of the (packet) fragments of the requested Action were successfully sent over the connection.
- (3) All of the (packet) fragments of the requested Action were successfully acknowledged as having been received.
- (4) At least one (packet) fragment of the response of the requested Action has been received.
- (5) The received (packet) fragment of the response is identified as belonging to the

requested Action.

- (6) The first (packet) fragment of the response does not contain an error code other than "Success".
- (7) All (packet) fragments of the requested Action are received.
- (8) Acknowledgments of all (packet) fragments of the requested Action are sent successfully.

If any one of the above 8 items is not true then:

- If 1. or 6. is not True, then the requested Action Failed.
- If 2. or 5. or 7. is not True and if the connection is terminated, then the requested Action is marked Aborted, otherwise the requested Action is marked Failed.
- If 3. or 4. or 8. is not True, then the requested Action is marked Aborted.

Load DynamiX Scenario Control Actions

Never marked as Succeeded, Failed or Aborted.

Load DynamiX Open Connection (CIFS-SMB, SMB2, HTTP, etc) Actions

Are marked as Failed if the Address Resolution process fails, or if the open request is refused, or if the open request times out (based on [Network Profile](#) settings), otherwise marked as Successful.

Load DynamiX Close Connection Actions

Always marked as Successful.

Successful Failed and Aborting Scenarios

A Successful Scenario is one in which all of the Actions in the Scenario complete with a Success status, otherwise the Scenario is marked as Failed or Aborted as follows:

- Any Scenario that does not complete by the end of the Project Duration time is marked as Aborted.
- Any Scenario terminated before executing its last Action is marked as Failed.
- Any Scenario which has an open connection terminated by the Peer is marked Aborted.
- Any Scenario that has an Action fail or abort due to 1.-8. not being True is marked Failed.
- Any Scenario in which an Action has a [Completion Status](#) of Terminate is marked Failed.

Scenarios that are marked as Aborted can be caused by a variety of reasons:

- A connection with the DUT was not achieved (IP address mismatch, routing or switching issue).
- Actions executed against the DUT failed.
- Actions executed against the DUT timed out.
- Ramp Up time is too short (not enough time to get to the desired load).
- Ramp Down time is too short (not enough time to cleanup running Scenarios before Project Duration expires).
- Project Duration is too short.

What can be done to address Aborted Scenarios?

- Check the correctness of the IP addresses in Open Actions, if the test environment uses Spanning Tree protocol allow 30 seconds in the Ramp Up for Spanning Tree to resolve addresses, make sure that IP addresses for DUT are reachable from the Load DynamiX Appliance's test ports (using trace route and other network utilities).

- Look in the PCAP file to see why the Actions are failing. Permissions, missing files or share folders, write or read maximum sizes are considerations.
- Time outs. Is the TCP inactivity timeout value in the Network Profile sufficiently large (or set to 0 to disable timeouts completely)? Is the TCP retransmit count high enough?
- Ramp Up time can be increased until the desired load is reached.
- Ramp Down time can be increased to give Projects more time to clean up at the end.
- Check Project Duration - is it long enough to complete all Scenarios.

Automation

While the Load DynamiX TDE is a convenient, graphical means to define, execute and analyze tests, in production test environments, it is necessary to be able to automate the execution of tests. The answer to this requirement is delivered with the Load DynamiX product and is named LdxCmd.exe. The LdxCmd.exe executable can be found in the same folder that the Load DynamiX GUI (LdxTDE.exe) is installed in. LdxCmd.exe provides, from the command line, what the Compile Project and Start Test and Generate Automation File GUI features provide in the GUI. To learn more about Load DynamiX automation, see [Appendix: Test Automation and LDX-E Integration](#).

How to use LdxCmd

Create, compile and execute your Swifttest Project using standard Load DynamiX GUI ("LdxTDE") to verify that it produces the desired results.

Command line execution of a Load DynamiX Project is controlled by two files that are stored in the Project folder. When LdxTDE is installed, it creates a folder (for example: C:\Users\<user>\Documents\SwifTest\My Projects\) for storing Projects that are to be run on the user's PC.

In each Project's folder, there exists a file with the extension .swift_test and a folder named "obj". In the "obj" folder are the Project's "object" files which are used when the Project is executed from the TDE. If Automation files have been "Generated" then there will also be a folder containing the Automation files (by default the folder is named AutomationConfig). The AutomationConfig folder contains the file named AutomationConfig.XML as well as other Automation related files.

This file AutomationConfig.XML is used to run the test from the command line using LdxCmd.exe. The LdxCmd.exe executable is located in the Load DynamiX **{InstallationFolder}** (see [Product Installation chapter](#)). This is not in the Project Directory but the program installation directory which is by default: C:\Program Files (x86)\Load DynamiX\Load DynamiX TDE.

LdxCmd produces very limited output (to files or to the console/screen) unless options like /Statmode and /Applog are used. If no output is seen and no command line prompt appears, LdxCmd is likely executing.

See [Appendix: Test Automation and LDX-E Integration](#) for details of LdxCmd syntax and examples.

Viewing Results from Automation

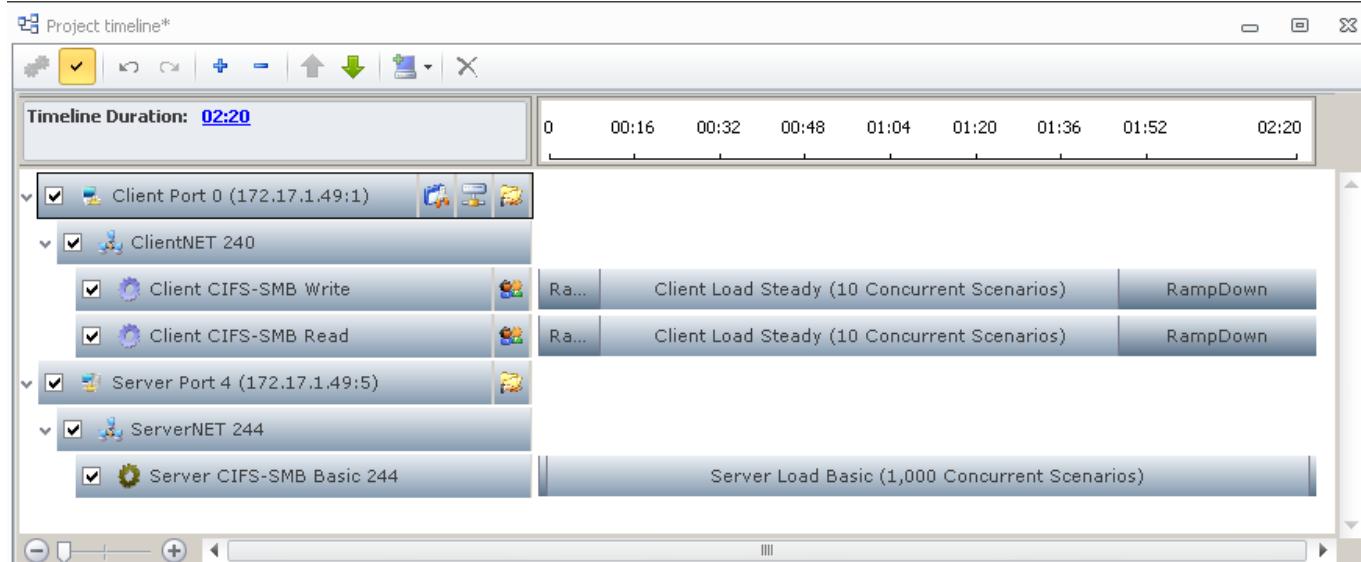
Project statistics are captured and stored in a Results folder just as if the Project was executed by the TDE. The Results are captured either in the AutomationConfig folder or wherever /Out: indicates and then copied into the TDE's Results folder when the Automated project completes execution. The Automation Results folder will be visible in the Project's Results Explorer the next time the Project is opened by the TDE. If the Project is open in the TDE when the Automation is executed, the new Results folder will not appear in the Results Explorer until the Project has been closed and re-opened.

Port Delay impact

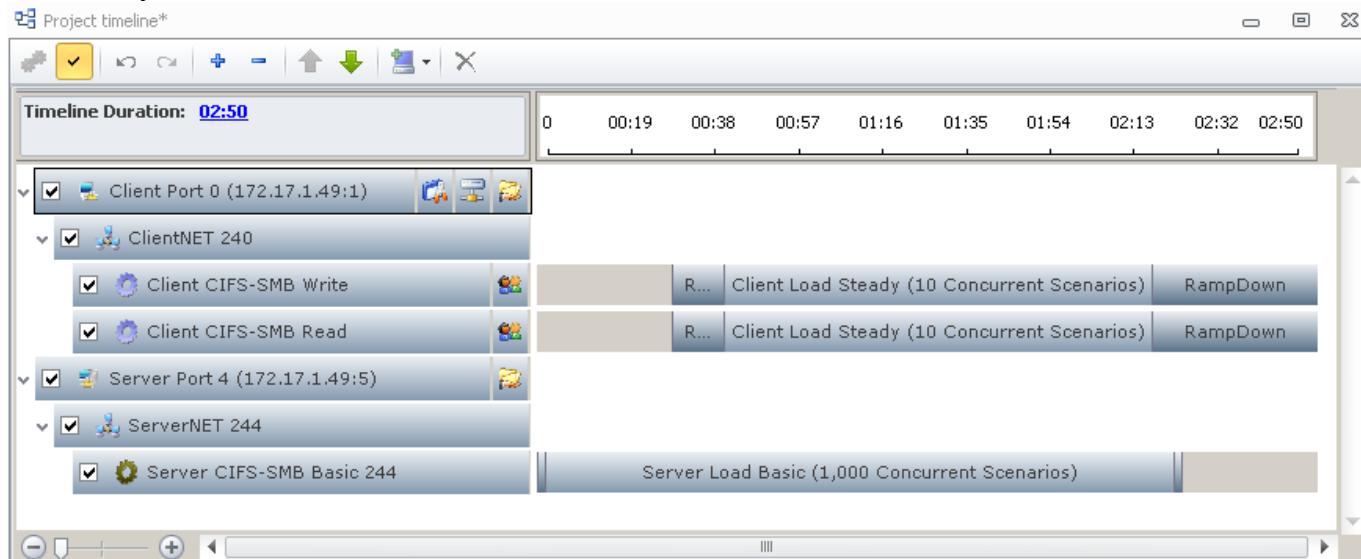
When Port Delay is set to a positive non-zero value, the execution of the Scenarios assigned to the Logical Port with the Port Delay value > 0 , is delayed for number of seconds specified in Port Delay . The Tester will see the impact of Port Delay in the graphs of the execution of that Project as demonstrated below. Shown are both the Timeline view with Port Delay == 0 and Port Delay == 30 as well as (an example) of the statistics output for Port Delay ==0 and Port Delay == 30. The Logical Port in the Project below with Port Delay == 30 shows in the Timeline view and the Load Status graph that execution of the Scenarios begin 30 seconds after the Tester clicks on the Start button. Also note that in the Port Delay == 30 case, the overall Duration of the Project has been increased to incorporate the 30 seconds of delay.

TimeLine

Port Delay == 0

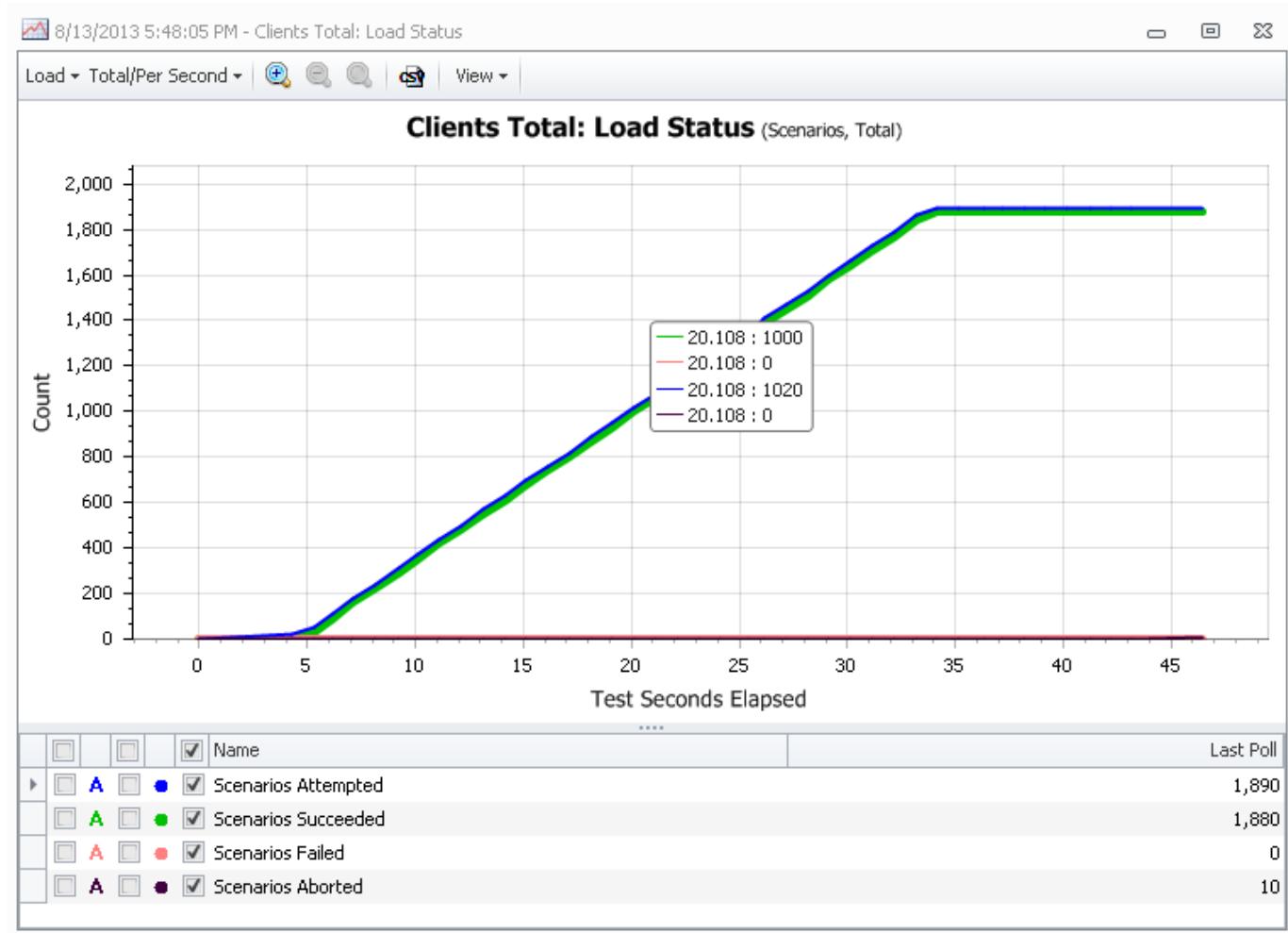


Port Delay == 30

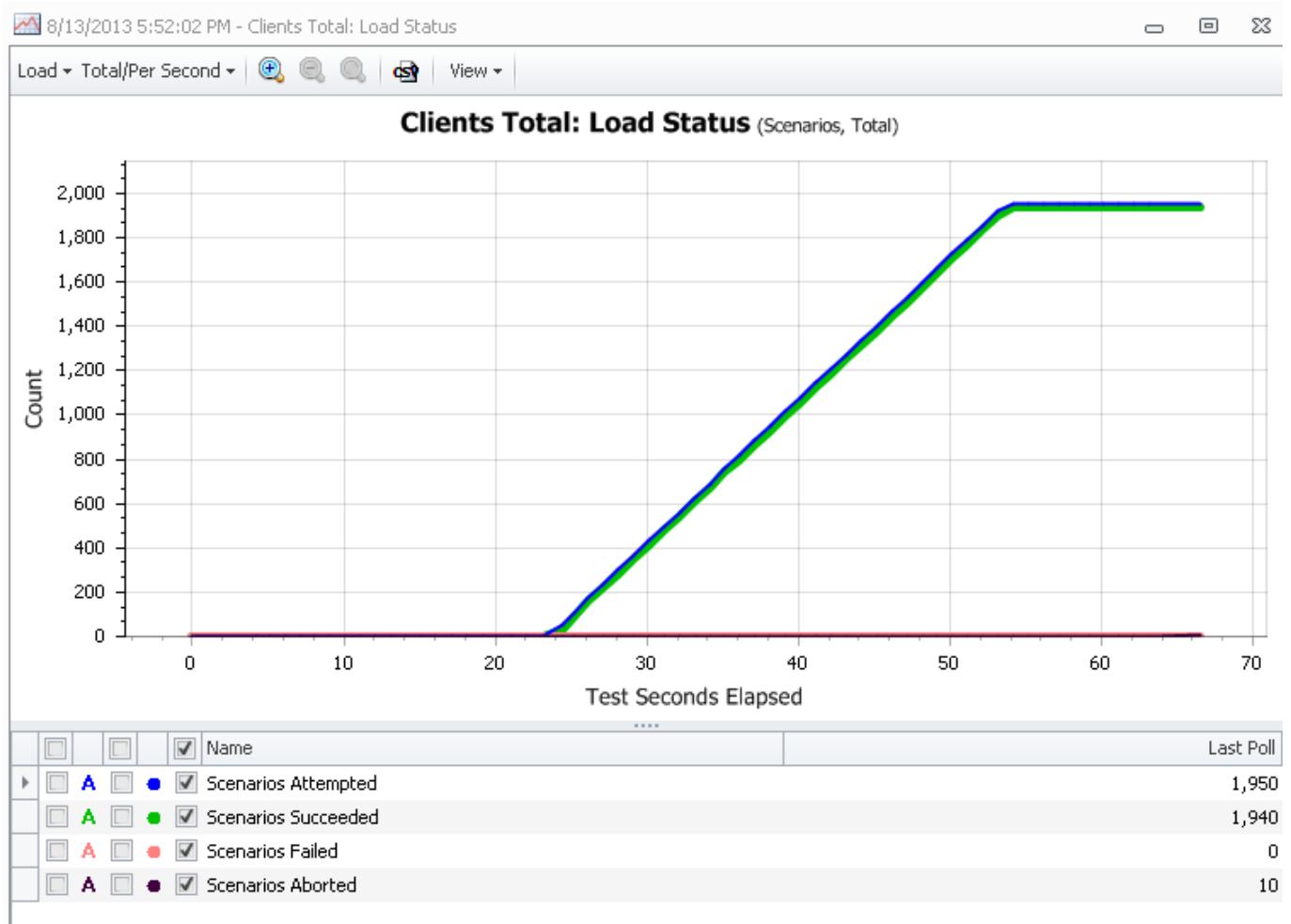


Statistics (Load Status example)

Port Delay == 0



Port Delay == 20

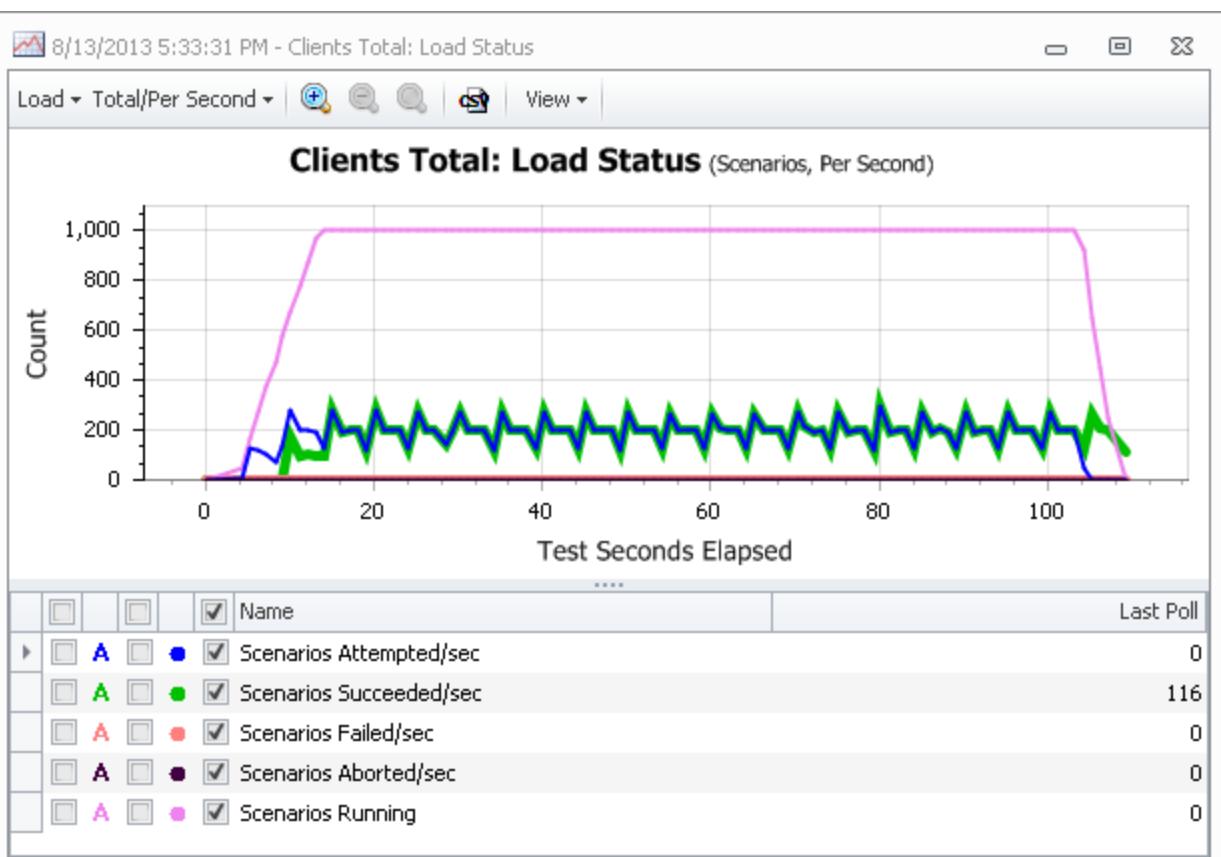


Output reporting and analysis

The Output window displays the test run processing as it occurs. Toolbar buttons and windows also exist to provide network and protocol statistics:

- Project Summary – Displays information based on the Project Timeline and Load
- Timeline – Shows a tree diagram for the executed Scenario(s) that shows the processing time spent based on the information configured in the Load Profile
- Load – Shows a graphical representation of the load processing based on the metric specified in the Load Profile (e.g., Actions, Actions/second, etc.).
- Client and Server Scenario statistics – shows the packet data rate that occurred during the test

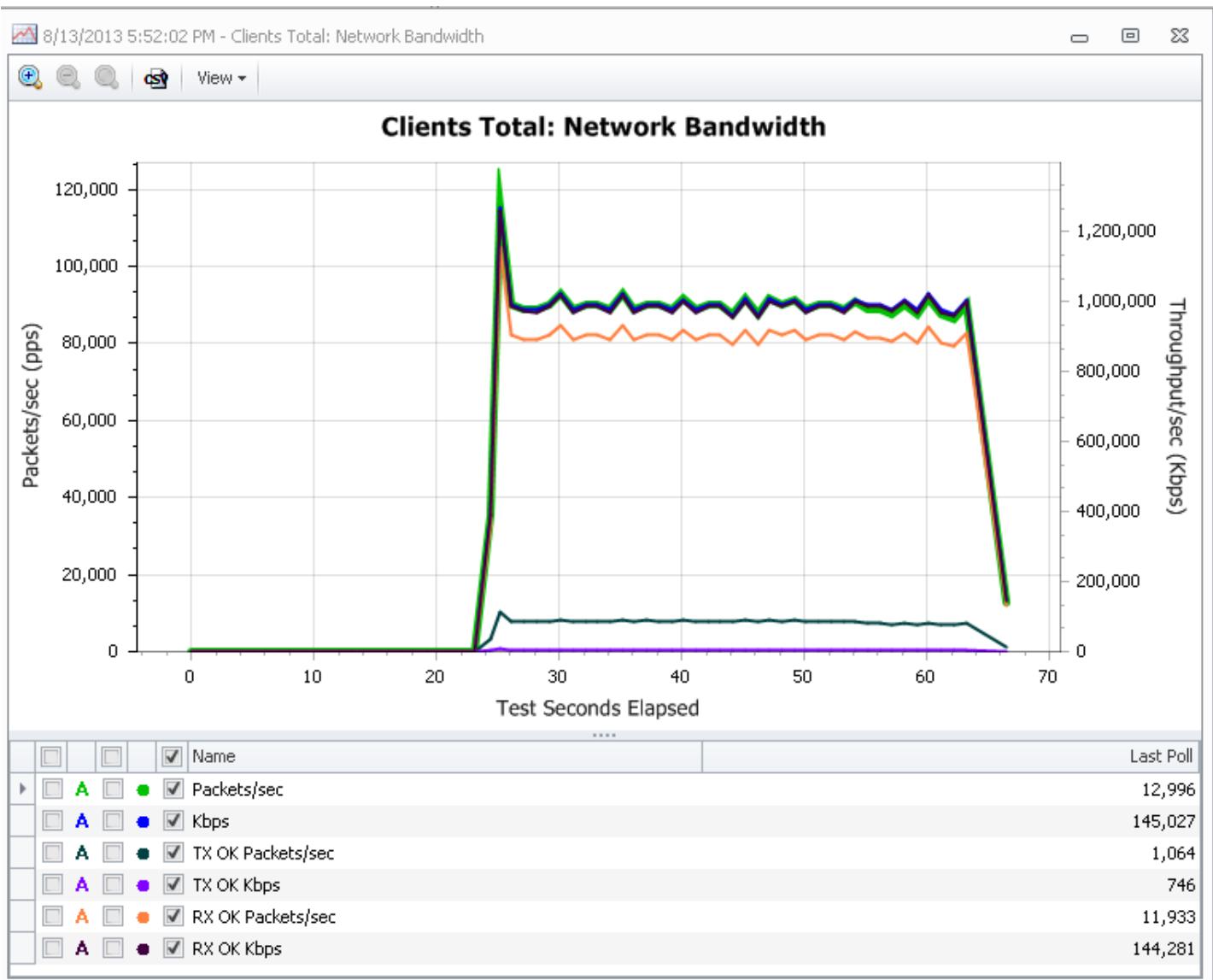
The Load Status (Scenarios Per Second) graph below shows an overall sawtooth behavior for the Scenarios (green/blue lines) and the 1000 per second execution rate as captured by the pink line.



Results Explorer

NOTE: Substituting scaled or custom fonts for standard Windows fonts may cause issues when graphs are displayed.

The Results Explorer window provides access to output graphs associated with each successful test run. The application generates a folder with a name based on the time-stamp of when the test was run.



To display the graphical representation of a file

- Open the appropriate folder (e.g. Clients Total)
- Identify the statistics of interest (e.g. SMB2 Commands)
- Double-click the file name.

Note: Changes made to default graph settings are retained with the graph and will be used when displaying that information in the future. For example, in the graph above if the A box is checked to show the packets/second values and the graph is closed, the next time this graph is opened, the A box will automatically checked.

To export the results to a CSV file

- Select a file or directory of files.
- Click the Export to CSV button.
- Specify the CSV file location. You may want to create a new folder if you are exporting all of the results from a test.
- Click OK.

Load DynamiX Results Folder Statistics Reports

Results Folder Statistics Report Name	Contents
Load Status	Scenario, Action or Connections attempts, successes and failures
Network Bandwidth	Bandwidth or TCP throughput in packets/sec or kilobits/sec, receive and transmit
SMB Actions	SMB Action counts or Actions/sec (average for all or individual Actions)
SMB Commands	SMB command counts (attempts, successes, failures or aborts)
SMB Details	SMB command transmission/receipt OK/Fail/Drop in packets/sec or kilobits/sec
SMB Response Time	SMB command response time (average for all or individual commands)
SMB Throughput	SMB packet or byte throughput on per command or All basis
SMB Data Verification (client results only)	SMB data verification operations attempts, successes, failures
SMB Signing	SMB commands count or per second when SMB Signing enabled (attempts, successes, failures)
SMB Latency Time (server results only)	SMB server response latency average in micro-seconds, milli-seconds, seconds
TCP Connection Time	IPv4 TCP Connection time in micro-seconds
TCP Connections	IPv4 TCP connections attempted/opened/closed, count or per second
TCP Details	IPv4 TCP packet/byte transmitted or received, count or per second
TCP Throughput	IPv4 TCP throughput in packets/sec or kilobits/sec, receive and transmit
TCPv6 Connection Time	IPv6 TCP Connection time in micro-seconds
TCPv6 Connections	IPv6 TCP connections attempted/opened/closed, count or per second
TCPv6 Details	IPv6 TCP packet/byte transmitted or received, count or per second
TCPv6 Throughput	IPv6 TCP throughput in packets/sec or kilobits/sec, receive and transmit
SMB2 Actions	SMB2 Action counts or Actions/sec (average for all or individual Actions)
SMB2 Commands	SMB2 command counts (attempts, successes, failures or aborts)
SMB2 Details	SMB2 command transmission/receipt OK/Fail/Drop in packets/sec or kilobits/sec
SMB2 Response Time	SMB2 command response time (average for all or individual commands)
SMB2 Throughput	SMB2 packet or byte throughput on per command or All basis
SMB2 Data Verification (client results only)	SMB2 data verification operations attempts, successes, failures
SMB2 Latency Time (server results only)	SMB2 server response latency average in micro-seconds, milli-seconds, seconds
SMB2 Olock and Lease Breaks	SMB2 Olock and Lease Break processing (received, acked, ignored, aborted, etc)
MSRPC/SMB2 Actions	MSRPC/SMB2 Action counts or Actions/sec (average for all or individual Actions)
MSRPC/SMB2 Commands	MSRPC/SMB2 command counts (attempts, successes, failures or aborts)
MSRPC/SMB2 Details	MSRPC/SMB2 command transmission/receipt OK/Fail/Drop in packets/sec or kilobits/sec
MSRPC/SMB2 Response Time	MSRPC/SMB2 command response time (average for all or individual commands)
MSRPC/SMB2 Throughput	MSRPC/SMB2 packet or byte throughput on per command or All basis
RPC Actions	RPC/NFS Action counts or Actions/sec (average for all or individual Actions)
RPC Commands	RPC/NFS command counts (attempts, successes, failures or aborts)
RPC Opcodes	RPC/NFS opcode counts (attempts, successes, failures or aborts)
RPC Response Time	RPC/NFS command response time (average for all or individual commands)
RPC Throughput	RPC/NFS command throughput packets/sec or kilobits/sec
RPC Data Verification	RPC/NFS data verification operations attempts, successes, failures
Kerberos Throughput	Kerberos command throughput packets/sec or kilobits/sec
Kerberos Actions	Kerberos Actions or Actions/sec (average for all or individual Actions)
Kerberos Commands	Kerberos command counts (attempts, successes, failures or aborts)
Kerberos Details	Kerberos command transmission/receipt OK/Fail/Drop in packets/sec or kilobits/sec
Kerberos Response Time	Kerberos command response time (average for all or individual commands)
HTTP Actions	HTTP Action counts or Actions/sec (average for all or individual Actions)
HTTP Commands	HTTP command counts (attempts, successes, failures or aborts)
HTTP Details	HTTP command transmission/receipt OK/Fail/Drop in packets/sec or kilobits/sec
HTTP Response Time	HTTP command response time (average for all or individual commands)
HTTP Authentications	HTTP Authentication counts (attempts, successes, failures or aborts)
HTTP Authentications Time	HTTP Authentication attempt time required
HTTP Throughput	HTTP packet or byte throughput on per command or All basis
HTTPS Actions	HTTPS Action counts or Actions/sec (average for all or individual Actions)
HTTPS Commands	HTTPS command counts (attempts, successes, failures or aborts)
HTTPS Details	HTTPS command transmission/receipt OK/Fail/Drop in packets/sec or kilobits/sec
HTTPS Response Time	HTTPS command response time (average for all or individual commands)
HTTPS Authentications	HTTPS Authentication counts (attempts, successes, failures or aborts)
HTTPS Authentication Time	HTTPS Authentication attempt time required
HTTPS Throughput	HTTPS packet or byte throughput on per command or All basis
SSL/TLS Connection Time	SSL/TLS Connection time in micro-seconds
SSL/TLS Connections	SSL/TLS connections attempted/opened/closed, count or per second
SSL/TLS Details	SSL/TLS packet/byte transmitted or received, count or per second
CDMI Actions	CDMI Action counts or Actions/sec (average for all or individual Actions)

CDMI Commands	CDMI command counts (attempts, successes, failures or aborts)
CDMI Data Verification	CDMI data verification operations attempts, successes, failures
CDMI Details	CDMI command transmission/receipt OK/Fail/Drop in packets/sec or kilobits/sec
CDMI Response Time	CDMI command response time (average for all or individual commands)
CDMI Throughput	CDMI packet or byte throughput on per command or All basis
OpenStack Swift Actions	OpenStack Swift Action counts or Actions/sec (average for all or individual Action)
OpenStack Swift Commands	OpenStack Swift command counts (attempts, successes, failures or aborts)
OpenStack Swift Data Verification	OpenStack Swift data verification operations attempts, successes, failures
OpenStack Swift Details	OpenStack Swift command transmission/receipt OK/Fail/Drop in packets/sec or kilobits/sec
OpenStack Swift Response Time	OpenStack Swift command response time (average for all or individual command)
OpenStack Swift Throughput	OpenStack Swift packet or byte throughput on per command or All basis
iSCSI Actions	iSCSI Action counts or Actions/sec (average for all or individual Actions)
iSCSI Commands	iSCSI command counts (attempts, successes, failures or aborts)
iSCSI Details	iSCSI command transmission/receipt OK/Fail/Drop in packets/sec or kilobits/sec
iSCSI Response Time	iSCSI command response time (average for all or individual commands)
iSCSI Throughput	iSCSI packet or byte throughput on per command or All basis
iSCSI Data Verification (client results only)	iSCSI data verification operations attempts, successes, failures
iSCSI Latency Time (server results only)	iSCSI server response latency average in micro-seconds, milli-seconds, seconds
FC SCSI Actions	Fibre Channel SCSI Action counts or Actions/sec (average for all or individual Action)
FC SCSI Commands	Fibre Channel SCSI command counts (attempts, successes, failures or aborts)
FC SCSI Details	Fibre Channel SCSI command transmission/receipt OK/Fail/Drop in packets/sec
FC SCSI Response Time	Fibre Channel SCSI command response time (average for all or individual command)
FC SCSI Throughput	Fibre Channel SCSI packet or byte throughput on per command or All basis
FC SCSI Data Verification	Fibre Channel SCSI data verification operations attempts, successes, failures
FC Network Bandwidth	Fibre Channel Bandwidth or throughput in frames/sec or kilobits/sec, receive and transmit
FC Sessions	Fibre Channel Sessions attempted/opened/closed, count or per second
FC Session Time	Fibre Channel Session time in micro-seconds
UDP Connection Time	UDP Connection time in micro-seconds
UDP Details	UDP packet/byte transmitted or received, count or per second
UDP Throughput	UDP throughput in packets/sec or kilobits/sec, receive and transmit
UDP Transport Initializations	UDP Transports Opened/Failed, DNS Resolutions Attempted/Succeeded/Failed, Resolutions Attempted/Succeeded/Failed
UDP Transport Summary	UDP Transports Opened, Closed, Failed, and Reset; Timeouts: Data and Inactivity

See the [Advanced Concepts: Test Execution Rules](#) for a complete list of the statistics generated when DynamiX Projects are executed.

Simple Reports and the Report Wizard

For a given Project results folder, it is possible to produce a report that combines User-selected graphs into a single document. The Reports Wizard is a means to produce a final report on a test against a device or set of devices.

To get started, highlight a specific Results folder and click the Report Wizard Button  and then click

[Next >](#)

Welcome to Report Wizard

Welcome to the Load DynamiX report wizard! This wizard will guide you through the process of generating detailed reports from your existing test results.

In this wizard you can select which statistics to include, change their order, and include your own notes and company logo.

You may also instantly generate a report by skipping the optional steps and clicking the finish button to create a Load DynamiX report with the default settings.

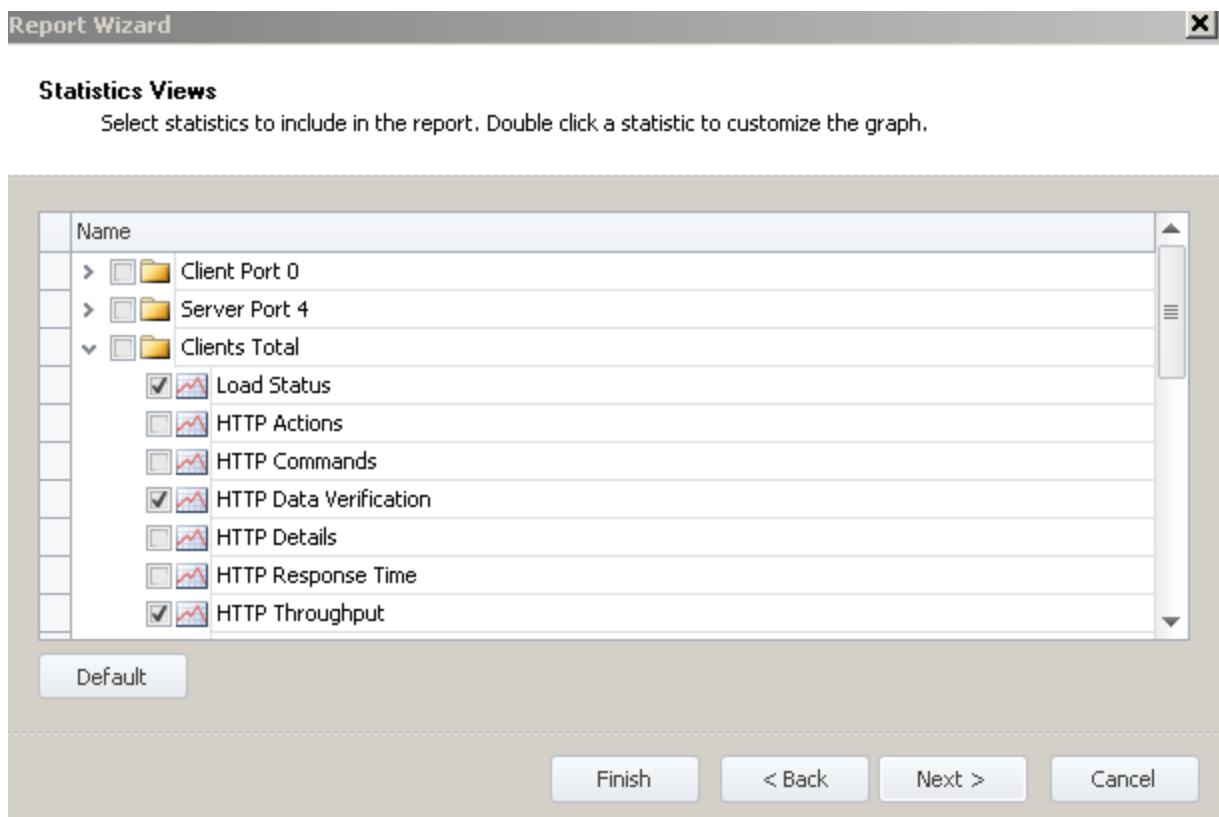
To continue, click Next

< Back

Next >

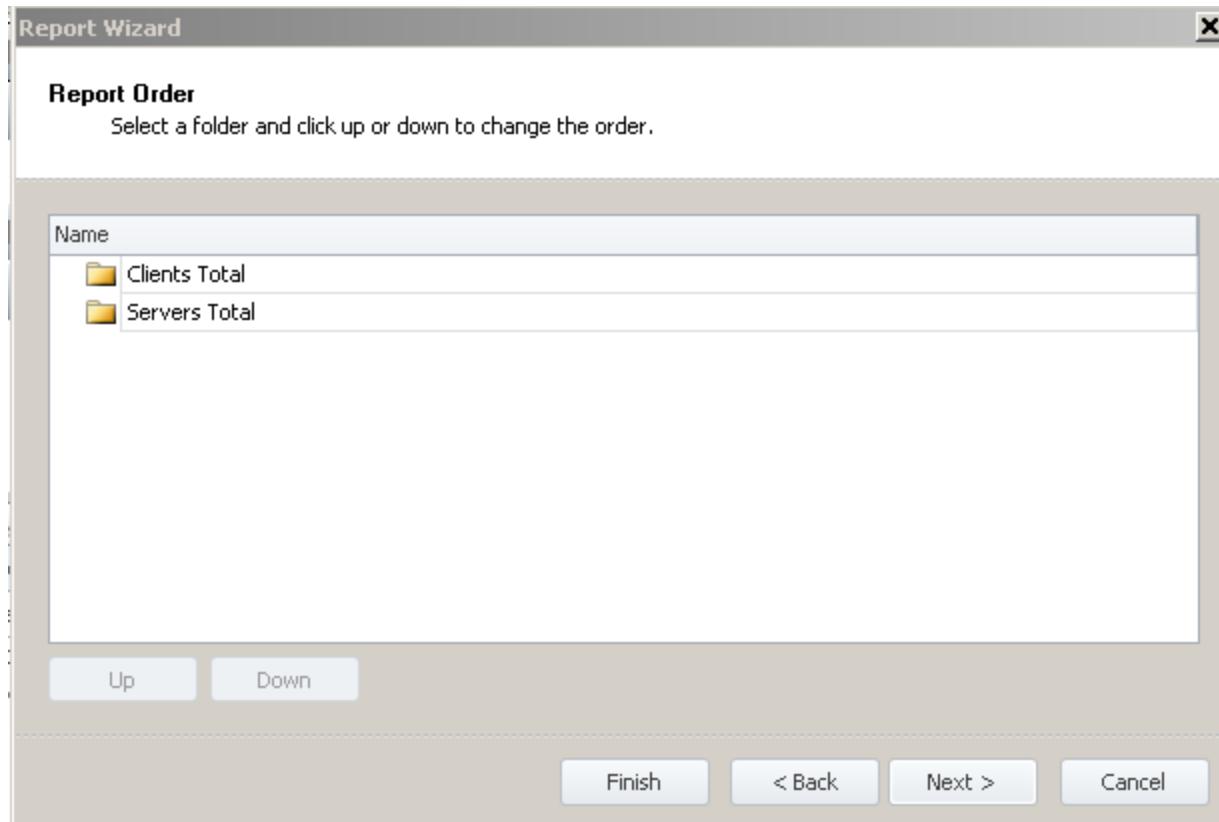
Cancel

In the Statistics Views window select the graphs that are to appear in the Simple Report. Below, Load Status, iSCSI Actions, ISCSI Throughput and Network Bandwidth have been selected to appear in this report. When done selecting the graphs, click or .

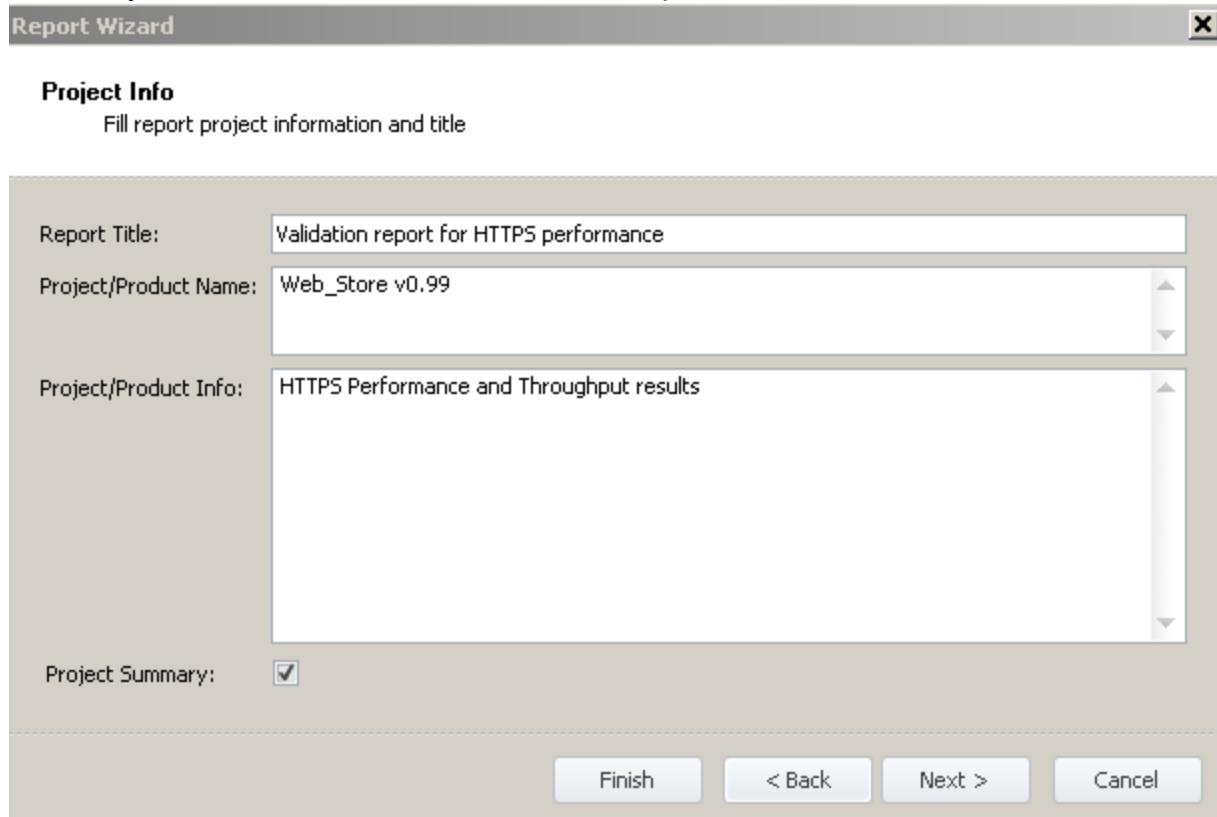


Selecting **Next >** provides the Tester the ability to select the order of folders if more than one is selected and add some information to a cover page for the report. Clicking **Finish** takes the Tester to the final document containing an empty cover page and the selected graphs.

Select **Next >** and see that two folders have been selected for this report - Clients Total and Servers Total.

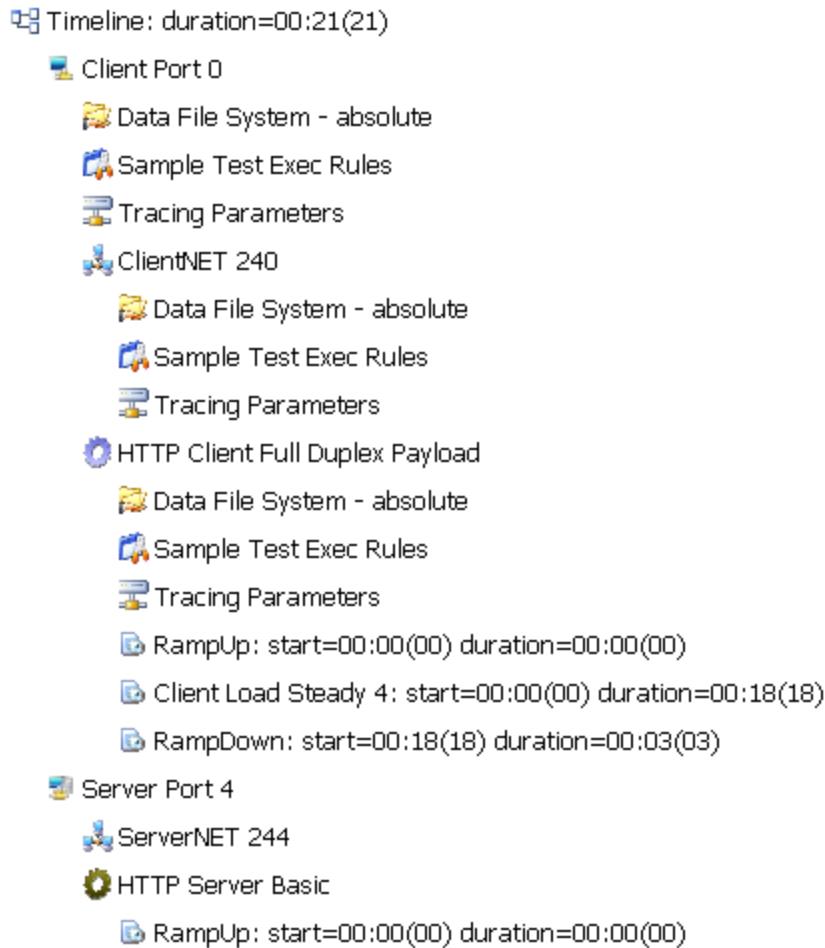


Select [Next >](#) to provide information on the Project and Product being tested and decide if the Project Summary information (the same information as in the Project Summary page) should be added to the report. The Project Summary information is added by default. Un-check the Project Summary check-box if this information is not required.

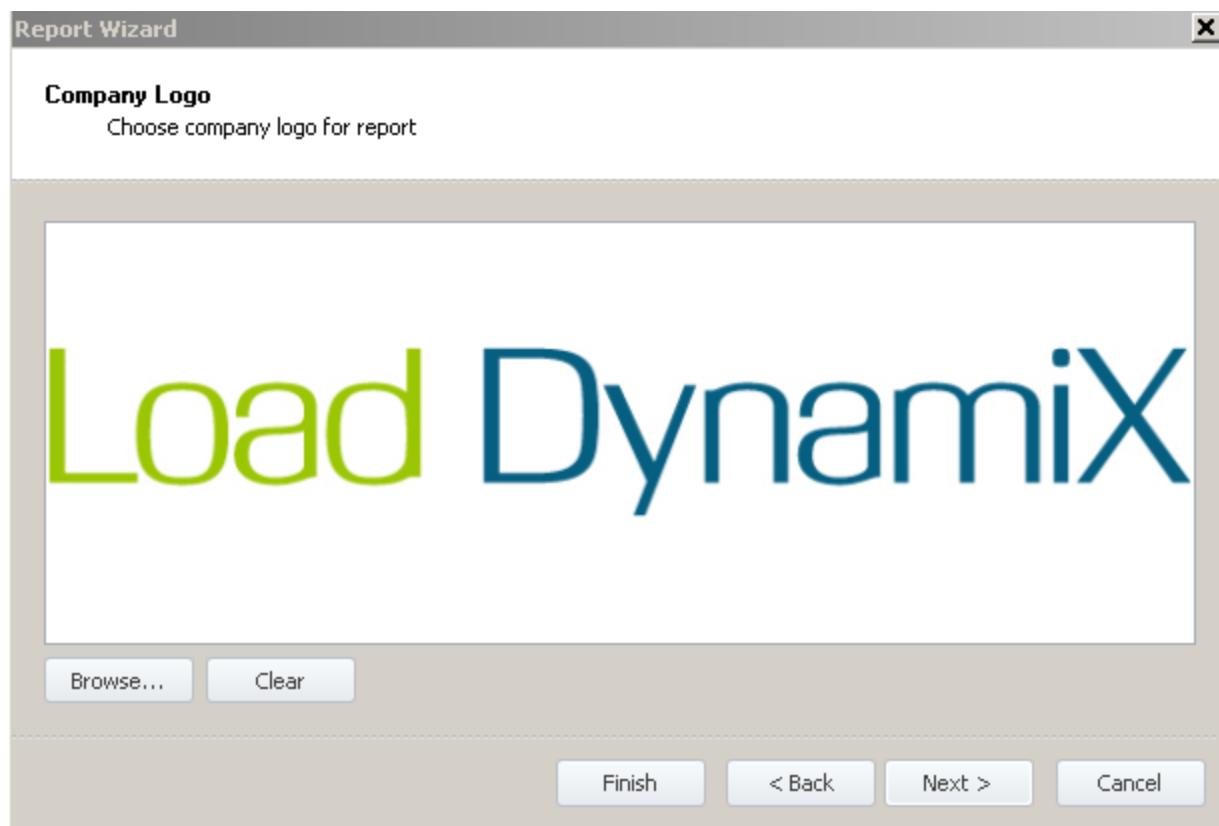


An example of the Project Configuration information is

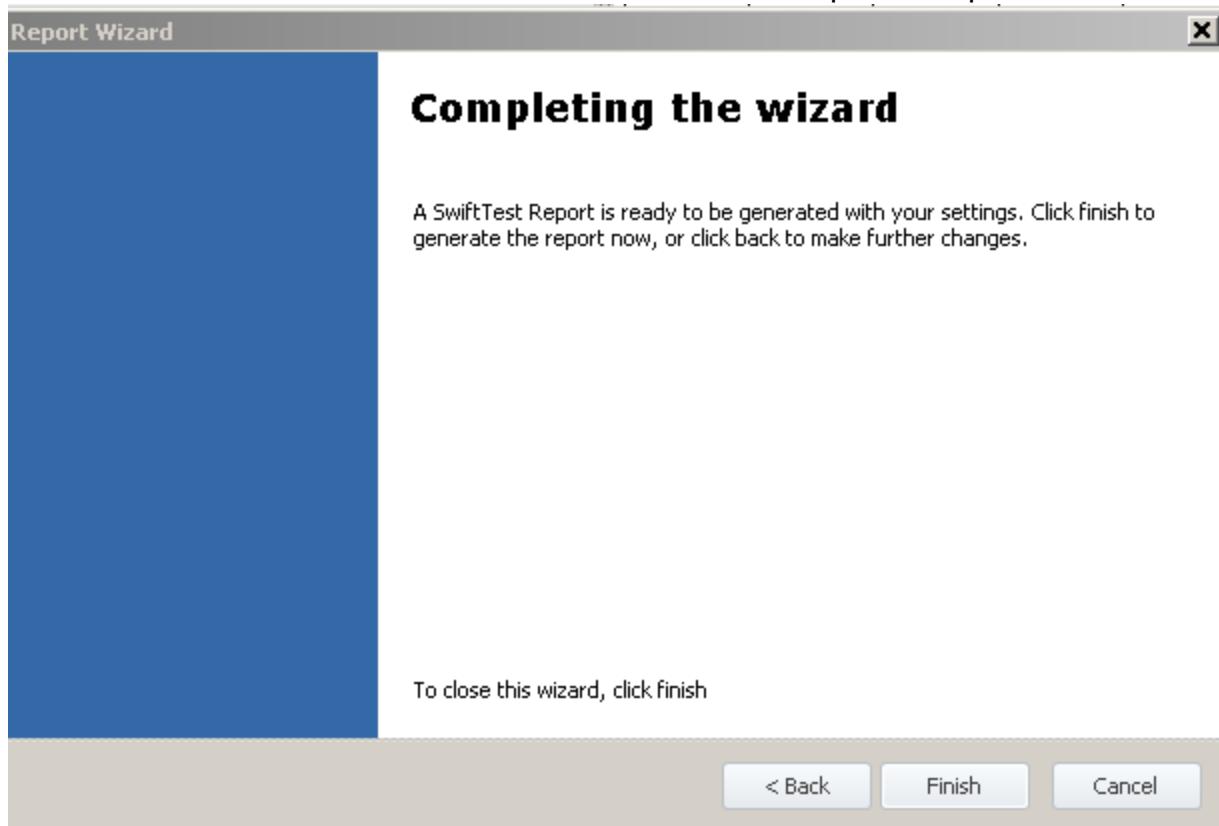
2. Project Summary



Then click to provide a company logo for the report.



Select **Next >** to finish the Wizard and then **Finish** to open the report document.



The cover sheet for this report will look like



LOAD
DYNAMIX

Test Report

Validation report for HTTPS performance

Load Dynamix

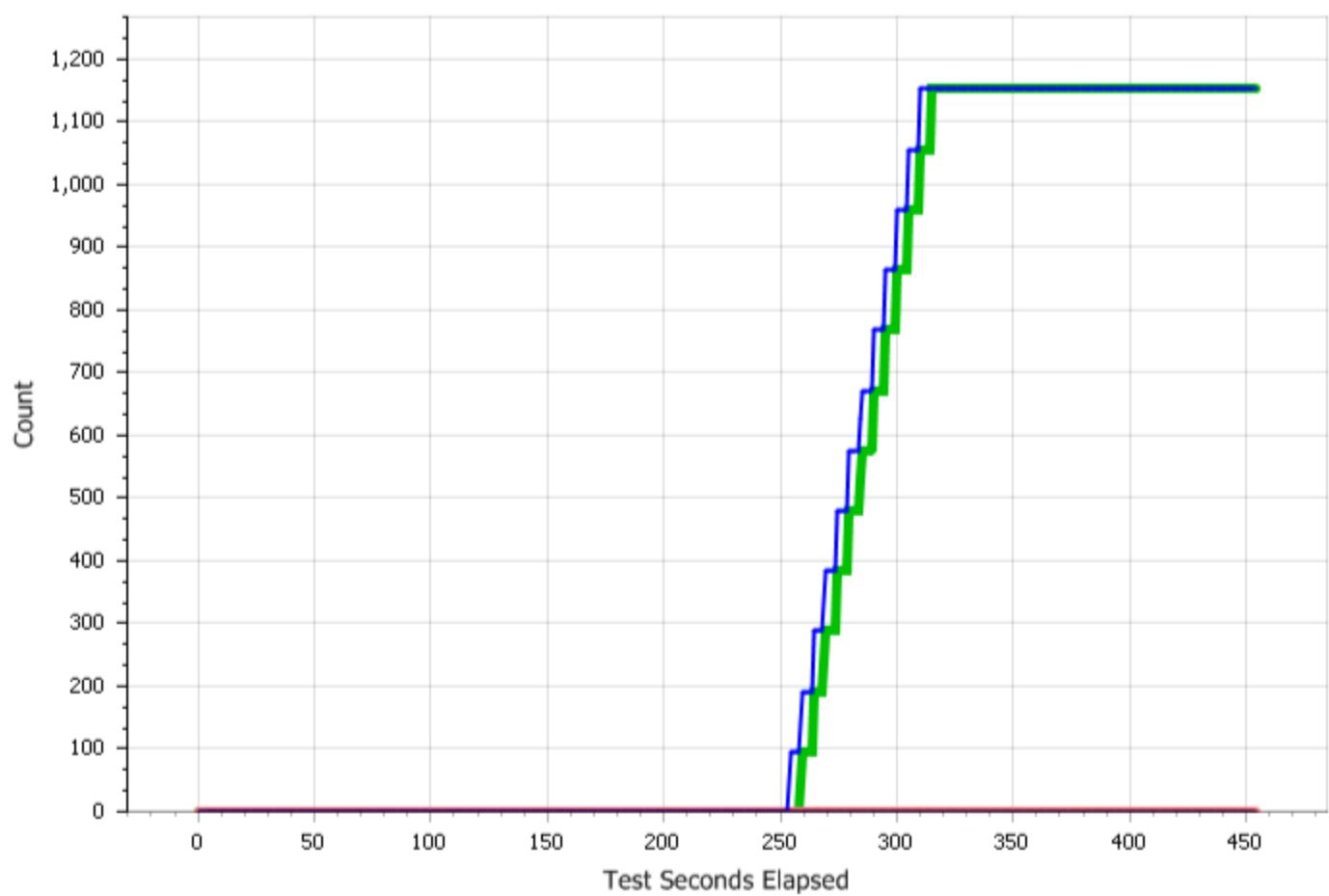
When the final report document opens on the screen, there are a set of tools presented to Tester:



The Export button can be used to create a PDF document from the report.

The report contains a page for every graph that was selected in the second step above. Below is an example - the Load Status graph.

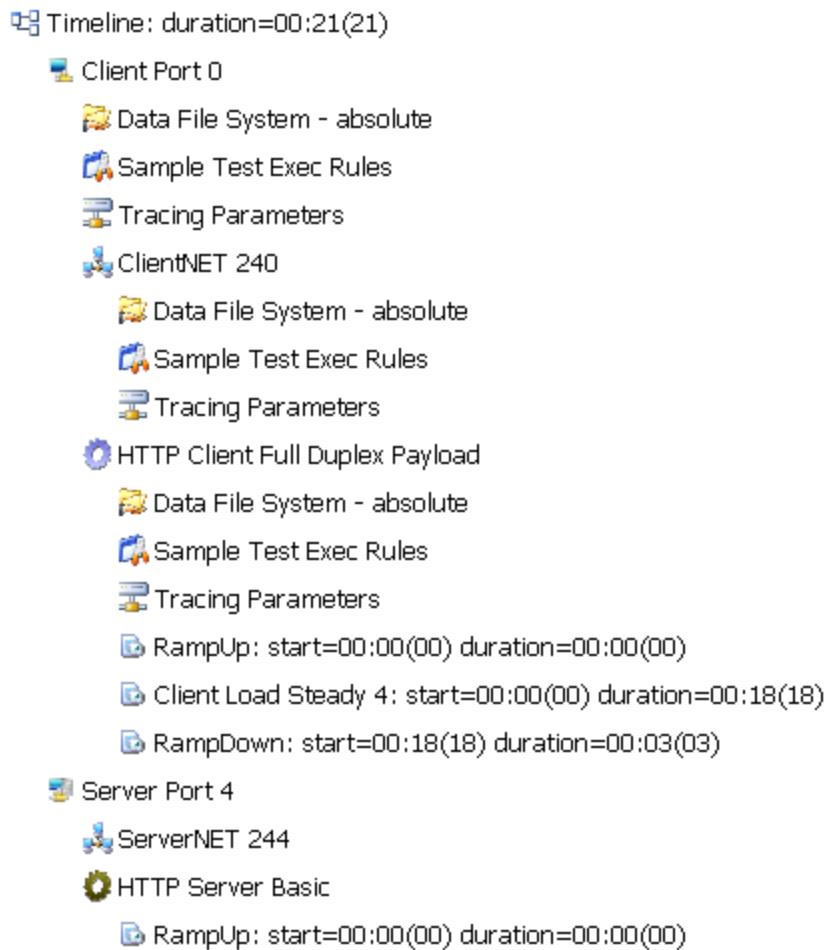
1. Test Results

Clients Total: Load Status (Scenarios, Total)

Color	Labels	Points	Name	Last Poll
■ Blue	□	□	Scenarios Attempted	1,152
■ Green	□	□	Scenarios Succeeded	1,152
■ Red	□	□	Scenarios Failed	0
■ Purple	□	□	Scenarios Aborted	0

The Project graphs will be followed by the Project Summary page

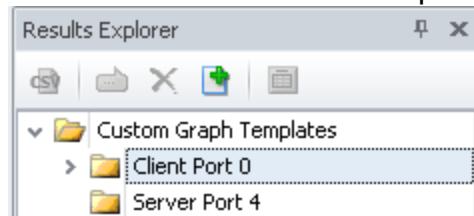
2. Project Summary



Multi Charting/Custom Graphs

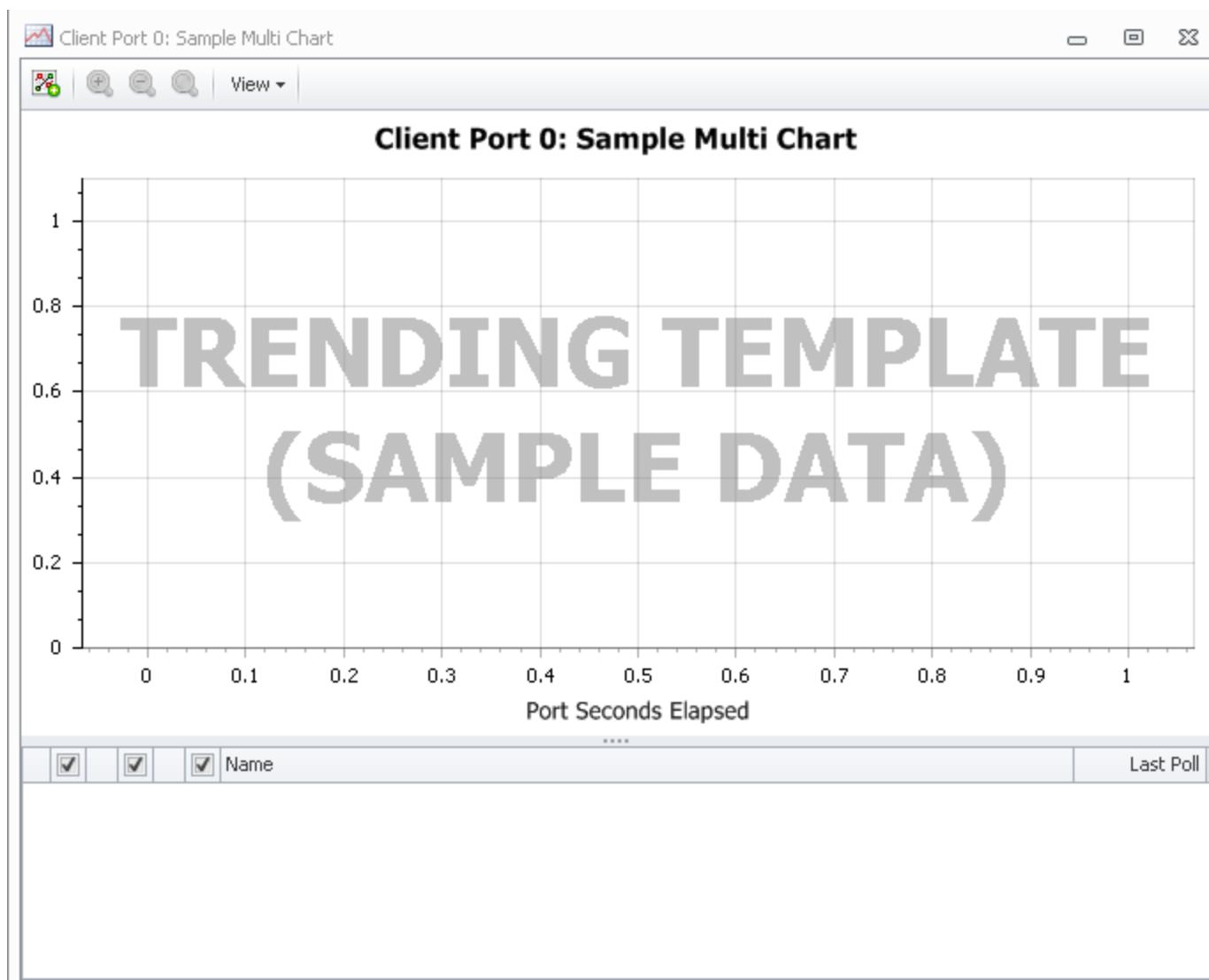
It is possible to create custom graphs using the statistics returned when a Project executes. The Custom Graph Templates folder is the location in which Custom Graphs (Tester-created) graph **Templates** are created. The Client/Server Port X or Totals Results Folder is where the **Instance** of the Template containing real data is captured. The Custom Graph Templates folder contains (at a minimum) a folder for each Logical Port in the Project. To create a Custom View, follow the steps described below:

Create the Custom View Templates in the Custom Graph Templates folder of the Results Explorer

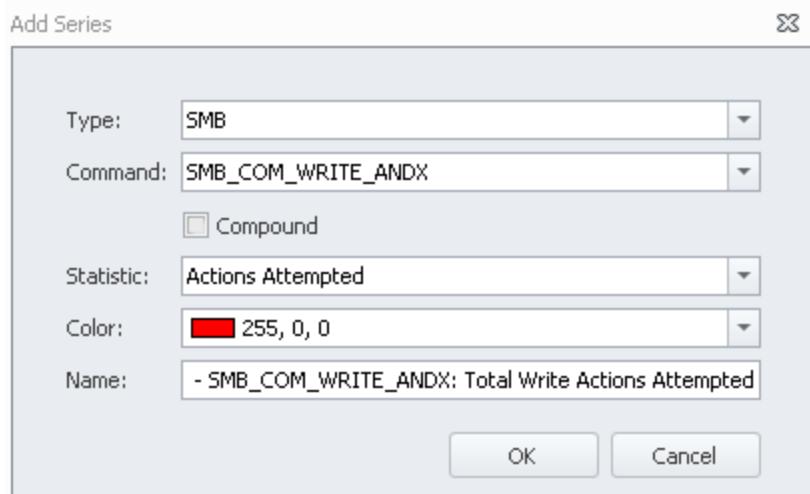


Highlight the Client Port 0 folder and click the Custom View button 

Give the Custom View Template a name (SMBWRITE) and then double click the Custom View Template to open the Custom View Editor

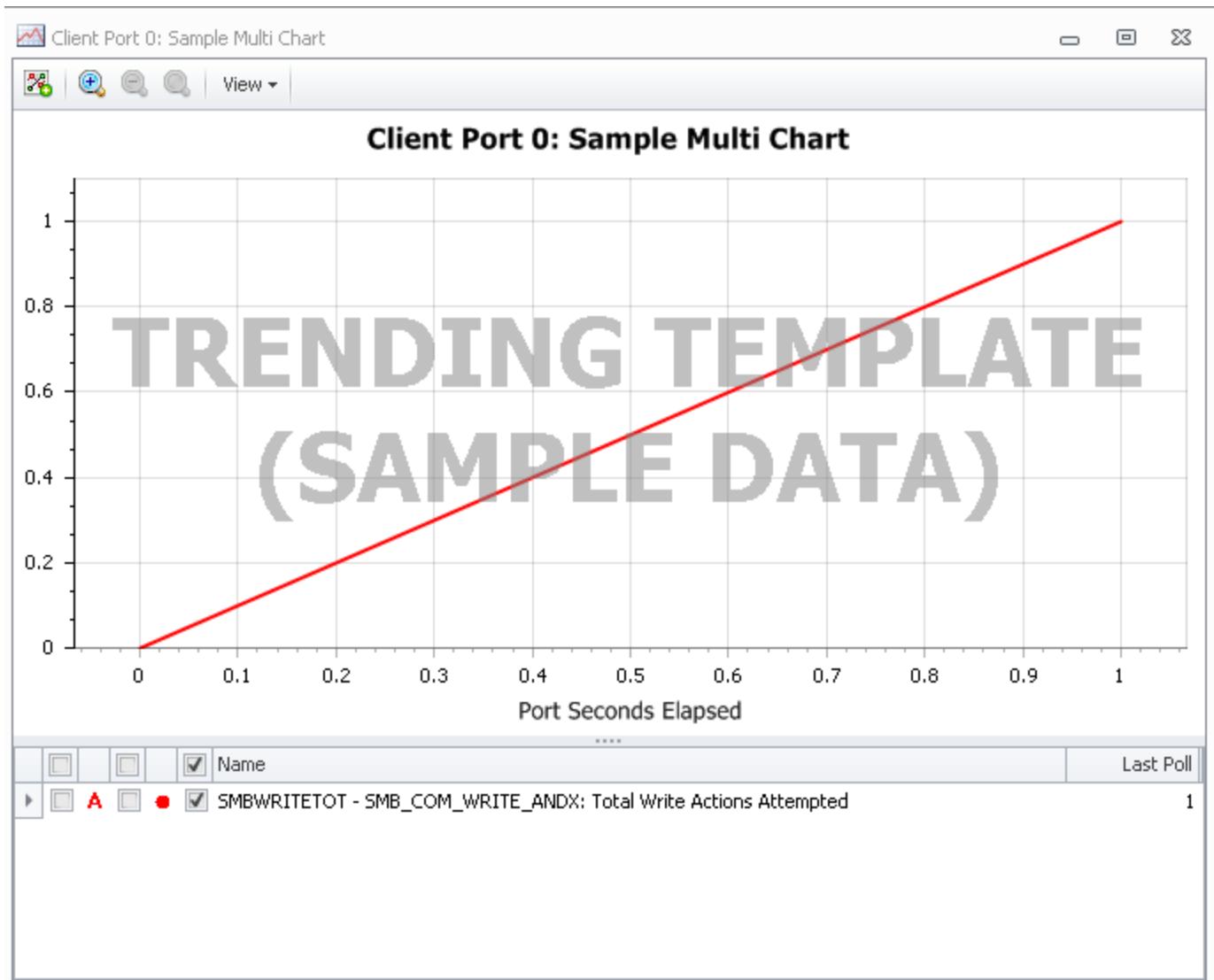


Click the Add Series button  on the left to select a statistic to be graphed. The Add Series window will open

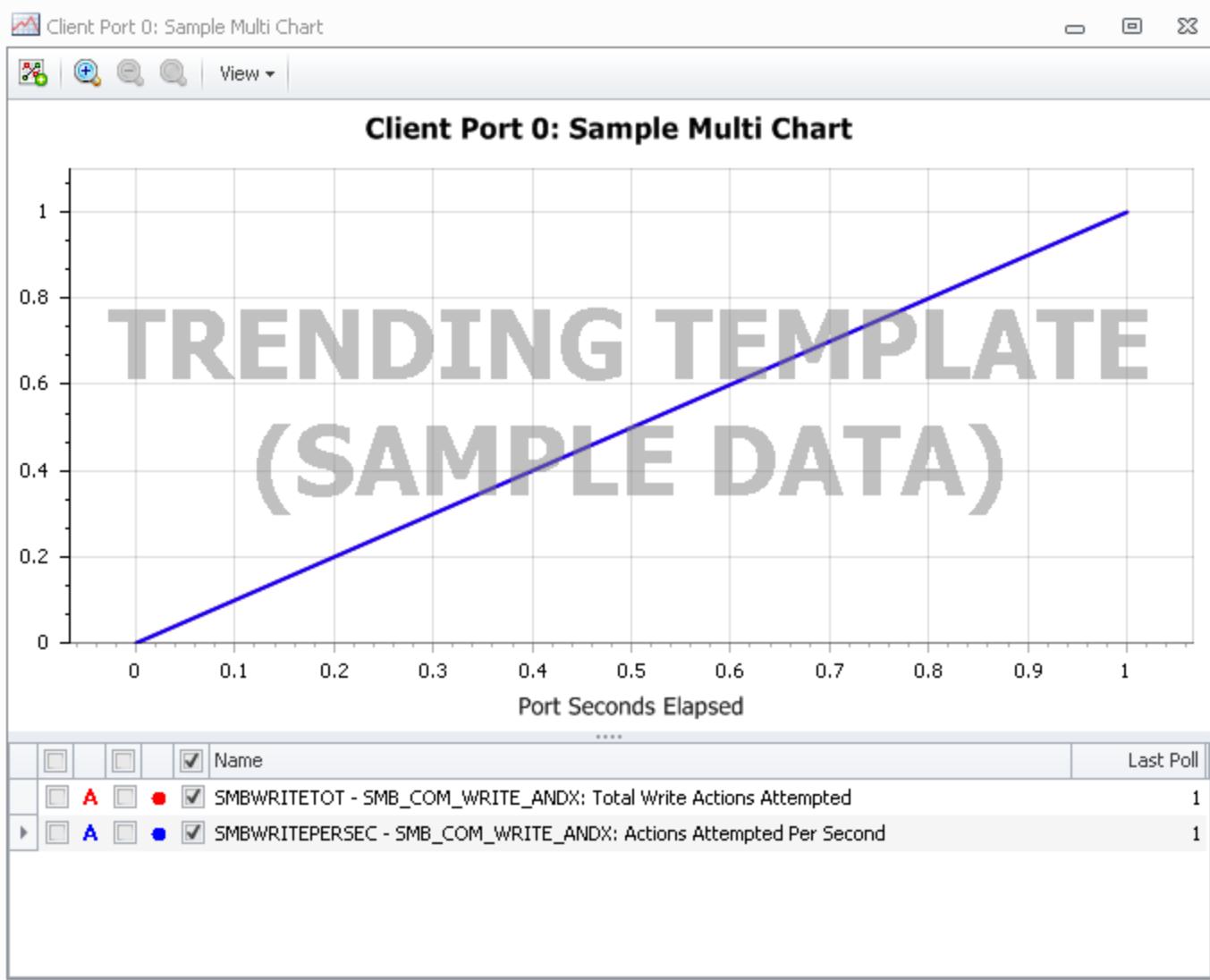


Select the Type (Protocol) and Command (SMB_COM_WRITE_ANDX), the Statistic (Actions Attempted), the Color (Red) and a name (SMBWRITETOT) and click  or the Enter key on the keyboard to complete the input. A line showing the selected color with a range from 0 to 1 and a

Data Series entry in the lower portion of the Template will appear to indicate the addition of the statistic to the Template.

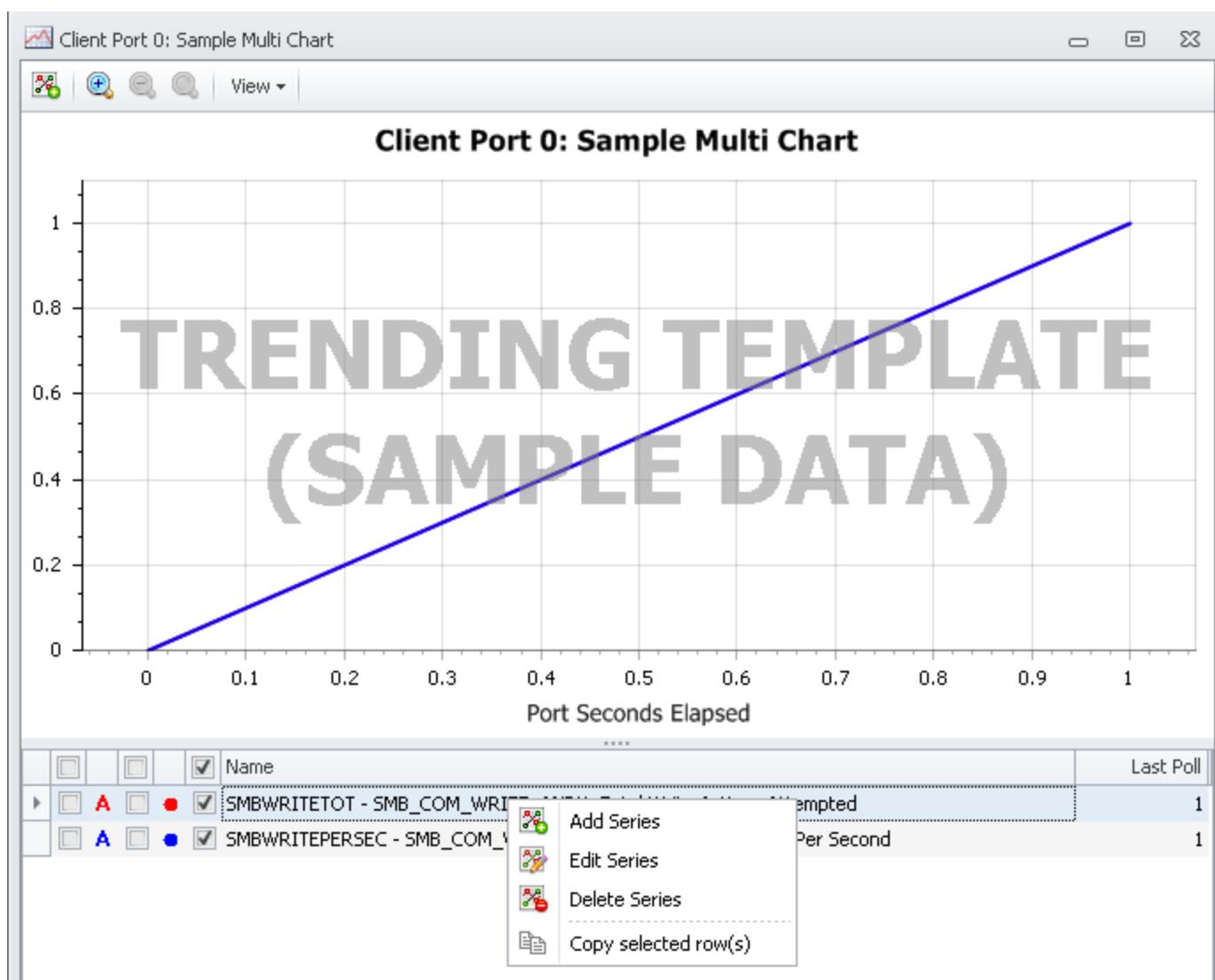


Repeat the Add Series process and add a per second data item - SMBWRITEPERSEC.



Click the in the upper right hand corner to close and save the Custom Graph.

The Data Series in the Template can be Deleted, or modified (Edited) by highlighting the desired series and doing a Right-Click

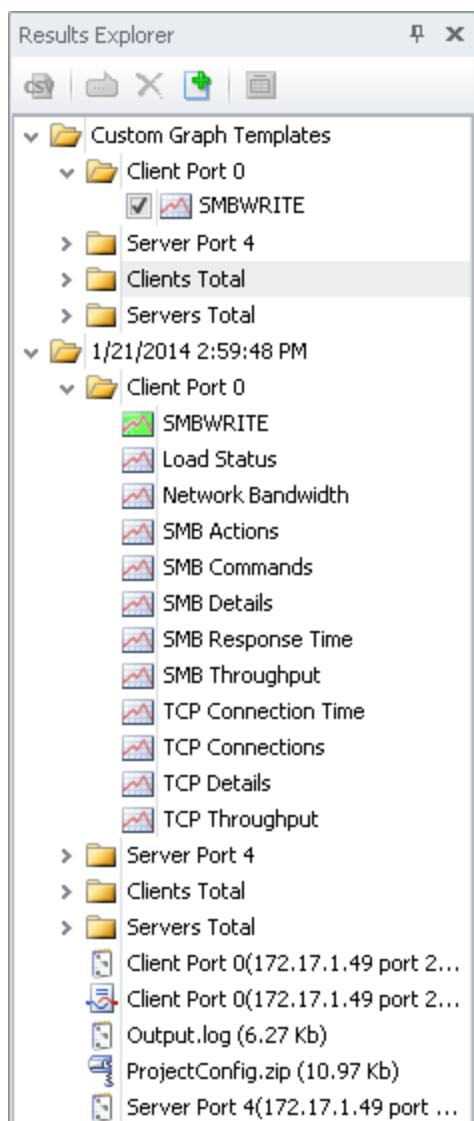


Add Series will allow another Data Series to be added to the Template

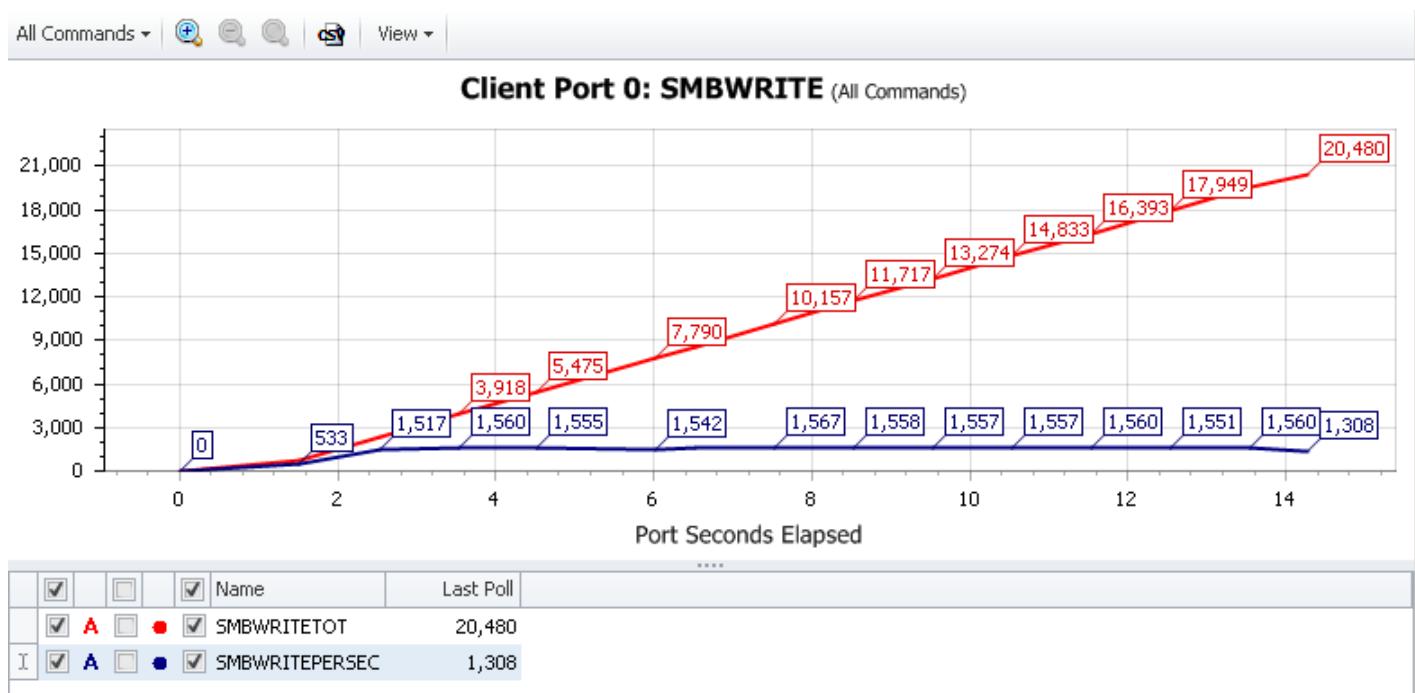
Edit Series will bring up the Data Series editor to allow this Data Series to be changed (color, name, statistic, command, protocol - any of the elements of the Data Series)

Delete Series will remove the series from the Template

Clicking the Check Box to the left of the Custom View Template will create an Instance of the Template from data in the Client Port 0 folder. If the checkbox is not clicked, no Instance of the Template will be generated, however, the Template remains in the Trending > Client Port 0 folder for later use.

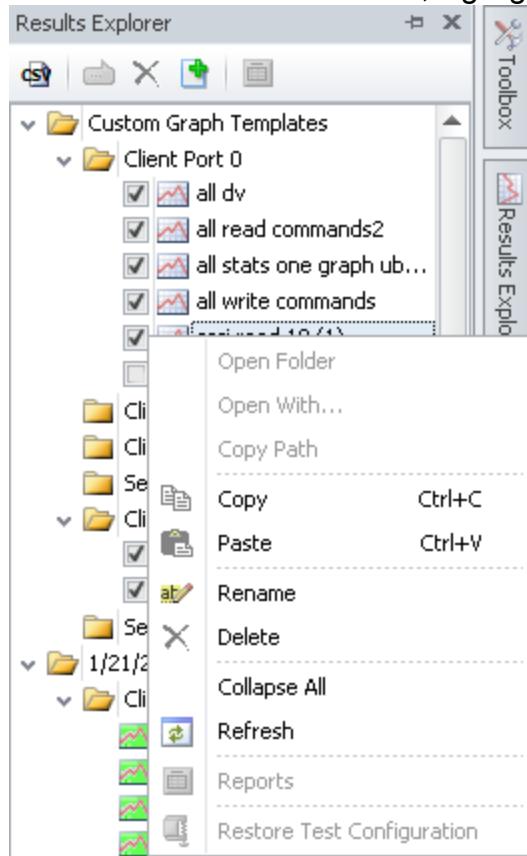


Below is an example of the graph created by this template for a 14 second run.

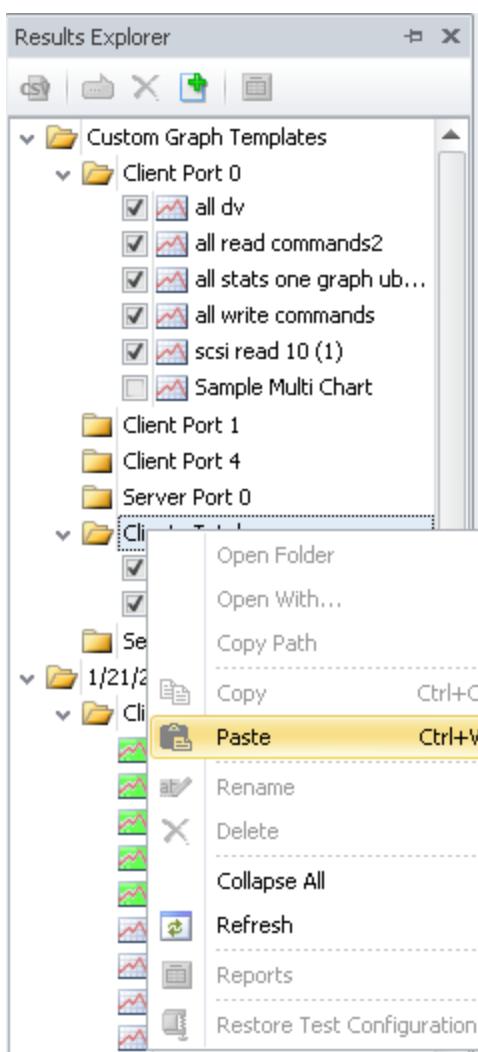


Sharing Templates between Results Folders

Templates only apply to the data from Results Folder in which they exist (ex: Client Port 0, Client Total, etc). If the same Template is used in multiple Results Folders, create the Template as described above in some Results Folder, highlight the Template and right click Copy



Highlight the Results Folder where the Template is desired and right click Paste



Note: there are Data Series that exist in Client/Server Port X Results that do not exist in Client/Server Totals. An example of this is IPv4 statistics. They exist in the Client/Server Port X Results but not in the Client/Server Totals Results. So, if a Template created in Client Port 0 that contains IPv4 Data Series is copied into Client Total Results Folder, its Instance will show IPv4 Data as NA yet the Instance of that Template in the Client Port 0 Results Folder will show the IPv4 data.

Template/Instance Settings

Graph settings (ex: Log Scale, Units, etc) apply to the Template or instance in which the setting is configured AND also the instance(s) or Template. So, make a change to the Template for Log Scale (for example), Log Scale will be applied to every Instance of that Template. Make the same Log Scale change to an instance of a Template and that change will be reflected in all instances and the Template.

Project Annotations

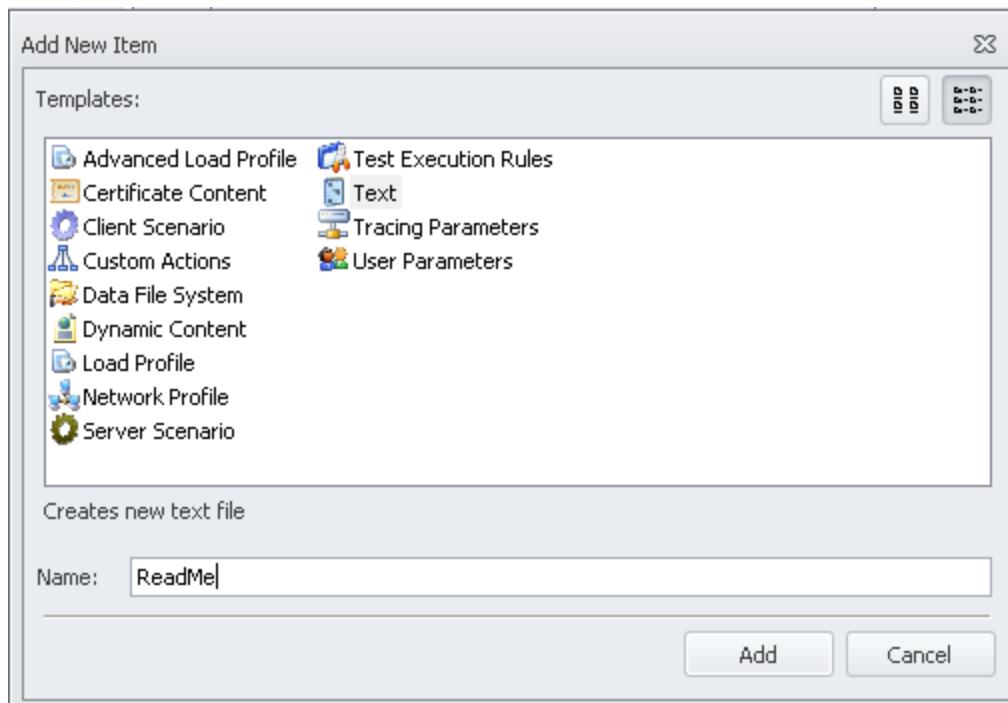
It is possible to textually annotate a test in two ways:

- TEXT Project files. Using the Add Item interface, it is possible to add a Text file to a Project.
- Comments on Results Folders.

Text Files

In the Add New Item interface, click on the Text item, give the file an appropriate name (ReadMe,

Project Name, ?) and click Add

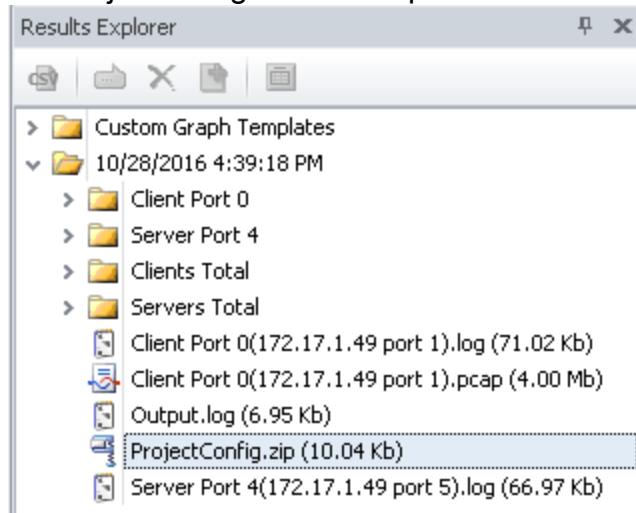


which will open a empty text file template in which you can document the Project. This file can store and text content that is desired: test description, test results (e.g. text results from Results folder client and/or server log files), assumptions, etc. Upon closing this template, it will be saved in the Project folder using whatever name it was given when Add was clicked.

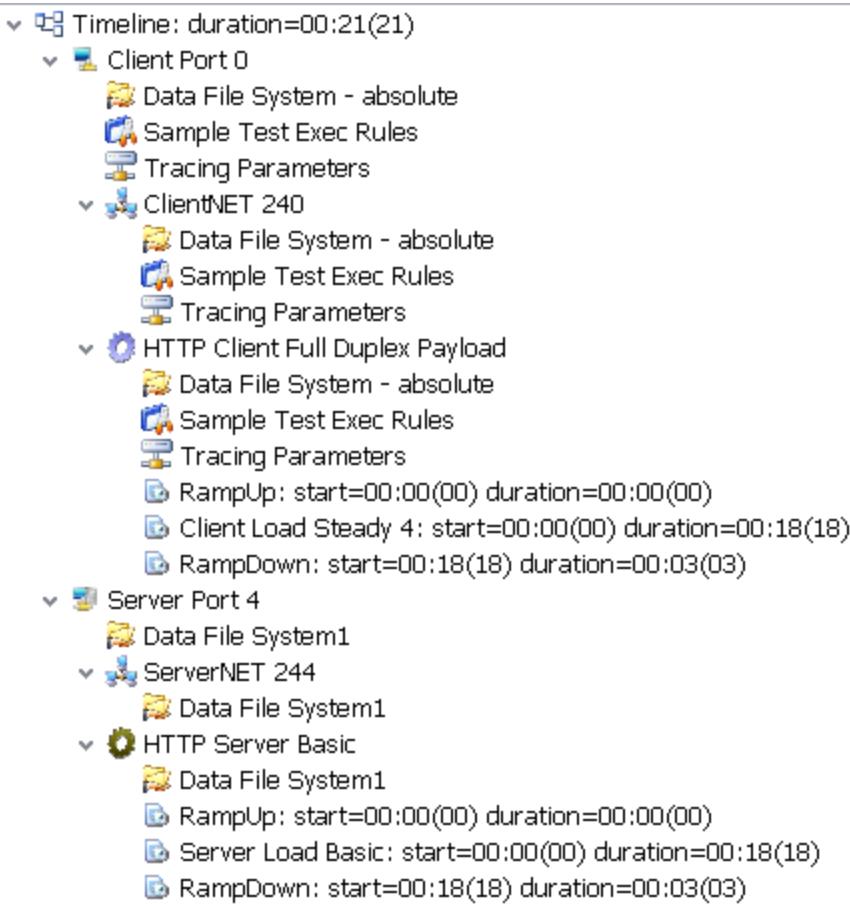
Restore Test Configuration

Whenever a Load DynamiX Project is executed (Start or Validate by TDE or LdxCmd), the configuration information of that Project run is saved in the Results Folder in case it is ever necessary to return the Project to that specific configuration.

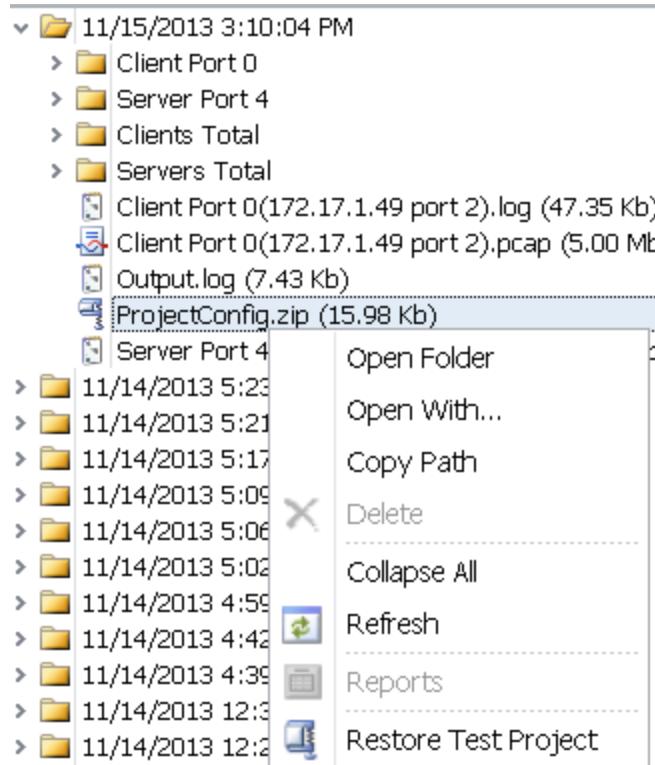
In the Results Folder is a .zip file named ProjectConfig.zip and it contains the files necessary to recreate the Project configuration that produced the results in this Results Folder.



When ProjectConfig.zip is double-clicked, it produces the Project Summary/Test Configuration report



When ProjectConfig.zip is right-clicked and **Restore Test Configuration** selected, the Project is restored to the configuration contained in the ProjectConfig.zip file.



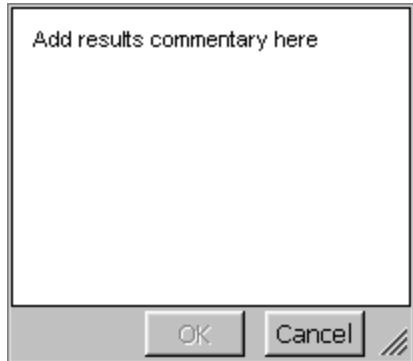
This feature is very convenient for returning a Project that has been modified to a prior configuration.

Restore Test Configuration Caveats/Notes

1. The contents of the DATA folder in a Project is not captured in the ProjectConfig.zip file because it may contain very large data files. If changes to the DATA directory have been made to a Project and an older ProjectConfig.zip is Restored, it is possible that the DataFileSystem Resource references may not be correct.
2. LdxCmd when used to Start or Validate a Project will produce a ProjectConfig.zip in the Results folder. If that Results Folder is copied into the Project Results Explorer folder then the configuration run by LdxCmd can be restored to a the instance that is open in the TDE.

Results File Comments

In the Results Explorer there are dated folders for each test execution. Each of these folders can have comments added by highlighting the folder name and clicking the Results Comments button . This button opens up a text dialog box in which test results annotations can be added.



Advanced Concepts: User Parameters

Advanced Concepts: User Parameters

OVERVIEW

All Load DynamiX-supported Actions require input to accomplish their task within the constraints of the test network and the DUT. For very simple/short Projects, it may be reasonable for the Tester to enter the information necessary to run the test successfully as fixed constants, but for any test of the size and complexity required to test today's networked storage devices and services, run-time delivery of this input is a must. One of the means to deliver input to Actions at runtime that Load DynamiX TDE provides is User Parameter files.

User Parameter files are what the name implies - a file that holds values that can be delivered to the input fields of Load DynamiX Actions while those Actions are executing. This capability allows a Tester to create test templates that are provided with the information needed to operate at run time. Thus, a single template can be used to simulate variable numbers of unique users or create variable numbers of unique files or directories, etc.

User Parameter files are best thought of as a spreadsheet. The columns in this spreadsheet contain similar information (e.g. user names, passwords, file names, read or write block sizes, etc). Each column in this spreadsheet is, generally speaking, independent of the other columns. However, in some cases, such as User names and Passwords, two columns might need to be used in lock step. User Parameter files can be created within the Load DynamiX TDE using the Add New Item interface in the Project Explorer or they may be created outside of the TDE using any tool that can generate .CSV (comma separated variable) files such as a spreadsheet tool. Either way, once added to a Project, these files may be edited (added to or changed) using the tools in the TDE.

When in use in a Project, the columns deliver information to the input fields in Load DynamiX Actions. It is up to the Tester to decide when it is necessary to use User Parameters versus other means of delivering input such as functions like @RANDOM or @LOOPTOTAL or ... (see [Appendix: Functions and Formula](#)). User Parameters cannot be used in all Action input fields. Fields that are binary (e.g. true/false, enabled/disabled, etc) do not accept User Parameters. Fields that contain fixed protocol values (e.g. SMB Create Disposition values "Open if exists, Create otherwise" or "Fail if exists, Create otherwise") do not accept User Parameters. Fields that contain file and directory name text strings, IP addresses, port numbers, file read/write size, etc do allow the Tester to deliver these values via User Parameter Files at run time.

To acquire data from a User Parameter file, the Tester inserts into Action field a reference to the User Parameter file and column that is desired. User Parameter files are either Local (directly associated with a test Scenario by being dragged onto that Scenario, Network Profile or Logical Port) or Global (available to any and all Scenarios in a Project). It is up to the Tester to decide if Local or Global (or both) User Parameters are required to address the needs for parameterized data and to use the approach that makes most sense for a given test.

LOCAL vs GLOBAL

Local User Parameter files are, as the name implies, local to the Load DynamiX TimeLine component onto which they are dragged. There can be at most one Local User Parameter file per TimeLine component (e.g. one dropped onto the Port definition, one dropped onto the Network Profile and one on a Scenario. User Parameters are only referencable in the input fields of Actions in a Scenario, but they can be dragged onto any of the components of a Project - Logical Port, Network Profile, or

Scenario. The component onto which the Local User Parameter file is dragged determines its accessibility within the Scenarios of the Project. If a Local User Parameter file is dragged onto a Scenario component then its contents are visible to the Action input fields of only that Scenario. If a Local User Parameter file is dragged onto a Network Profile component then its contents are visible to the Action input fields of all of the Scenarios associated with that Network Profile. If a Local User Parameter file is dragged onto a Logical Port component then its contents are visible to the Action input fields of all of the Scenarios associated with that Logical Port. Operationally, if all of the Local User Parameter files of a Project are the same file (i.e. the same file from the Project Explorer is dragged onto the components of a Project then it is the same file that is used during test execution. If each file is a different file (or copies of the same file) then they are treated as separate entities. Local User Parameter file references are of the form \$(<column header>) where <column header> are the typical spreadsheet column names A, B, ... A Local User Parameter reference can be made using the @UPL(<COL>) function so as to be compatible with Functions. @UPL(<COL>) behaves exactly like @UP(<INDEX>,<COL>) except that there is no need to specify the <INDEX> value for a Local User Parameter. **Local User Parameter file references are NOT allowed in Functions (e.g. it is not allowed to have a function that looks like: =@STRING(xyz) + \$(A).** However, the Function-compatible form of a local user parameter file reference, @UPL(A) is allowed in Functions. =@UPL(A) produces the same result as \$(A).

Global User Parameter files are global to all of the Logical Ports that make up a Load DynamiX Project. Global User Parameter files are created by first creating User Parameter files in the TDE or importing them from a .CSV file and then dragging these files into the User Parameter Map. As files are dragged into the User Parameter Map, they are assigned an Index and the value of that Index is used in the reference. Global User Parameters are referenced by a string of the form @UP(<index>, <column header>). There are no structural differences between Local and Global User Parameter files only in the scope of access, how the reference is specified and whether they may be used in formulas as managed by the Load DynamiX Function Editor. Global User Parameter file references are allowed in Functions.

OPERATIONAL BEHAVIOR

Whether Local or Global, the behavior of User Parameter data during the execution of a Scenario are the same with the exception of use in Functions. The presence of a simple User Parameter reference \$(<column header>) or =@UP(<index>,<column header>)) in the input field fetches from the specified file the next instance of the data element in the named column. For example \$(A) or =@UP(0,A) both pull the next element in the column named "A" from the designated User Parameter file. Within the context of an executing Scenario, repeated references to that same variable \$(A) or =@UP(0,A)) will yield the same value unless the Tester has Advanced or Reset the User Parameter file using the Advance User Parameters or Reset User Parameters Scenario Control Actions (see User Parameter File Advance and Reset section below). When the next instance of a Scenario starts executing, the pointer to the next item is incremented so that the next reference to column A will fetch the next value in column A. When the last element of the column has been used, the references begin again at the top (first element) of the column. Each column in a User Parameter file behaves as if it were a circular list - the next reference after the last item in the list is the first item in the list.

Testers have the ability to influence the order in which the values in a User Parameter file are accessed by the use of the Load DynamiX Scenario Control Actions: Advance User Parameters and Reset User Parameters. Reset User Parameters causes the next item pointer to be set to the top of the column or to the top of all or a specified column in a User Parameter file. The Advance User Parameters statement allows the test developer to increment the next item pointer for a column or entire User Parameter file. See [Appendix: Load DynamiX Scenario Control Actions](#) and the User Parameter File Advance and Reset section below for more detailed discussion of the behavior of Advance and Reset User Parameters.

User Parameter files may be accessed as described above -
`$(<column_header>)/@UPL(<column_header>)` for Local User Parameter files or
`@UP(<index>,<column header>)` for Global User Parameters. Or they may be referred to by an Alias
(a text name that would indicate the contents of a column better than "(0,A)", say for example
"User_Name" or "password" or "blocksize", etc. See the section below on Aliases for details of what
Aliases can be called, how to set them up and any constraints.

To illustrate the use of some of these concepts, the User Parameter File Tutorial below walks a Tester through the process of deciding how to address test requirements through the use of User Parameter Files.

Load DynamiX User Parameter File Tutorial

Problem Statement

The Load DynamiX Test Development Environment (TDE) provides a rich set of CIFS-SMB, NFS, HTTP and iSCSI protocol and Scenario Control Actions, many of which require user-specified information to operate as intended. Examples of this information are IP addresses, directory names, file names, etc. To make test Scenarios work, a Tester could hard code values into each field and then replicate these Scenarios as many times as necessary to meet the needs of the test. While this is a solution to the problem, it is not efficient for test development and on-going test maintenance.

User Parameter files are often used to host User ID and Password combinations. For some caveats regarding the behavior of blank entries in the User ID and Password columns, please see the [Tips and FAQ chapter](#), Project Configuration section.

A More Effective Solution

The Load DynamiX TDE's User Parameters File (UPF) is the means by which Testers can automate the delivery of information into the input fields of Actions in Project Scenarios, thus allowing the test developer to create a test template once and then feed the information that is required to make the test behave as needed during execution. The following sections of this document describe how to create and use a Load DynamiX User Parameter File in the context of a simple SMB2 test Scenario.

Note: The following icons appear in User Parameter File windows that contain multiple rows of data:

-  Arrow indicates the row currently selected
-  Asterisk denotes the end User Parameter File
-  Cursor indicates that a cell is being edited

User Parameters File Format

The Load DynamiX User Parameters File is fundamentally a table of user defined information in which columns contain similar information. For example, the following UPF contains 3 columns of information: user names, passwords and file names.

A	B	C
USER001	Pass001	FILE01.TEST
USER002	Pass002	FILE01.TEST1
USER003	Pass003	FILE01.TEST2
USER004	Pass004	FILE01.TEST3
USER005	Pass005	FILE01.TEST4
USER006	Pass006	FILE01.TEST5
USER007	Pass007	FILE01.TEST6
USER008	Pass008	FILE01.TEST7
USER009	Pass009	FILE01.TEST8
USER010	Pass010	FILE01.TEST9
USER011	Pass011	FILE01.TEST10
USER012	Pass012	FILE01.TEST11

This information could be used to provide user names and passwords to a CIFS-SMB or SMB2 Session Setup operation and file names to CIFS-SMB or SMB2 Create File operation. This example UPF comes from the Load DynamiX TDE Project Library CIFS-SMB Full Duplex Payload sample test delivered with each Load DynamiX TDE release.

Note that this UPF has column names to document the content of each column. Column names can be provided via the import of CSV files by inserting a row at the top of the file with column names. An existing UPF can be given names by exporting it to a CSV, inserting the column names row and then re-importing that CSV file. However, UPF columns are always referred to by their canonical names: A, B, C, D, etc.

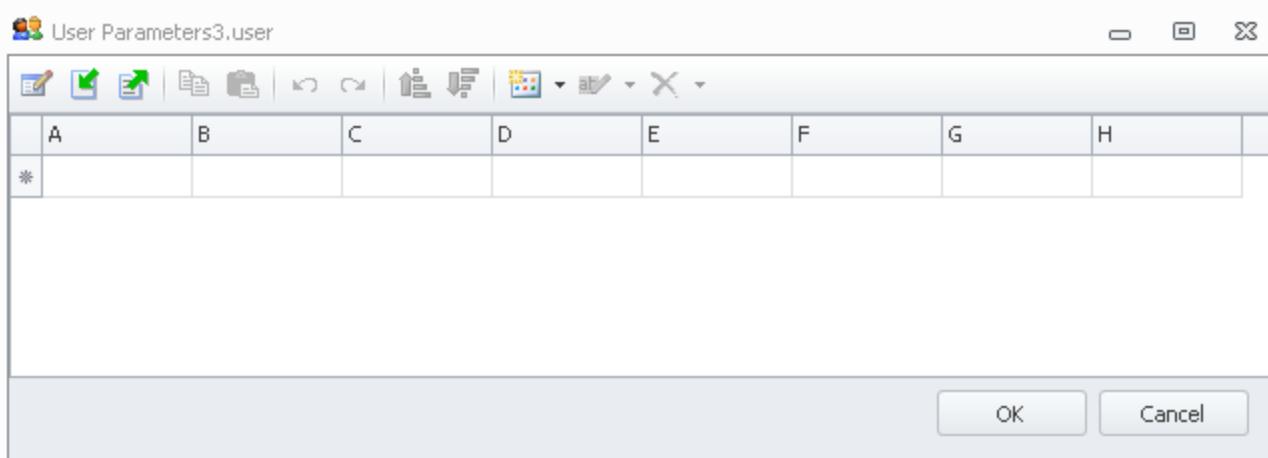
There are other test Scenarios in which this information could be used. For example, NFS operations: Lookup, Create File, Remove File, Get File Attribute could use the file names in column C as input.

Creating/Editing User Parameters Files

User Parameters Files can be created by hand or imported as a .CSV (Comma Separated Variable) file using the Load DynamiX TDE GUI. To create by hand or import, open a Load DynamiX test Scenario and click on the Add New Item button in the Project Explorer tool bar



or select Add New Item in the Project drop down menu. The empty user parameter file below will be opened for user input.

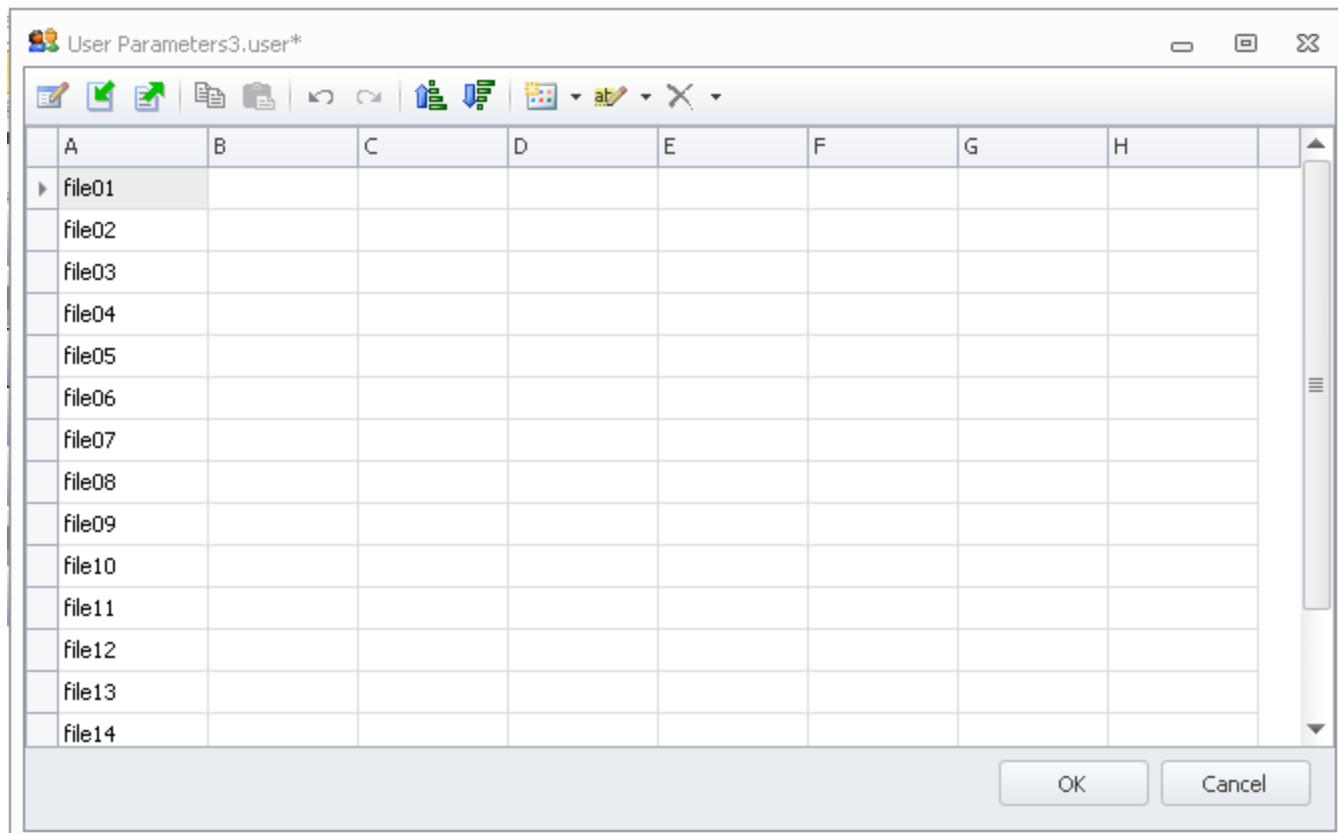


To create a UPF by hand, type the information required into the appropriate rows and columns. A much more expeditious method of creating a UPF is to create the data with a spreadsheet tool like Microsoft Excel and import that information. If the information is created by a spreadsheet tool, it must be saved in .CSV format to be imported.

To import a .CSV file, click on the Import icon



Clicking this icon will open a Windows file browser. Use it to find the desired .CSV file. Double clicking on the desired .CSV file will import it into the blank UPF. In the example below, a .CSV file containing one column of information was imported. It contains 14 file names in column A.



When this new UPF is closed (by clicking the X box in the top right hand corner or clicking the **OK** button), the file is saved in the test Project as a file named (in this case) User Parameters1.user. This file can be renamed to something more descriptive if desired by highlighting

the file in the Project Explorer area and clicking on the Rename function in the Project Explorer tool bar.



Using the Rename function, rename the UPF “File names”.

Name	Type
Resources	
Client Load Max Connections	LoadProfile
ClientNET 160 Legacy UP test	NetworkProfile
ClientNET 162	NetworkProfile
ClientNET 163 Global UP Test	NetworkProfile
ClientNET 164 Global UP Test	NetworkProfile
Demo Global UP access1	ClientScenario
Demo Legacy UP access1	ClientScenario
Demo Legacy UP access2	ClientScenario
Demo Legacy UP access3	ClientScenario
File names	UserParameters

So far, a User Parameters File has been created with a set of file names. Those file names may be sufficient for some set of tests. For the purpose of this tutorial, suppose that it is required that a user name and password is also supplied. Those names could be hard coded into the test Scenario or they could be supplied by a UPF. To add those user names and passwords to the file that was just created, double click on the file name in the Project Explorer window on the left. The file will be opened.

There are two user names and passwords that are required by the test so add the users in column C and the passwords in column D.

A	B	C	D	E	F	G	H
file01		user1	pass1				
▶ file02		user2	pass2				
file03							
file04							
file05							
file06							
file07							
file08							
file09							
file10							
file11							
file12							
file13							
file14							

As the test design continues, two new requirements are added to be able to vary the number of times a set of operations is executed and to insert some delay in the Scenario execution. There is a need now for loop control and delay information in the User Parameters File. Add a column of information into this file in column F for loop control and H for delay. With values for 1, 5 and 10 in F and 1000,2000,3000,4000 in H. The resulting file will look like this:

A	B	C	D	E	F	G	H
file01		user1	pass1		1		1000
file02		user2	pass2		5		2000
file03					10		3000
file04							4000
file05							
file06							
file07							
file08							
file09							
file10							
file11							
file12							
file13							
file14							

Click the **OK** button and save the file. Now the final changes are in place for the User Parameters File to be used to control a test Scenario.

User Parameter Files Deployment

For the purposes of demonstrating User Parameters File use and behaviors, suppose that there is a need to develop an SMB2 test with the following requirements:

- Creates one or more files per test Scenario execution, the names of which must all end in an odd number
- These files must be created by one of two users, “user1” and “user2”
- These files must be created in an repeating pattern of 1, 5 and 10 creations per test Scenario execution.
- There must be 1,2,3 and 4 second delays at the end of each test Scenario execution.

The Load DynamiX Project that follows implements these requirements using the “File names” UPF that was just created. While much of the information required by this Scenario could be provided by a UPF, this example focuses on the use of file names, login information (user names and passwords) and execution (loop and delay) controls.

There are two ways in which User Parameters Files can be used within Scenarios. This tutorial demonstrates how both Local and Global User Parameter Files are used.

Local User Parameters File deployment

Note: Local User Parameter references (e.g. \$(A)) may not be used in Functions (i.e. =@\$(A)+@\$(B) is not allowed).

Local UPF deployment requires associating a UPF with a specific component of a test Scenario. The UPF can be associated with a Scenario, a Network Profile or a Logical Port simply by dragging and dropping the UPF onto that component. At most, a Scenario can have one UPF deployed on it. If multiple UPF are required, they can be deployed on separate elements of the Project (e.g. Network Profile or Logical Port) and their behavior becomes more complex. For the purposes of this tutorial, a single UPF is used but the more complex behavior of multiple local UPFs is described in the behavioral information at the beginning of this section.

Local UPF references are of the form \${<column name>}, where <column name> is A, B, C, D, etc. Each column of a UPF behaves independent of the other columns but, fundamentally, each time a Scenario begins execution, the next row of the UPF becomes the current row and its contents become the current values for a reference to a specific column. Using the File names UPF and normal sequential access (i.e. no use of the Advance User Parameters or Reset User Parameters Scenario Control Actions), the first reference yields:

A=file01, C=user1, D=pass1, F=1, H=1000

The next Scenario execution would yield second row references:

A=file02, C=user2, D=pass2, F=5, H=2000

The third Scenario execution would yield:

A=file03, C=user1, D=pass1, F=10, H=3000

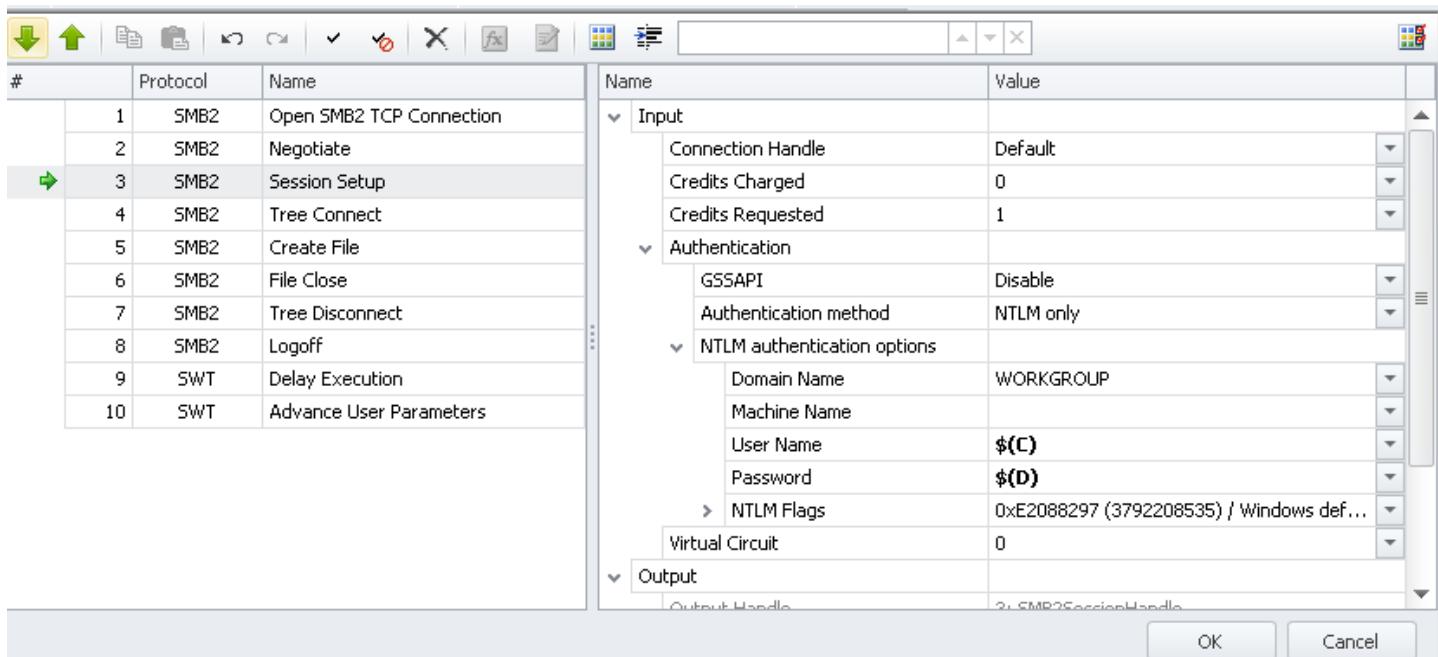
Notice that Columns <C> and <D> have cycled back to beginning because they do not have more than two rows of data. This is the normal behavior when a column reaches the end of the information in it. Likewise for column F at the next reference.

The fourth Scenario execution yields:

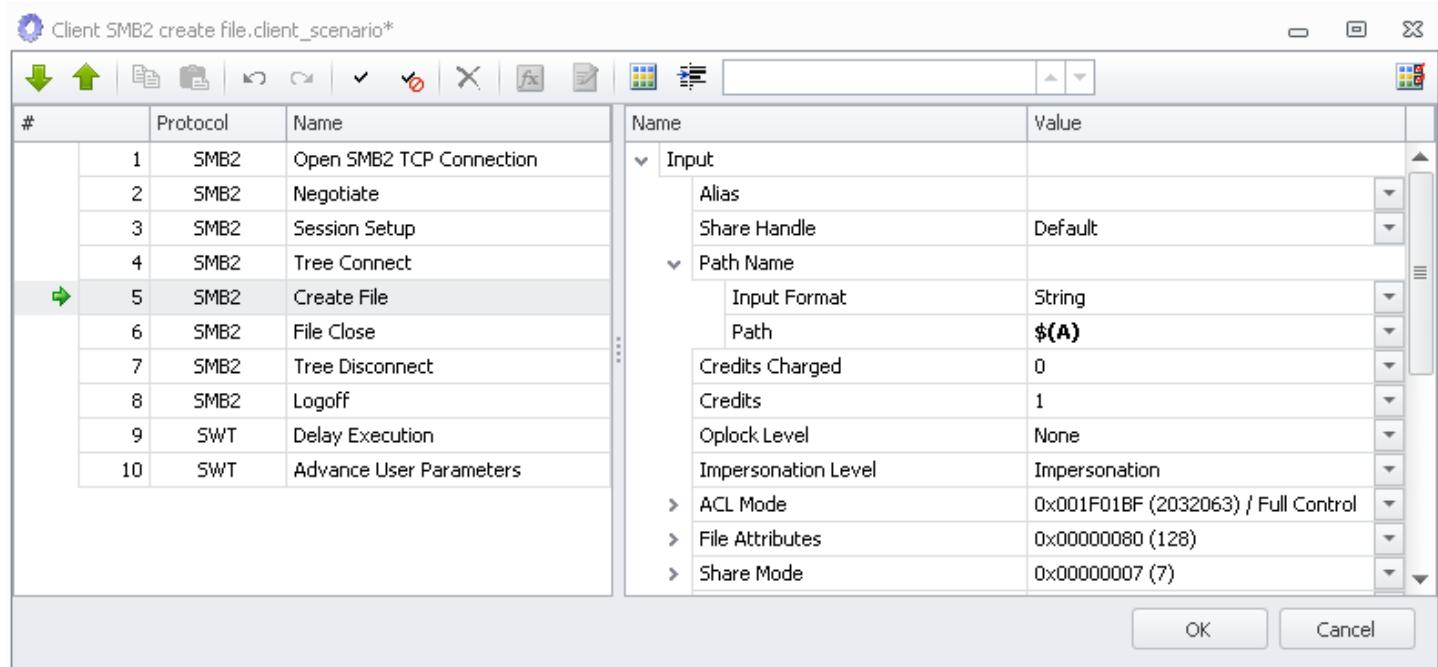
A=file04, C=user2, D=pass2, F=1 and H=4000

and so forth. The requirements for this test demanded the use of file names that always ended in an odd number yet our UPF contained file names that end in both even and odd numbers. The Advance User Parameters Scenario Control Action provides a means to skip the even number file names in Column <A> by inserting an extra advance of Column <A> at the end of each Scenario execution. The next few screen shots illustrate the local style references in the context of the SMB2

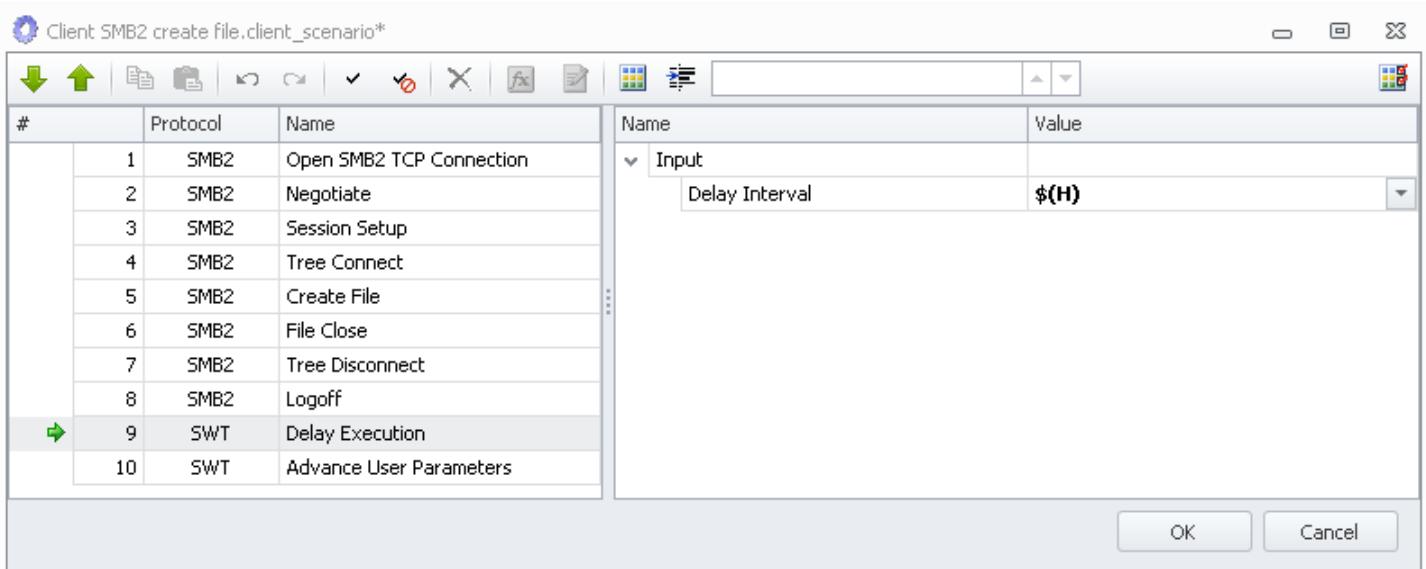
Session Setup Action (providing user name and password for login)



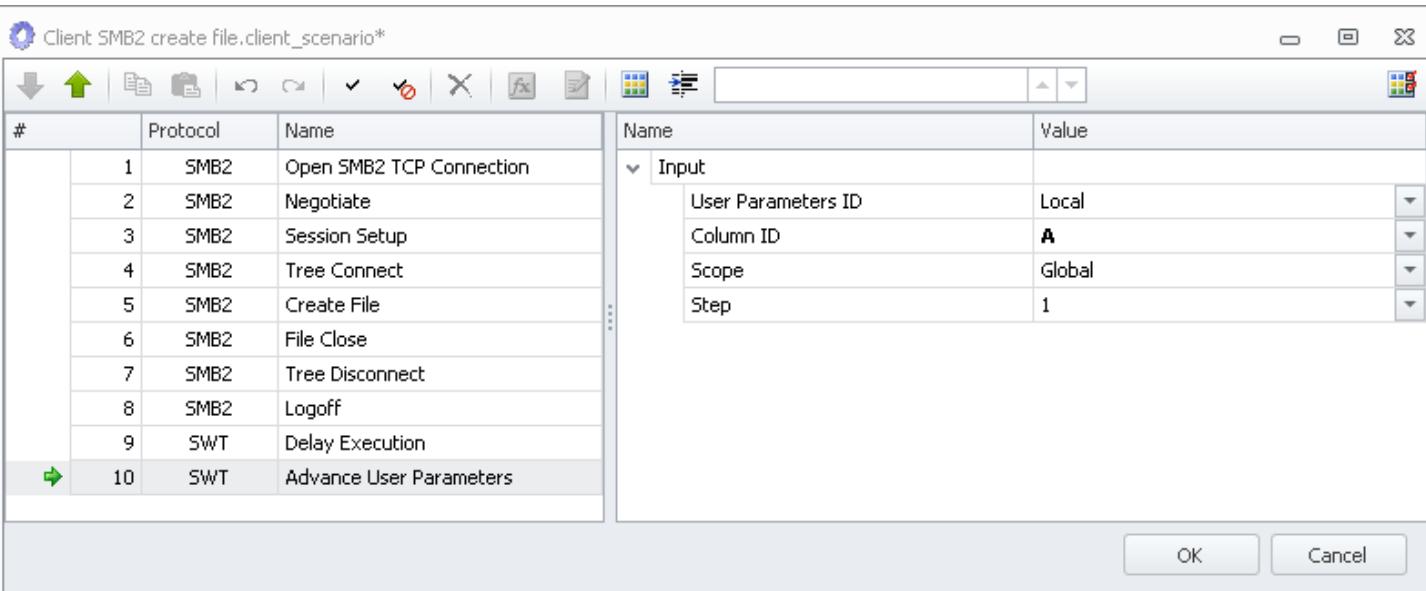
Create File (providing file name)



Delay Execution (providing the delay interval in milliseconds)



Advance User Parameters (specifying the Column to advance)



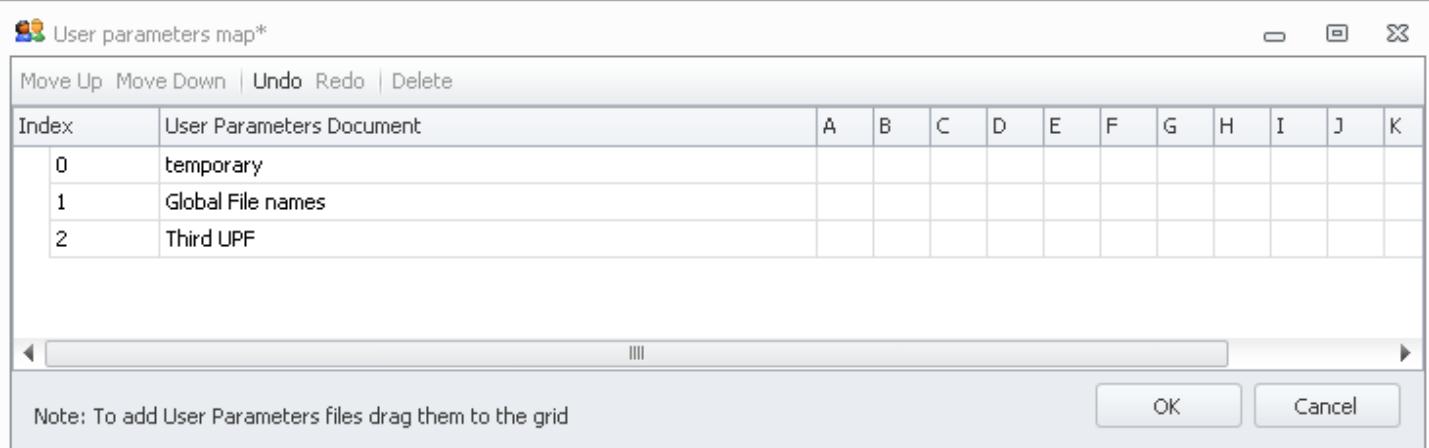
Execute the test Scenario and look at the packet capture information using Wireshark with a display filter for SMB2 Create commands and destination IP address 172.16.170.1 to see that the desired behavior was achieved

No. .	Time	Source	Destination	Protocol	Info
22	0.000976	172.16.160.100	172.16.170.1	SMB2	Create Request File: file01
61	0.002376	172.16.160.101	172.16.170.1	SMB2	Create Request File: file03
69	0.002452	172.16.160.101	172.16.170.1	SMB2	Create Request File: file03
77	0.002521	172.16.160.101	172.16.170.1	SMB2	Create Request File: file03
85	0.002590	172.16.160.101	172.16.170.1	SMB2	Create Request File: file03
93	0.002657	172.16.160.101	172.16.170.1	SMB2	Create Request File: file03
132	0.005021	172.16.160.102	172.16.170.1	SMB2	Create Request File: file05
140	0.005089	172.16.160.102	172.16.170.1	SMB2	Create Request File: file05
148	0.005161	172.16.160.102	172.16.170.1	SMB2	Create Request File: file05
156	0.005229	172.16.160.102	172.16.170.1	SMB2	Create Request File: file05
164	0.005296	172.16.160.102	172.16.170.1	SMB2	Create Request File: file05
172	0.005396	172.16.160.102	172.16.170.1	SMB2	Create Request File: file05
180	0.005499	172.16.160.102	172.16.170.1	SMB2	Create Request File: file05
188	0.005567	172.16.160.102	172.16.170.1	SMB2	Create Request File: file05
196	0.005642	172.16.160.102	172.16.170.1	SMB2	Create Request File: file05
204	0.005710	172.16.160.102	172.16.170.1	SMB2	Create Request File: file05
243	0.009072	172.16.160.103	172.16.170.1	SMB2	Create Request File: file07
282	0.013459	172.16.160.104	172.16.170.1	SMB2	Create Request File: file09
290	0.013535	172.16.160.104	172.16.170.1	SMB2	Create Request File: file09
298	0.013609	172.16.160.104	172.16.170.1	SMB2	Create Request File: file09
306	0.013676	172.16.160.104	172.16.170.1	SMB2	Create Request File: file09

Only files with odd numbers (Advance User Parameters) in the name are used and large increments in time between Scenario iterations (Delay Execution) are observable as well as the obvious loops of size 1, 5 and 10.

Global User Parameters File deployment

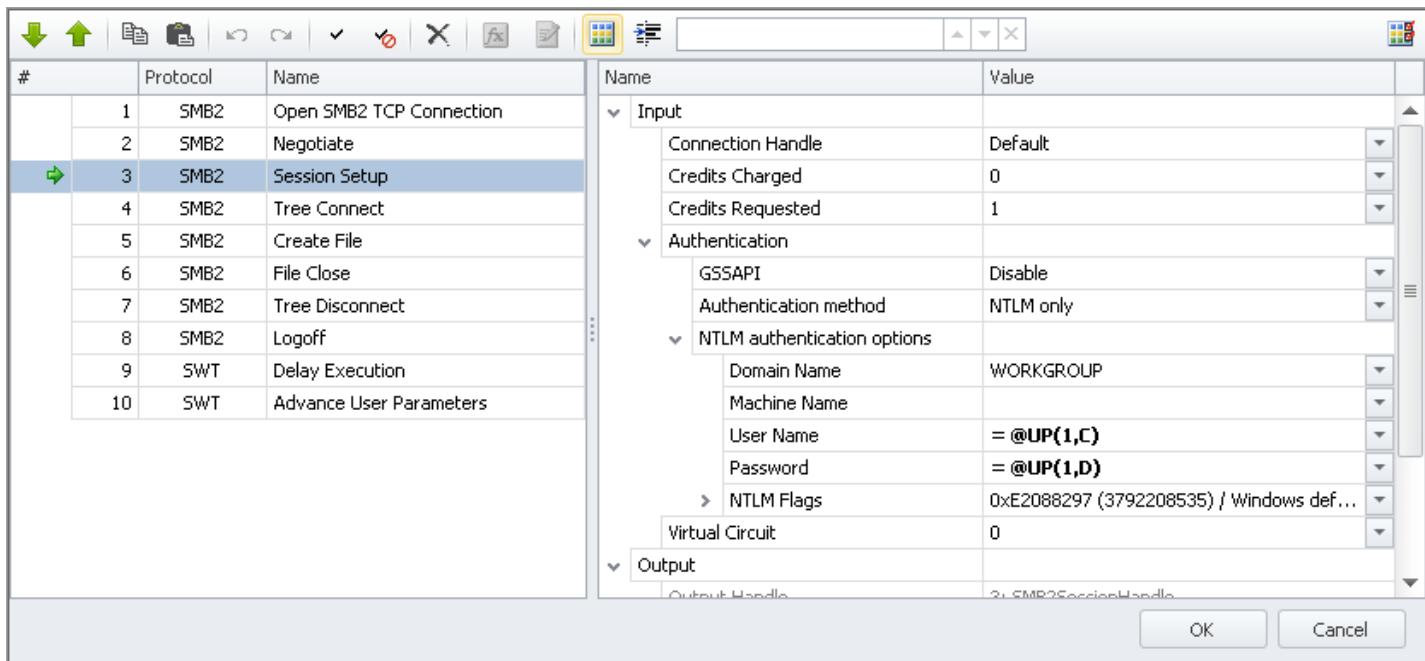
Global UPF must be registered in the User parameters map. To do this, first double click the User parameters map icon in the Project Explorer window which will open it as a window and then drag instances of UPF from the Project Explorer window into the User parameters map window. As each instance is dropped onto the window, it will be given a index number starting at 0, incrementing by one. In the example below, a copy of the “File names” UPF called “Global File names” has been dropped into the User parameters map file as index #1.



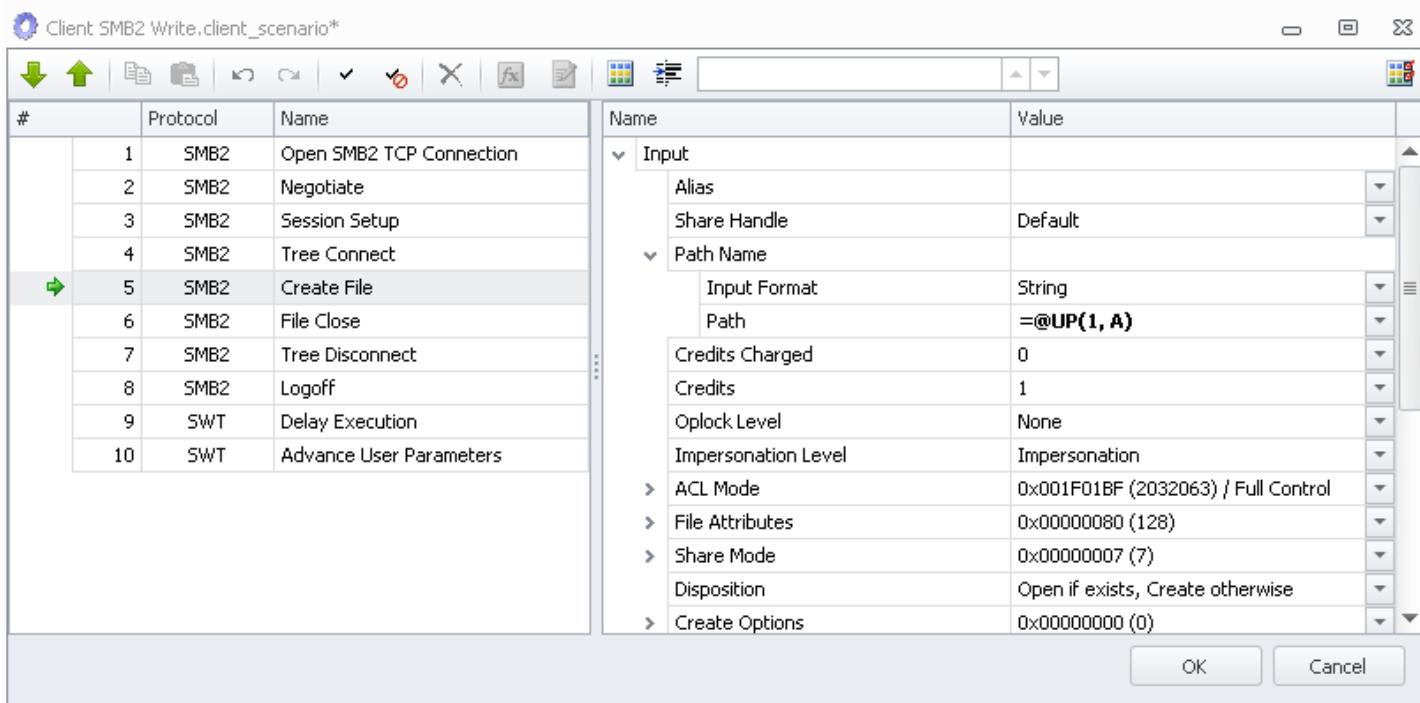
References to Global UPF data is of the form =@UP(X,Y) where X is the index into the User parameters map.

So, a test Scenario using Global UPF that will produce the same results as the Local User Parameter file example above would look like:

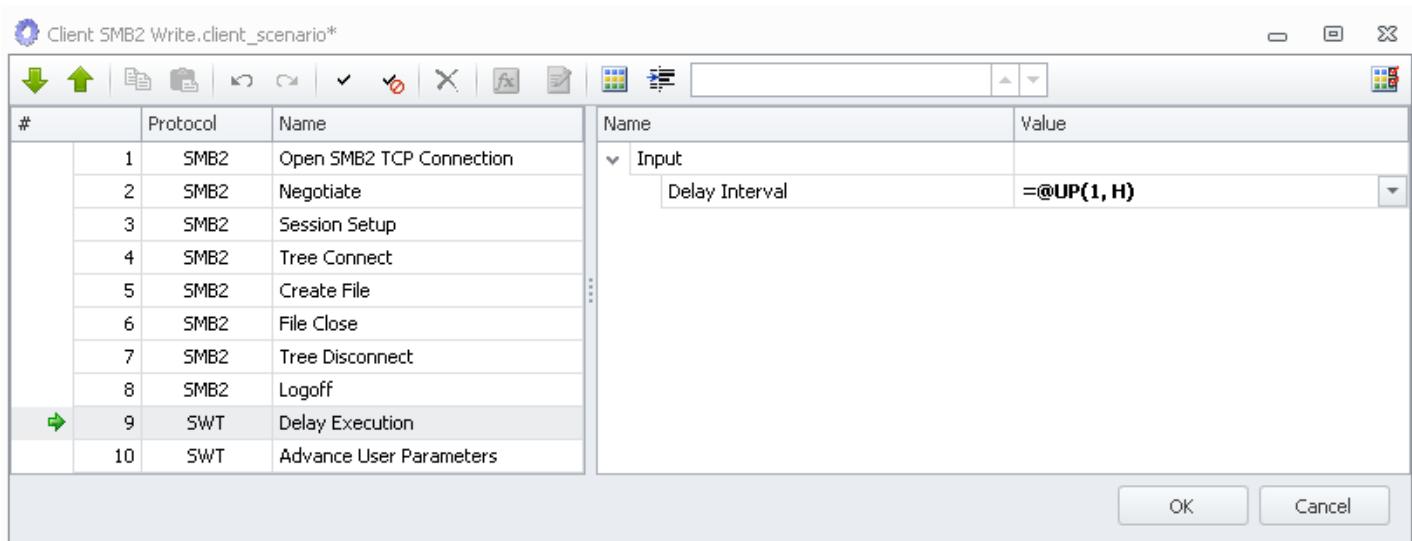
Session Setup:



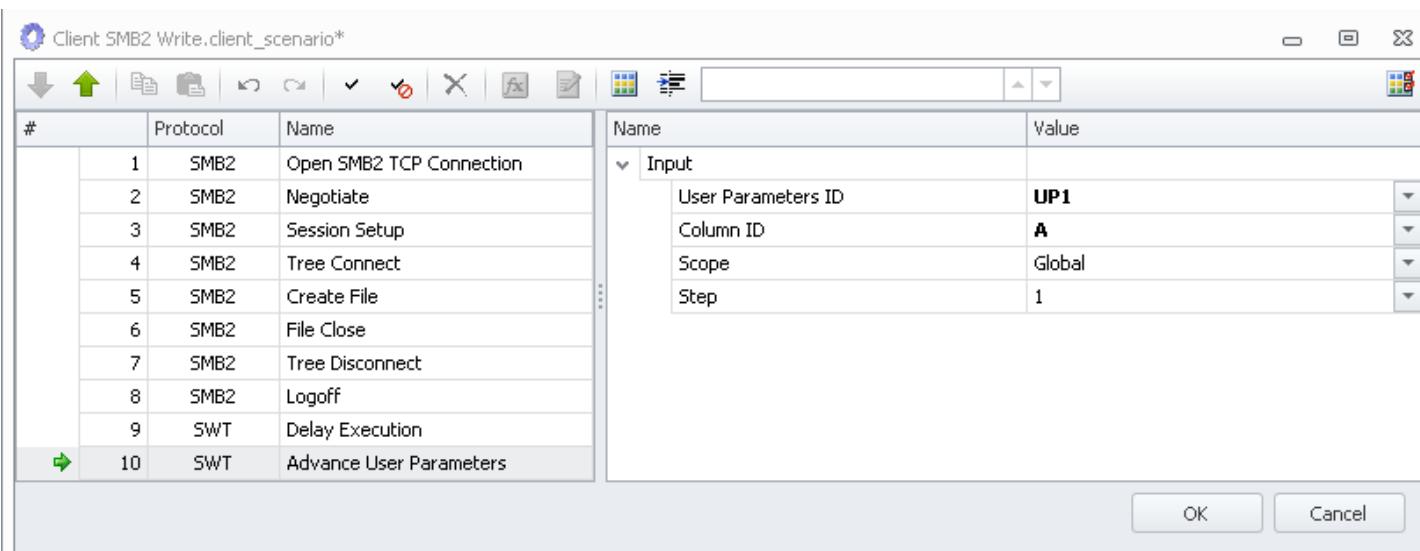
Create File:



Delay Execution:



Advance User Parameters:

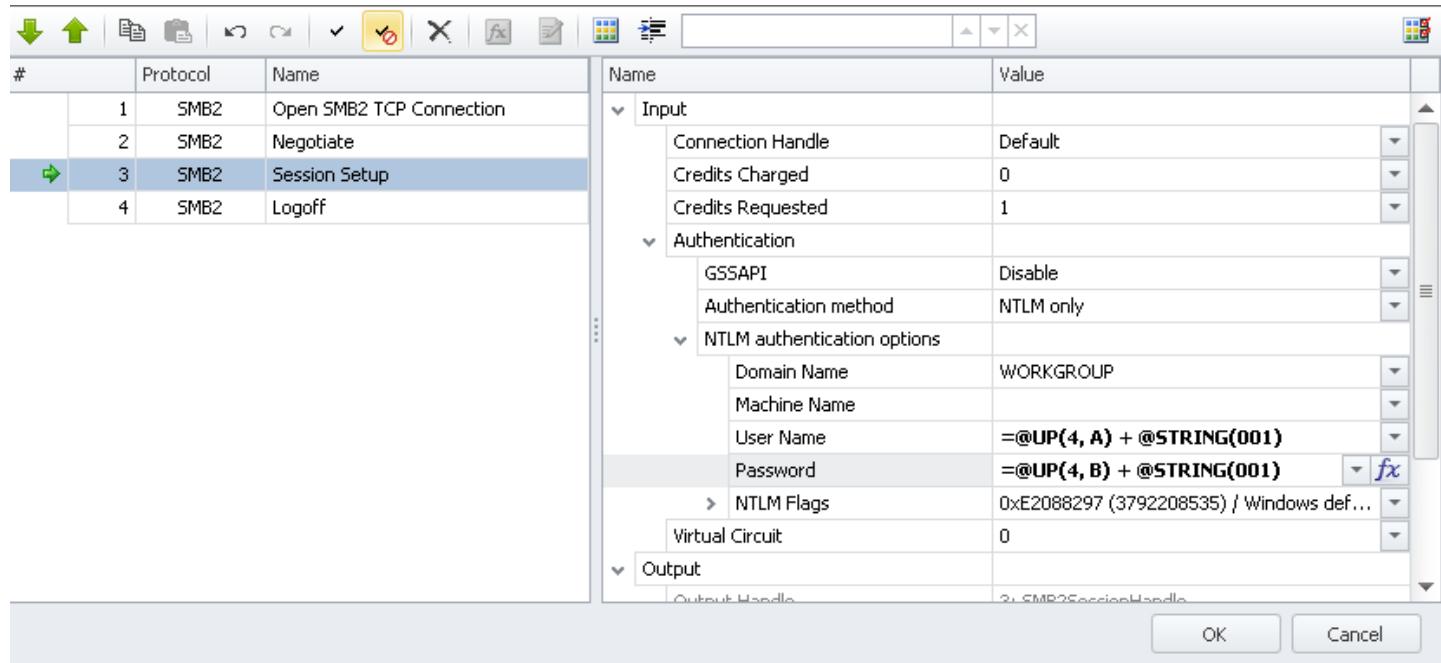


Review the packet capture of the execution of this test Scenario by Wireshark with a display filter for SMB2 Create commands and destination IP address 172.16.173.1 to see that the desired behavior was achieved

No. .	Time	Source	Destination	Protocol	Info
22	0.000878	172.16.163.100	172.16.173.1	SMB2	Create Request File: file01
61	1.001410	172.16.163.101	172.16.173.1	SMB2	Create Request File: file03
69	1.001486	172.16.163.101	172.16.173.1	SMB2	Create Request File: file03
77	1.001556	172.16.163.101	172.16.173.1	SMB2	Create Request File: file03
85	1.001624	172.16.163.101	172.16.173.1	SMB2	Create Request File: file03
93	1.001692	172.16.163.101	172.16.173.1	SMB2	Create Request File: file03
132	3.002300	172.16.163.102	172.16.173.1	SMB2	Create Request File: file05
140	3.002368	172.16.163.102	172.16.173.1	SMB2	Create Request File: file05
148	3.002441	172.16.163.102	172.16.173.1	SMB2	Create Request File: file05
156	3.002508	172.16.163.102	172.16.173.1	SMB2	Create Request File: file05
164	3.002577	172.16.163.102	172.16.173.1	SMB2	Create Request File: file05
172	3.002677	172.16.163.102	172.16.173.1	SMB2	Create Request File: file05
180	3.002778	172.16.163.102	172.16.173.1	SMB2	Create Request File: file05
188	3.002847	172.16.163.102	172.16.173.1	SMB2	Create Request File: file05
196	3.002923	172.16.163.102	172.16.173.1	SMB2	Create Request File: file05
204	3.002992	172.16.163.102	172.16.173.1	SMB2	Create Request File: file05
243	6.003711	172.16.163.103	172.16.173.1	SMB2	Create Request File: file07
282	10.004576	172.16.163.104	172.16.173.1	SMB2	Create Request File: file09
290	10.004652	172.16.163.104	172.16.173.1	SMB2	Create Request File: file09
298	10.004726	172.16.163.104	172.16.173.1	SMB2	Create Request File: file09
306	10.004793	172.16.163.104	172.16.173.1	SMB2	Create Request File: file09
314	10.004860	172.16.163.104	172.16.173.1	SMB2	Create Request File: file09
353	11.005343	172.16.163.105	172.16.173.1	SMB2	Create Request File: file11

USING GLOBAL PARAMETER FILE REFERENCES IN FORMULAS

Global User Parameter file references can be used in Functions in Action input fields. For example, the following SMB2 Scenario uses Global User Parameter file references in a formula in the User Name and Password fields of the **Session Setup** Action



when this Action is executed, the result is a User Name value = the contents of UP(4,A) + the string "001" and Password = the contents of UP(4,B) and the string "001". If column A of the Global User Parameter file at index 4 looked like

USER
user

the packet capture file for this Scenario would look like

Protocol	Info
SMB2	SessionSetup Request, NTLMSSP_NEGOTIATE
SMB2	SessionSetup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
SMB2	SessionSetup Request, NTLMSSP_AUTH, User: WORKGROUP\USER001
SMB2	SessionSetup Response
SMB2	SessionLogoff Request
SMB2	SessionLogoff Response
SMB2	NegotiateProtocol Request
SMB2	NegotiateProtocol Response
SMB2	SessionSetup Request, NTLMSSP_NEGOTIATE
SMB2	SessionSetup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
SMB2	SessionSetup Request, NTLMSSP_AUTH, User: WORKGROUP\user001
SMB2	SessionSetup Response
SMB2	SessionLogoff Request
SMB2	SessionLogoff Response
SMB2	NegotiateProtocol Request
SMB2	NegotiateProtocol Response
SMB2	SessionSetup Request, NTLMSSP_NEGOTIATE
SMB2	SessionSetup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
SMB2	SessionSetup Request, NTLMSSP_AUTH, User: WORKGROUP\USER001
SMB2	SessionSetup Response

Where the user names in the Session Setup would alternate between USER001 and user001 as seen above.

USER PARAMETER FILE ADVANCE and RESET

The **Advance User Parameters** and **Reset User Parameters** Actions are described in the more detail below. Both Actions provide the ability to control the access to the next reference to User Parameter Files.

Reset User Parameters - reset the pointer to a User Parameter file to a) the beginning of the file (using the Scope == Global setting) or reset the pointer to a User Parameter file to the entry that the current Scenario was using when it started. Reset ALL columns or a specified column.

Advance User Parameters - advance the pointer to a User Parameter file one or more entries down a specified column or for all columns with Scope == Scenario (only impacts the current Scenario) or Scope == Global (impacts all Scenarios).

The following rules apply to UP file columns within an executing Scenario.

Scenario per-column Pointers:

Global UP Pointer - the pointer used by all Scenarios to determine the first UP entry for each column when the Scenario begins execution

Scenario Current Pointer - the pointer that is used within the Scenario as Scope == Scenario Advances and Resets are executed

Scenario Base Pointer - the pointer to the element that was the current Scenario element when the Scenario started executing

At Time = 0, the Global Pointer to the UP file points at row 0 (if row 1 is the very first row of the UP file).

On entry to a Scenario the Global Pointer (wherever it happens to be pointing at that time) is incremented by 1 and that becomes the Scenario's Global UP Pointer. Scenario Current Pointer, Scenario Base Pointer are set equal to the Global UP Pointer at the time Scenarios begin execution.

The Global UP Pointer is incremented whenever an Advance UP Col x or All, Scope == Global is executed in any Scenario; the Scenario Current Pointer is set equal to the Global UP Pointer when this happens.

The Scenario UP Pointer is incremented whenever an Advance UP Col x or All, Scope == Scenario is executed in the current Scenario.

The Scenario UP Pointer is set equal to the Scenario Base Pointer whenever a Reset Col x, All Scope == Scenario is executed in the current Scenario.

An Example using Advance and Reset User Parameters:

For the following example use the following:

UP File (in UP Map index 1)

	A	B	C
1	A	!	
2	B	@	
3	C	#	
4	D	\$	
5	E	%	
6	F	^	
7	G	&	
8	H	*	
9	I	(
10	J)	

Scenario

The screenshot shows the 'Scenario' configuration dialog for a specific scenario named 'Client SMB2 neg.client_scenario'. The left pane lists the steps in the scenario, and the right pane shows the properties for the selected step (step 5, 'Create Or Open File').

#	Protocol	Name
1	SMB	Open SMB TCP Connection
2	SMB	Negotiate
3	SMB	Session Setup
4	SMB	Tree Connect
5	SMB	Create Or Open File
6	SMB	File Write
7	SMB	File Close
8	SWT	Advance User Parameters
9	SWT	Advance User Parameters
10	SMB	Create Or Open File
11	SMB	File Write
12	SMB	File Close
13	SWT	Advance User Parameters
14	SWT	Advance User Parameters
15	SWT	Reset User Parameters
16	SMB	Create Or Open File
17	SMB	File Write
18	SMB	File Close
19	SWT	Advance User Parameters
20	SWT	Advance User Parameters
21	SWT	Reset User Parameters
22	SMB	Create Or Open File
23	SMB	File Write
24	SMB	File Close
25	SMB	Tree Disconnect
26	SMB	Session Logoff

Properties for Step 5: Create Or Open File

Name	Value
Input	
Alias	
Share Handle	Default
Path Name	
Input Format	String
Path	=@UP(1, A) + @STRING(.) + @UP(1, B) + @STRING(.) + @UP(1, C) <input type="button" value="fx"/>
Header Flags	
Unicode	Enabled
Open Flag	0x00000000 (0)
ACL Mode	0x001F01BF (2032063) / Full Control
Allocation Size	0
File Attributes	0x00000080 (128)
Share Mode	0x00000007 (7)
Create Disposition	Open if exists, Create otherwise
Create Options	0x00000000 (0)
Impersonation Level	Impersonation
Security Flags	0x00 (0)
OpLock Break Response	None
Output	
Output Handle	5: SMBFileHandle
Response Handlers	
Completion Status	
Scenario Impact	

The Scenario is relatively simple, create files with a name crafted from columns A, B and C from the

UP file in the UP Map index 1.

Following each **Create or Open File** Action, there are some UP-related Actions to change the pointers to the columns in UP1

After the first **Create or Open File** - Advance columns A and B one row each, Scope == Global

After the second **Create or Open File** - Advance columns B and C one row each, Scope == Global, and Reset column A, Scope==Scenario

After the third **Create or Open File** - Advance columns A and C one row each, Scope == Global, and Reset column B, Scope==Global

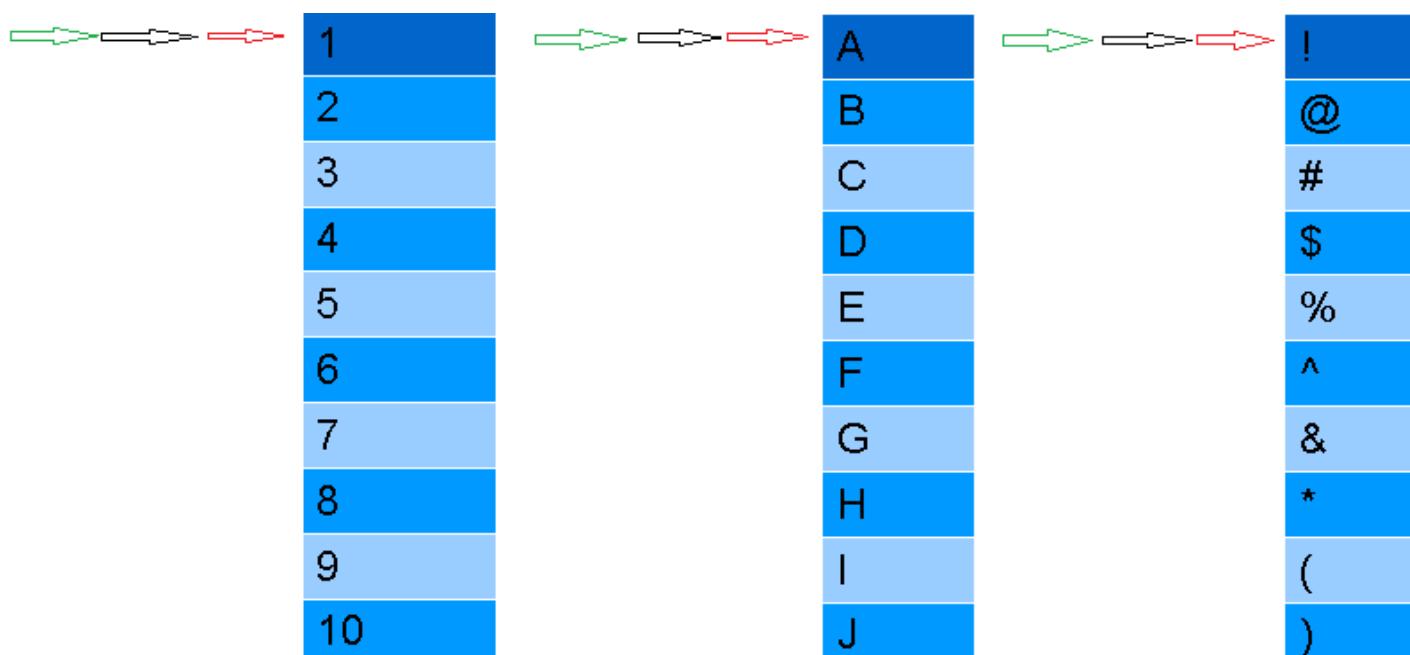
In the following diagrams, there will be Green, Black and Red arrows shown. These arrows indicate:

Red: Global UP Pointer

Black: Scenario Current Pointer

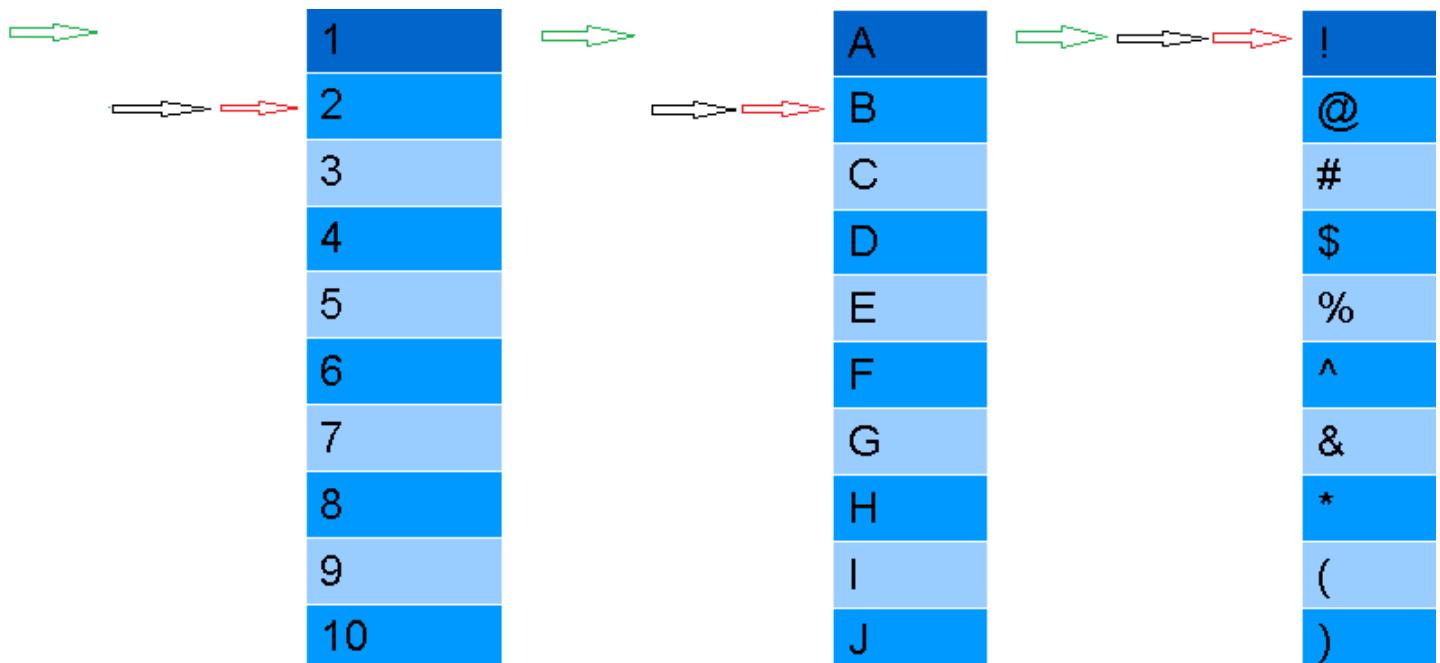
Green: Scenario Base Pointer

On entry to the first instance of the Scenario, the UP file pointers look like this:



This generates the input to the **Create or Open File** Action in line 5 of **1.A.!**

After the first set of Advances : Advance column A and B (lines 8 and 9), the pointers to the UP file entries look like:

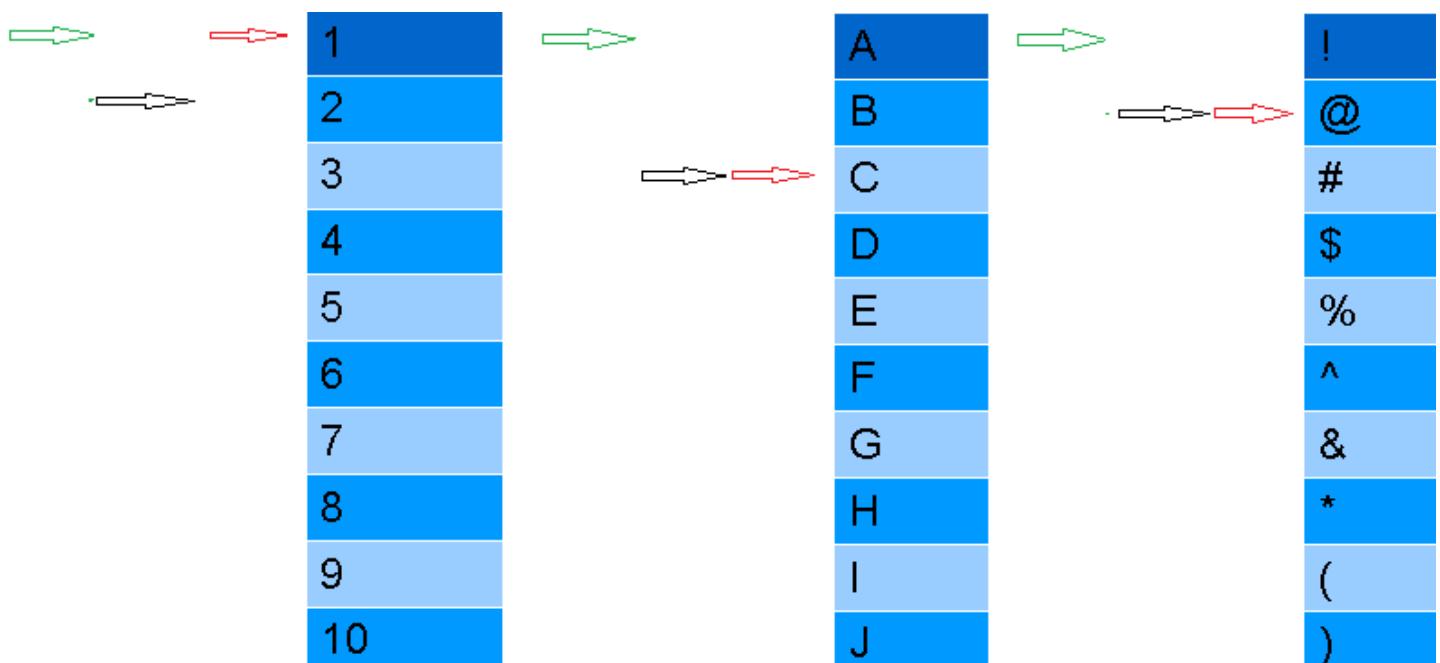


Notice the following:

- Both the Black and Red pointers advance whereas the Green (pointer on Scenario start) stays the same.
- Column C pointers are unmoved since no Advance or Reset operations were performed on column C.

This generates the input to the **Create or Open File** Action in line 10 of **2.B.!**

After the second set of Advances and Reset (lines 13,14,15), the UP file pointers look like:

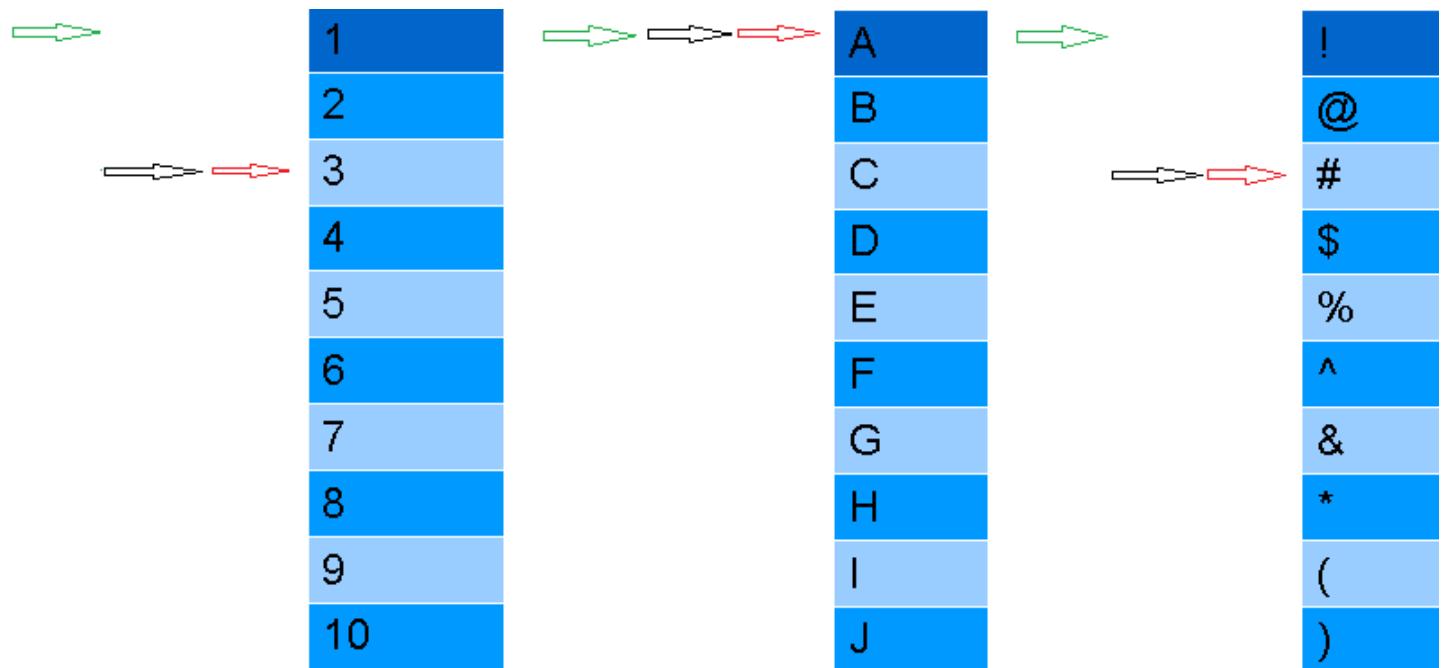


Notice the following:

- Both the Red pointer for column A is set back to match the Green pointer (Reset Column A, Scope == Scenario).
- Col B and C pointers are both Advanced, Scope==Global one row so the Black and Red pointers remain synchronized.

This generates the input to the **Create or Open File** Action in line 16 of **1.C.@**

Finally, after the third set of Advances and Reset (lines 19,20,21), the pointers to the UP file look like:



Notice the following:

- Both the Black and **Red** pointers for column A synchronize again due to the Advance Column A, Scope == Global.
- Both the Black and **Red** pointers for column C are moved ahead (Advance column C, Scope==Global).
- Column B **Green**, Black and **Red** pointers are reset to the first entry due to the Reset Column B Scope==Global.

This generates the input to the Create or Open File Action in line 22 of **3.A.#**.

When visualized in PCAP file, looking only at the Create or Open File packets:

15 3.001458 172.16.240.1	172.16.244.1	SMB	154 NT Create AndX Request, FID: 0x0001, Path: 1.A.!
941 3.017377 172.16.240.1	172.16.244.1	SMB	154 NT Create AndX Request, FID: 0x0002, Path: 2.B.!
1867 3.034720 172.16.240.1	172.16.244.1	SMB	154 NT Create AndX Request, FID: 0x0003, Path: 1.C.%
2793 3.051972 172.16.240.1	172.16.244.1	SMB	154 NT Create AndX Request, FID: 0x0004, Path: 3.A.#

If the Scenario described above were executed in a Project with a Client Load Specification of 1 Concurrent Scenario, the Second Scenario would generate the following PCAP entries for the Create or Open File packets:

995 3.018906 172.16.240.2	172.16.244.1	SMB	154 NT Create AndX Request, FID: 0x0001, Path: 4.B.\$
1008 3.019116 172.16.240.2	172.16.244.1	SMB	154 NT Create AndX Request, FID: 0x0002, Path: 5.C.%
1021 3.019340 172.16.240.2	172.16.244.1	SMB	154 NT Create AndX Request, FID: 0x0003, Path: 4.D.*
1947 3.037206 172.16.240.2	172.16.244.1	SMB	154 NT Create AndX Request, FID: 0x0004, Path: 6.A.^

And the third instance of the Scenario would generate the following PCAP entries for the Create or Open File packets:

1975 3.037682 172.16.240.3	172.16.244.1	SMB	154 NT Create AndX Request, FID: 0x0001, Path: 7.B.&
1988 3.037898 172.16.240.3	172.16.244.1	SMB	154 NT Create AndX Request, FID: 0x0002, Path: 8.C.&
2001 3.038125 172.16.240.3	172.16.244.1	SMB	154 NT Create AndX Request, FID: 0x0003, Path: 7.D.*
2927 3.055383 172.16.240.3	172.16.244.1	SMB	154 NT Create AndX Request, FID: 0x0004, Path: 9.A.(

If the Scenarios were executed with a Client Load Specification of other than 1 Concurrent Scenario (N Concurrent or X Scenarios per Second, etc) then the results would be very different and somewhat

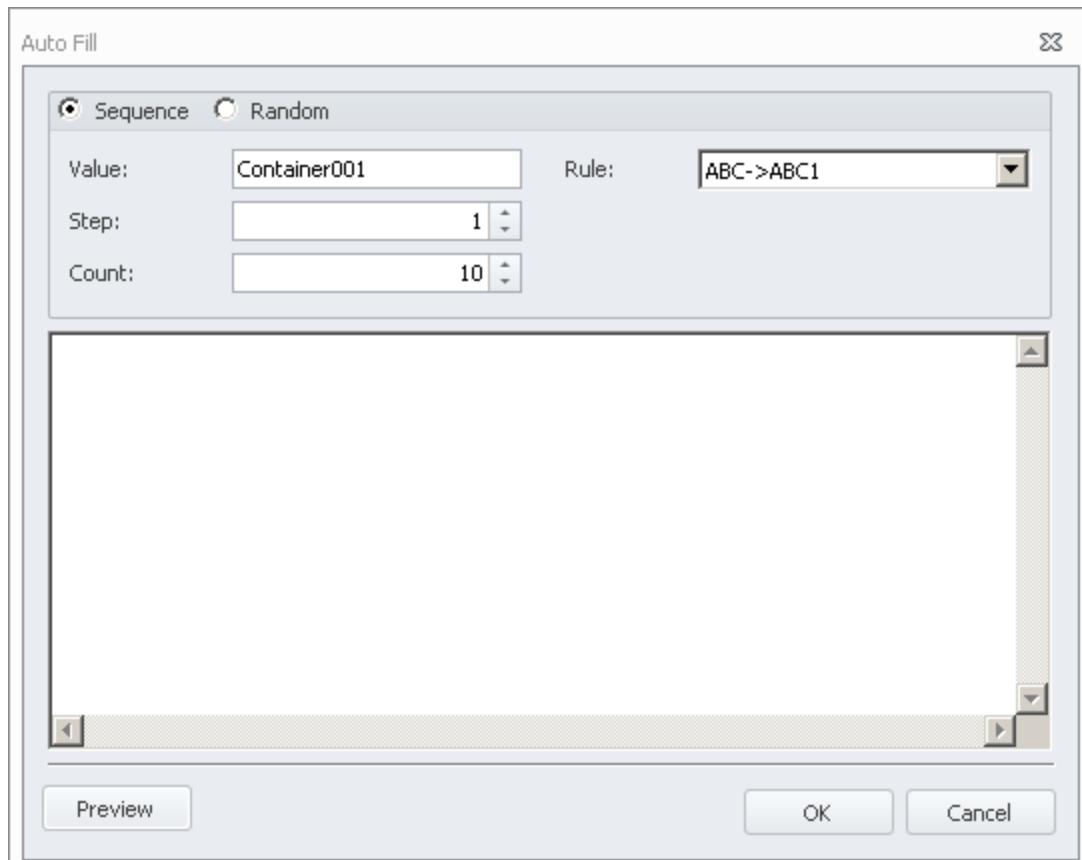
unpredictable due to the Global Advances and Resets. If all of the Advances and Resets were of Scope == Scenario then the predictability of the results is much higher.

AUTOFILL

Tabular data such as User Parameter files support the AutoFill dialog.

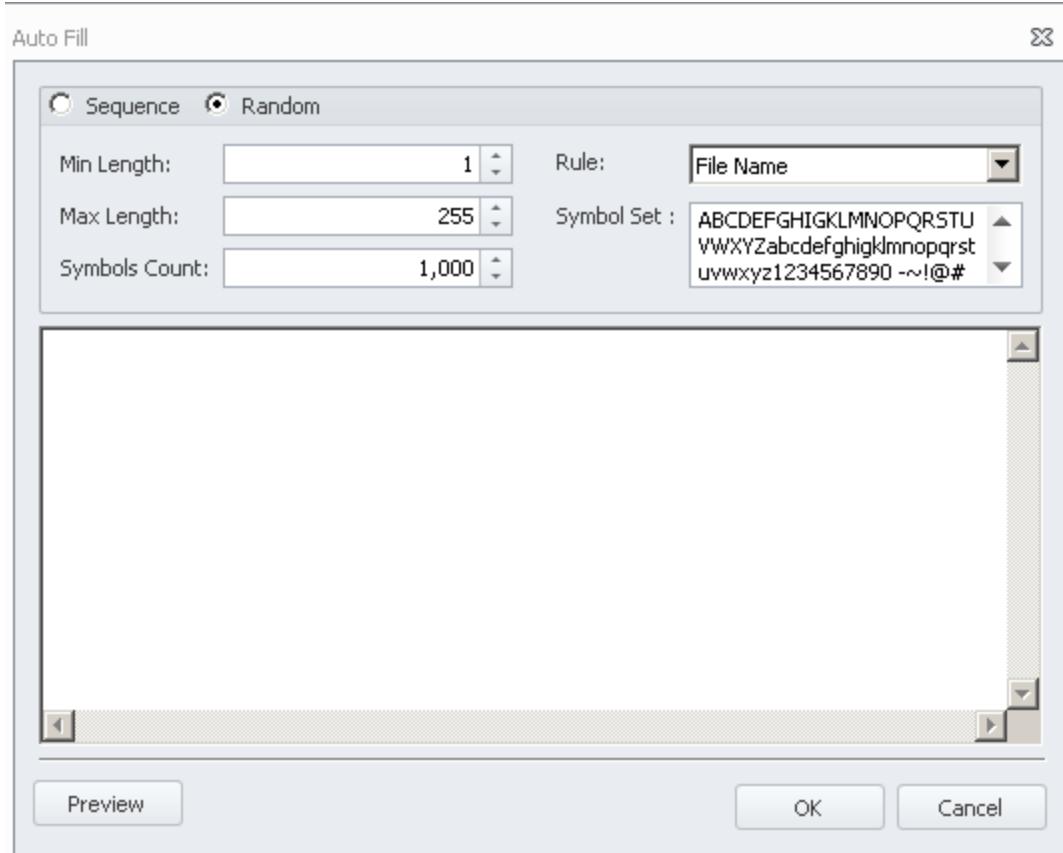
Dialogs that support the AutoFill capability display the Auto Fill button  . Clicking the button causes the AutoFill dialog to display. Based on the information entered, the application automatically generates a list of items. If Sequence is chosen, the items will be generated in Sequential order based on Value, Rule, Count and Step. If Random is chosen, then the items will be generated in Random order based on Rule, Symbols Set, Min Length, Max Length and Symbols Count.

Sequence:



Value	Base item value, which can be a string or IP address depending on the Data Type selection
Rule	<ul style="list-style-type: none"> Based on the Data Type selection, determines how the list is generated: String – use the ABC->ABC1, ABC1 -> ABC2, AB1CD->AB2CD rules to create string variables with incrementing numeric values IP Address – use the 1.1.1.1->2.2.2.2 rule to sequentially generate IP addresses by incrementing the last digit of the IP address Numbers - use 1->2 creating increasing numeric values <p>NOTE: applying a rule to a different value will cause the value to repeat as is (e.g. apply the 1->2 rule to a string will cause the string to repeat Count times)</p>
Step	Based on the Value and Rule, the difference between Sequential Values:

Count	Number of Sequential Values to generate
Preview	See what the output of the Rule applied to Value, Step and Count will be

Random:

Rule	File Name: Text pattern using ASCII character set Advanced File Name: Text pattern with broader character set Text: Text pattern with a subset of the ASCII character set Number: Random numbers
Symbols Set	The character set that will be used to generate the random patterns
Min Length	Minimum length of the patterns to generate
Max Length	Maximum length of the patterns to generate
Symbols Count	The Number of patterns to generate
Preview	See what the output of the Rule, Symbols, Min, Max and Count will be

USER PARAMETER FILE COLUMN ALIASES

By default, all User Parameter file columns are referred to by capital letters starting with A and increasing through the alphabet for every column in the User Parameter file. In Projects with a small number of User Parameter files or a limited number of different input values, it may be easy for the

Tester to remember the contents of Column A in a Local User Parameter file or column C in the Global User Parameter file at User Parameter File Map index 3 but as the number of columns and User Parameter files increases in a Project, remembering all of the associations may become problematic.

User Parameter File Aliases helps solve that problem by allowing the Tester to assign meaningful text to every User Parameter file column in use and refer to that column by that meaningful text. In the example below is a User Parameter file named Global File names. The column in use have been given meaningful names so that the Tester can use them in the appropriate input fields.

A:File_Names	B	C:UserName	D:password	E	F:Loop_Co...	G	H:DeLay
file01		user1	pass1		1		1000
file02		user2	pass2		5		2000
file03					10		3000
file04							4000
file05							
file06							
file07							

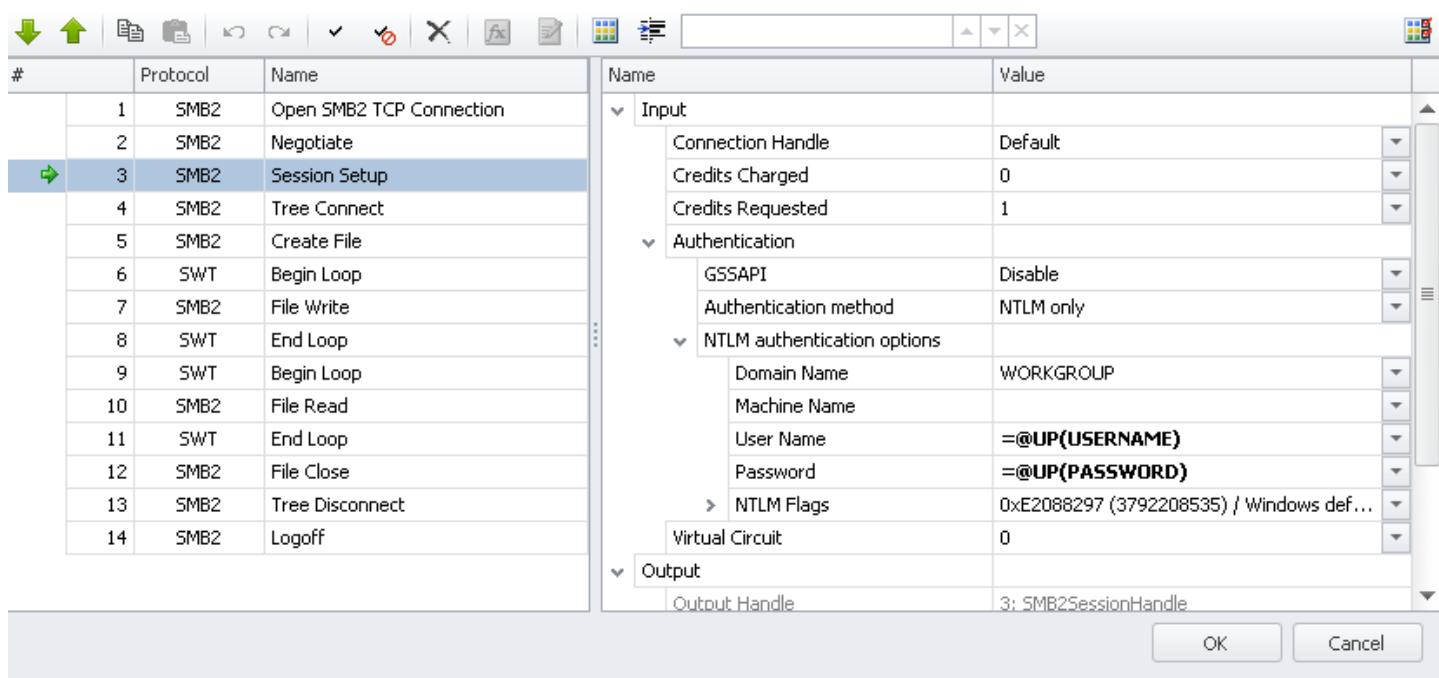
Aliases must obey the following rules

1. Aliases must be at least 3 characters long (to avoid overlap with default column names)
2. Aliases may contain the letters a...z, A...Z, 1-0, and underscore "_".
3. Aliases may not contain any special characters (!,@,#,\$,...) except underscore.
4. Aliases must be unique in a UP file.
5. Aliases are not case sensitive (e.g. DeLay is the same as delay is the same as DELAy, etc)

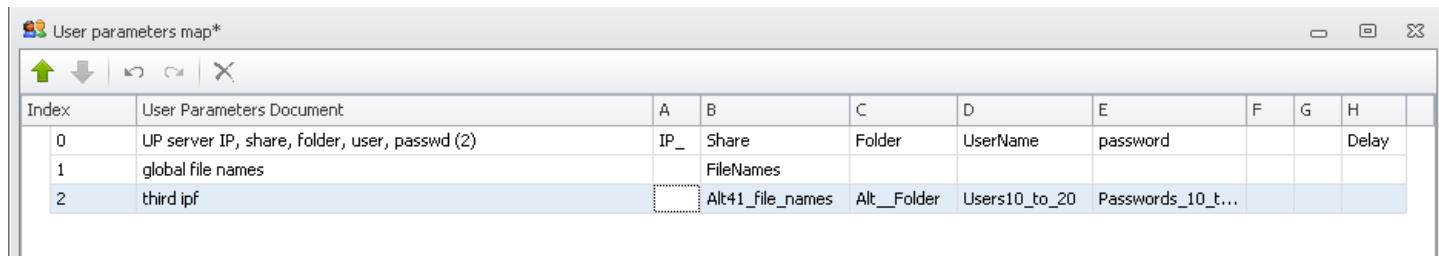
If the User Parameter file shown above was at index 1 of the User Parameter File Map of a Project, the various columns of this file could be referred to as:

`@UP(1,H)`
`@UP(1, Delay)`
`@UP(delay)` if there are no other UP files in the UP File Map with a column Alias named "Delay".

The example Project belows shows the SMB2 Session Setup command using Alias references for User Name and Password input fields.



When a UP file with Aliases is dropped into the UP File Map, the Column Aliases are shown in the UP Map.



Double clicking on a row in the UP Map will open the UP File named in that row for editing.

Importing CSV Files with Aliases

When a CSV File with Aliases is imported into a Project using the button, the import process will attempt to fix any Aliases that do not conform to these rules

- Aliases must be at least 3 characters long (to avoid overlap with default column names)
- Aliases may contain the letters a...z, A...Z, 1-0, and underscore "_".
- Aliases may not contain any special characters (!,@,#,\$,...) except underscore.
- Aliases must be unique in a UP file.
- Aliases are not case sensitive (e.g. DeDelay is the same as delay is the same as DELAY, etc)

For example,

- An Alias that is less than 3 characters long will be changed to be at least 3 characters long by appending "_" and some number to the Alias ("ST" will be converted to "ST_1").
- An Alias that contains spaces will have the spaces converted to "_" ("S T" will be converted to "S_T").
- An Alias that contains illegal characters (ex: # , \$ * &) will have those illegal characters converted to "_" ("S%^&*T" will be converted to S_____T").

Advance User Parameters and Reset User Parameters Actions and Aliases

The Advance User Parameters and Reset User Parameters Actions support Aliases. In the

screenshot below of an **Advance User Parameters** Action, UP5 is selected as the UP file index to Advance and the Alias "135" is the column selected to Advance.

Name	Value
Input	
User Parameters ID	UP5
Column ID	135
Scope	STU
Step	642 dummy5 dummy6 XYZ 246 A

The UP File at index 5's Aliases can be seen by opening the UP file or by opening the UP Map.

Index	User Parameters Document	A	B	C	D	E	F	G	H	I	J	K	L
0	A thru I and 1 thru 9	ABC	123	DEF	456	GHI	789						
1	P thru Z and 1 thru 6	PQR	135	dummy1	dummy2	dummy3	dummy4	STU	642	dummy5	dummy6	XYZ	246
2	P thru Z and 1 thru 6 port level	PQR	135	dummy1	dummy2	dummy3	dummy4	STU	642	dummy5	dummy6	XYZ	246
3	P thru Z and 1 thru 6 ser 176	PQR	135_BAD	dummy1	dummy2	dummy3	empty4	STU	456	empty5	empty6	XYZ	789
4	A thru I and 1 thru 9 (2)	ABC	123	DEF	456	GHI	789						
5	P thru Z and 1 thru 6 port level (2)	PQR	135	dummy1	dummy2	dummy3	dummy4	STU	642	dummy5	dummy6	XYZ	246

The Reset User Parameter File Action behaves the same as the Advance User Parameter File Action.

Name	Value
Input	
User Parameters ID	UP5
Column ID	135
Scope	STU
Step	642 dummy5 dummy6 XYZ 246 A

One of the possible selections in the User Parameters ID field is ANY. This selection forces the TDE to display the complete list of UNIQUE Aliases that are in the UP Map. For example, if the UP Map contained these files

Index	User Parameters Document	A	B	C	D	E	F	G	H
0	A thru I and 1 thru 9	ABC	123	DEF	456	GHI	789		
1	P thru Z and 1 thru 6	PQR	135	dummy1	dummy2	dummy3	dummy4	STU	642
2	P thru Z and 1 thru 6 port level	PQR	135	dummy1	dummy2	dummy3	dummy4	STU	642
3	P thru Z and 1 thru 6 ser 176	PQR	135_BAD	dummy1	dummy2	dummy3	empty4	STU	456
4	A thru I and 1 thru 9 (2)	ABC	123	DEF	456	GHI	789		
5	P thru Z and 1 thru 6 port level (2)	PQR	135	dummy1	dummy2	dummy3	dummy4	STU	642
6	Unique Aliases	PQR_123	135_BAD_aliases	dummy1234	dummy2234	dummy3234	empty4_4_5_6_...	STU_Unique	456_UNI

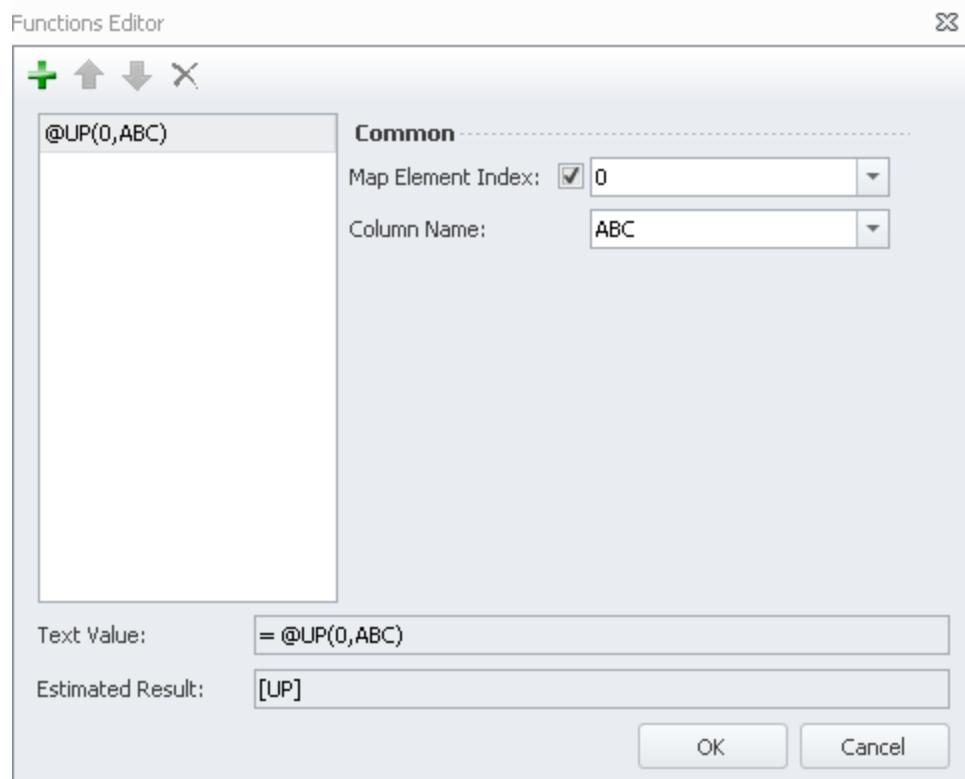
and the ANY selection is made in a Reset User Parameters Action, the TDE will display the UNIQUE Aliases from the UP Map. See below the part of the drop down menu displaying the unique Aliases

from the UP file at index 6.

Name	Value
Input	
User Parameters ID	Any
Column ID	135
Scope	dummy1234 dummy2234 dummy3234 empty4_4_5_6_7_890 STU_Unique 456_UNIQUE empty5_6_7_8_9_0_1_2

Function Editor and Aliases

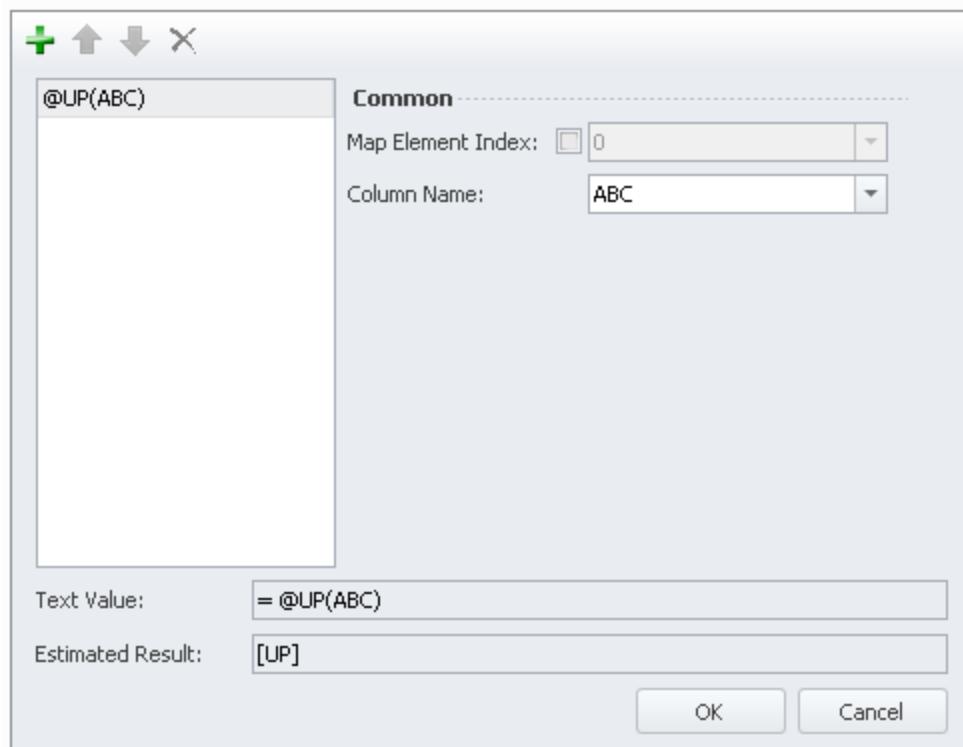
When adding a User Parameter File reference to a Function using the Function Editor, the Tester can choose to use the (index, alias) form of reference or the (alias) form by checking or un-checking the Map Element Index check box. The following screenshot shows the with index form which produces the @UP(0,ABC) reference.



When the Map Element Index box is unchecked, the @UP(Alias) reference is generated

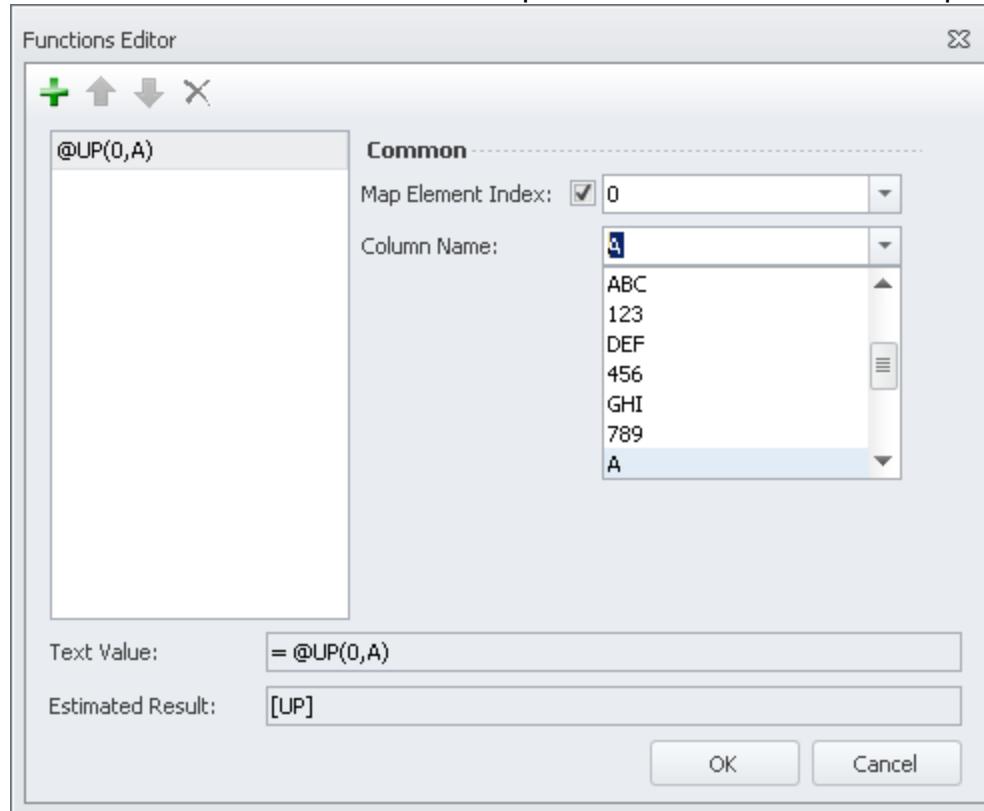
Functions Editor

X



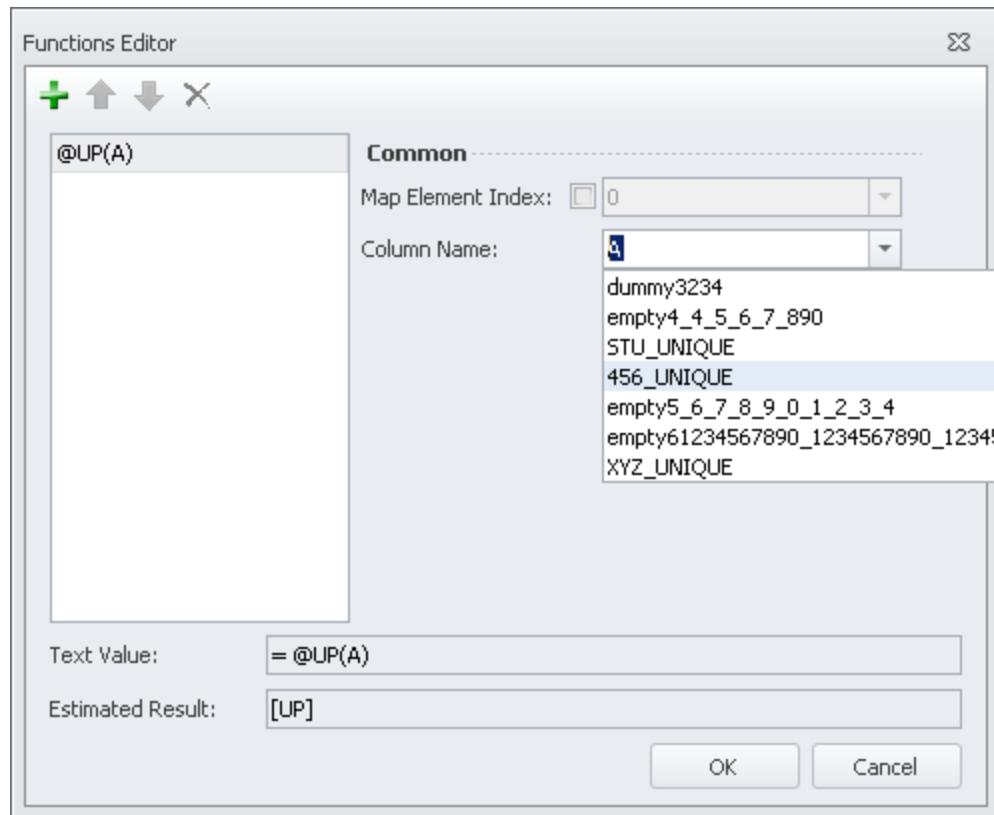
These two references are identical as long as there is no more than one column Alias named "ABC" in any UP File in the UP Map. If there were more than one Alias named "ABC", the `@UP(0,ABC)` form would be required.

With the Map Element Index enabled (checked), the Column Name drop down menu will present the Tester with the list of Aliases that are present in the UP file with the specified index value.

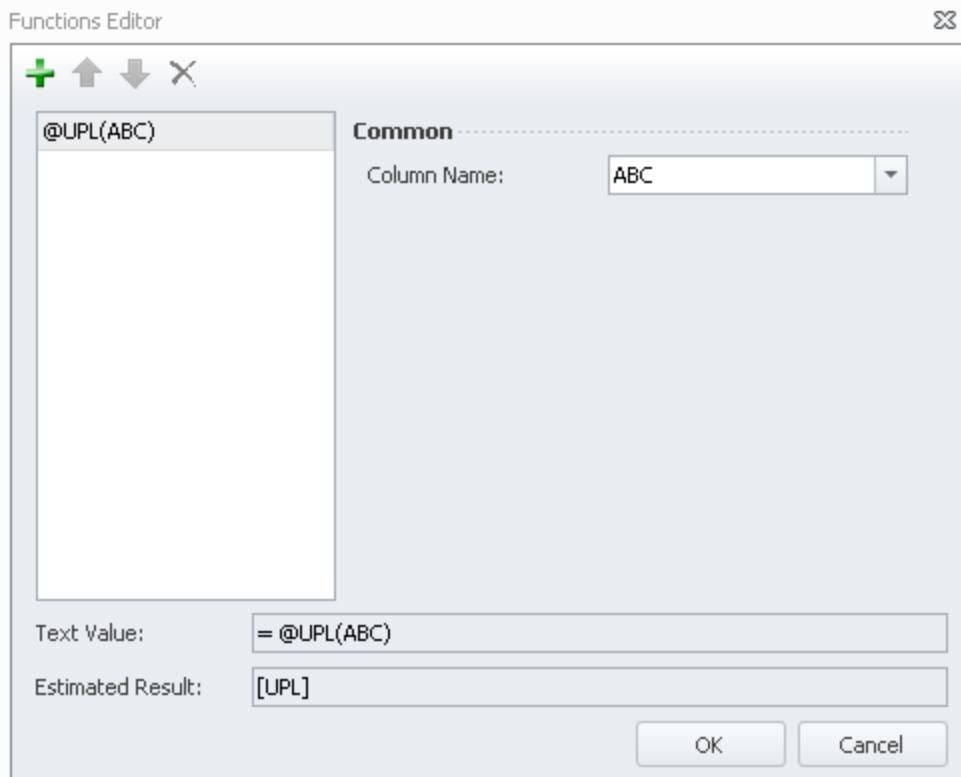


If the Map Element Index is disabled, the drop down menu will present the Tester with a list of all of the

UNIQUE Aliases in the User Parameter Map. All non-unique Aliases will not be shown in the drop down menu.

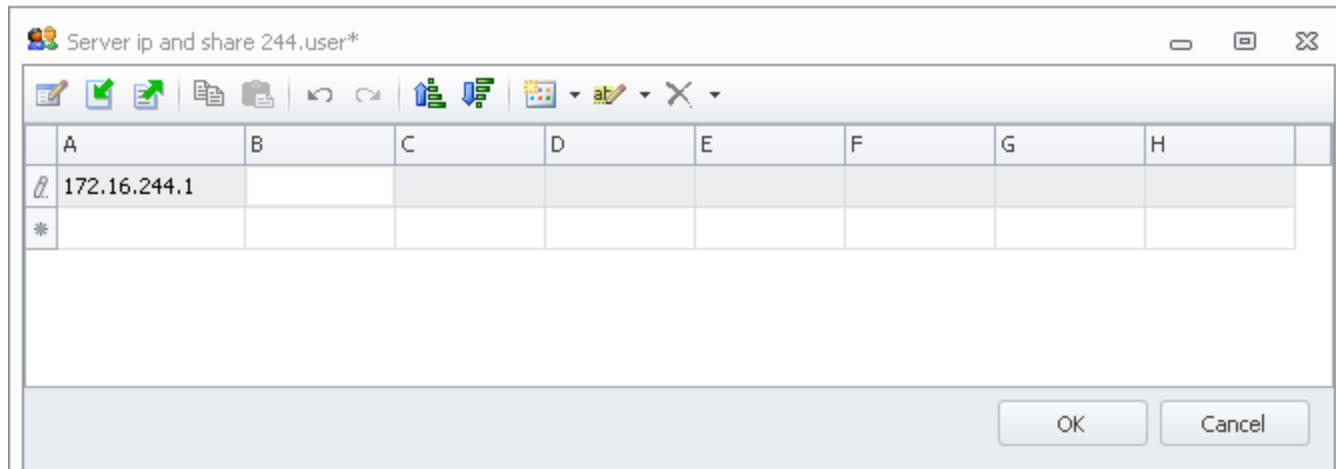


Inserting a Local User Parameter reference using the Function Editor would look like this. This reference would get the local UP file contents from a column with the Alias of "ABC". If not required to be used in a Function then the reference \$(ABC) is equivalent, or, assuming that this column and column A are the same, \$(A).



USER PARAMETER FILE EDITOR TOOLBAR

The User Parameter File Editor provides several editing tools for UPFs in its toolbar.



Several of these tools were not mentioned in the tutorial above:

Autofill allows creation of content in UPF columns using a selection of rules that create content based on the content of the input string

Import allows creating User Parameter files by copying the contents of a .CSV file

Export allows exporting the contents of a UPF into a .CSV file for editing outside of the TDE using a spreadsheet application

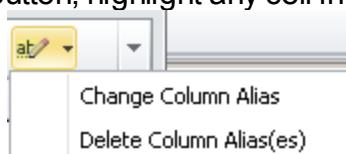
Copy and Paste allow creation of content in a UPF using typical Windows Copy and Paste operations (Ctl-C and Ctl-V also perform the same operations as these buttons do)

Undo and Redo allow Undo-ing and Redo-ing of changes made to UPF columns

Sort Ascending or Descending to sort the contents of a User Parameter file based on the contents of a selected column

Add Row/Column and Clear Delete Row/Column allows the user to Add Rows or Add Columns (default operation is Add Row) or Clear the contents of a cell or selected cells or Delete whole Rows or Columns (default operation is Clear)

Change/Delete Column Alias allows the user to Delete or Change a Column Alias. To use the Change/Delete Column Alias button, highlight any cell in the column to be changed and click down arrow



symbol on the right edge of the . Select Rename or A dialog box will open that

allows the user to select to either Change the existing Alias (including adding one for the first time) or Deleting an existing Column Alias. The Change Column Alias function can also be executed by double-clicking any column header field.

USER PARAMETER FILE ENTRIES and START SERVER ACTIONS

Load DynamiX Start Server Actions accept all eight forms of User Parameter File references for input to Start Server fields.

Global	@UP(<Map Index>, <Col Index>), @UP(<Map Index>,<Alias>), @UP(<Alias>)
Local	\$(<Col Index>), \$(<Alias>)
Function-Compatible Local	@UPL(<Col Index>), @UPL(<Alias>)

Since a Server Scenario is only ever started once, the Start Server Actions always and only reference the top (first data item) row of the specified UP file. For example in the SMB Start Server Actions below, all forms of UP access are used as input to the SMB Start Server Action. With the exception of the inputs for List of User Names and List of Passwords, only the first data item in each column of the specified Local or Global User Parameter file will be used. The User Parameter file columns referenced by the List of User Names and List of Passwords inputs will use all of the User Parameter File entries (rows) in the columns with Alias User_ID and PassWord in the UP File at index 3 in the User Parameter File map.

#	Protocol	Name	Name	Value
1	SMB	Start SMB Server	Input	
			IPv4 Address	= @UP(0,A)
			IPv6 Address	= @UP(1, IPV6_Address)
			Port	= @UP(SMB_Port)
			Keep Alive	True
			NetBIOS Enabled	False
			Domain Name	= @UPL(A)
			Machine Name	\$(Machine_Name)
			OS Name	= @UPL(OS_Name)
			LAN Manager Name	\$(C)
			Guest Account	Disabled
			Packet Signing	Disabled
			Plain Text Password	Disabled
			List of User Names	= @UP(3,User_ID)
			List of Passwords	= @UP(3, PassWord)

Advanced Concepts: Data File Systems & Data Verification

Advanced Concepts: Data File Systems and Data Verification

Using Data File Systems and Data Verification you can create file systems content and structure on Load DynamiX servers; specify the data content to be transmitted by all Object (CDMI, Open Stack Swift), HTTP/HTTPS, SCSI, CIFS-SMB and NFS Write I/O operations and verify the data content received by Object (CDMI, Open Stack Swift), HTTP/HTTPS, SCSI, CIFS-SMB and NFS Read I/O operations.

Load DynamiX DataContent supports to following file content types:

::DataContent Type	Content
::Sequential	64 bit integers, increasing value pattern
::Random	Random pattern, not repeatable, not reusable
::SeededRandom(Seed)	Random pattern, repeatable (same Seed value always produces same Random pattern) If Seed = 0, the Seed value is randomly selected for each Scenario instance
:Compressible(Seed,Percentage)	Mix of random pattern and zeros, where Seed creates the random pattern and Percentage defines the size of the non-compressible portion of the pattern
Physical	A copy or link to a real file somewhere on a filesystem accessible by the Load DynamiX workstation Physical files are always referred to by a ::DataContent(X)

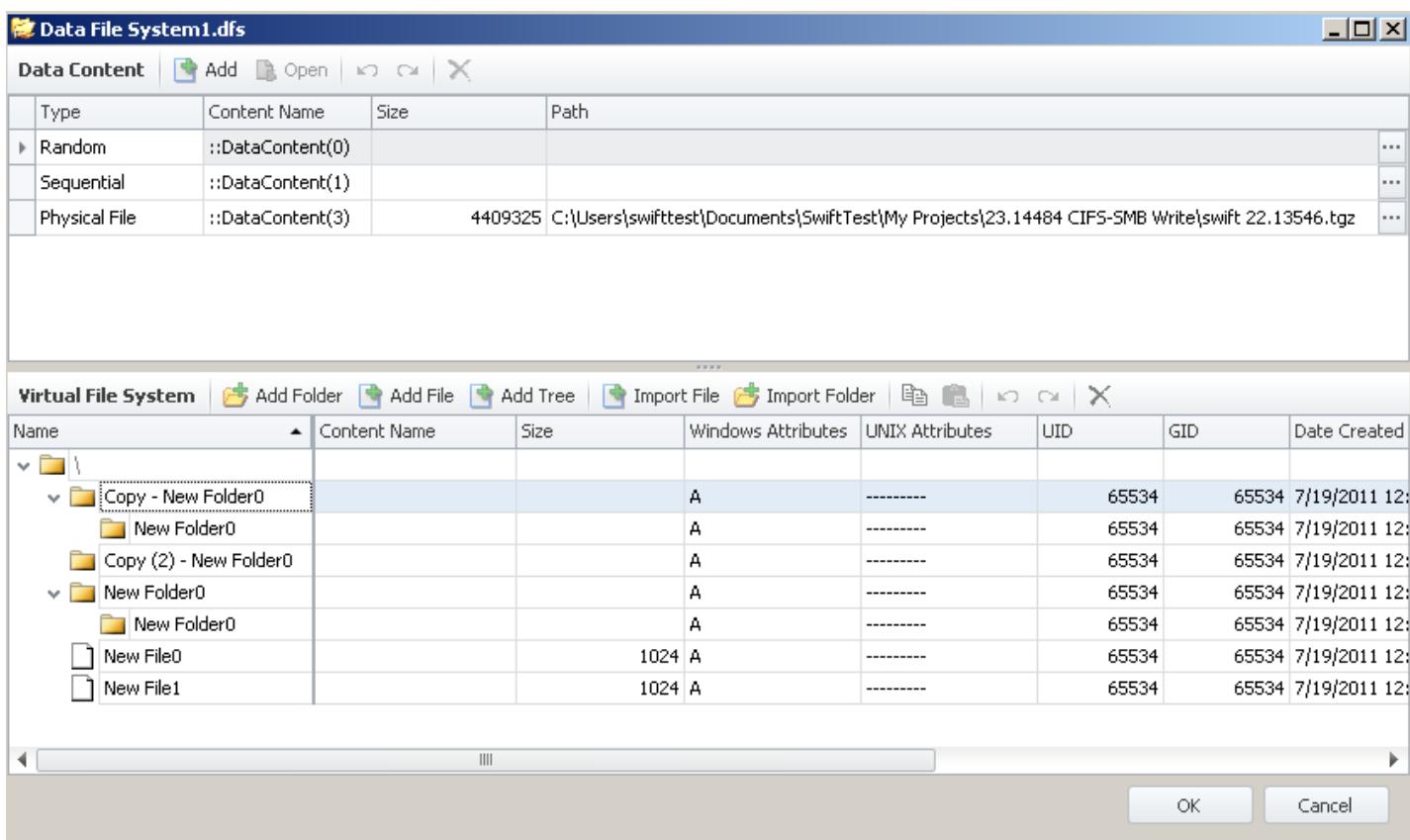
Data File System Configuration

The Data File System allows the Tester to define custom Data Content to be utilized by Load DynamiX HTTP Storage (CDMI, OpenStack Swift/Cinder, Amazon S3), HTTP/HTTPS, SCSI, iSCSI, CIFS-SMB and NFS Actions in context of Read/Write File, URL and Block I/O operations and to create whole file system structures and content that can be instantiated on Load DynamiX Servers.

A Data File System is created via Add New Item interface in the Project Explorer. Click the Add New Item interface and select Data File System. The Data File System Editor is opened. All Data File Systems are created in the context of the current Project but can be moved to the Resource Library and, from there, to any Project, thus making the same Data File System available to multiple Projects. Also, multiple Data File Systems can be used in a single Project.

Once created, to activate a Data File System, the Tester must associate the Data File System with one of the Logical Ports, Network Profiles or Scenarios in the Timeline. This is achieved by dragging the Data File System item from the Project Explorer window and dropping it onto the appropriate location in the Timeline window.

A Data File System dragged onto a Client Scenario provides the client Actions access to the ::DataContent(X) files for Read and Write use. A Data File System dragged onto a Server Scenario causes the file system structure and contents defined in the Data File System to be instantiated on a Load DynamiX Server for use by the Client Scenario Actions.



The Data File System Editor is opened immediately after selection of Data File System from the Add New Item interface or double clicking the Data File System item in the Project Explorer, Timeline or Resource Library. The Tester must configure the Data File System to meet the needs of the test Project.

The upper pane is used to create "::DataContent" files. All Data File System Resources contain ::DataContent files of type Sequential and Random by default. The Add button allows the Tester to import a user-defined ("Physical") file or add Seeded Random files. Physical files come from a filesystem accessible to the Load DynamiX workstation. Sequential files are always the same, a sequence of 64 bit integers. Random files are a randomly ordered set of 64 bit values unique in every reference. A Seeded Random file is a Random file that generates same Random data pattern every time that Seed is used. A Seeded Random file with Seed == 0 produces a Random file that has a randomly selected Seed value for each new instance of a Scenario.

File structure (which can be downloaded into Internal Load DynamiX servers) is created by using the lower portion of the Data File System Editor window. Whole trees, folders and files may be added to match any desired file tree for testing purposes. When a Data File System is dragged onto a Load DynamiX Server Logical Port, Network Profile or Server Scenario, the file structure defined in the lower portion is instantiated on the Server when the Project is executed. So, if the Data File System shown above is dropped onto an SMB Server scenario, the files shown under the root folder will be created in the defined structure, with the defined attributes (size, dates, etc) and contents. As defined, this structure would be present at the top of the Share hierarchy of the SMB Server. Values of the properties of files or folders can be changed by clicking in the desired row/column entry and making the necessary changes.

After the ::DataContent files and/or file structure are created, close the Data File System dialog box by clicking the X in the upper right hand corner or the OK button and this will save the contents to the Data File System for this Project. To be used, the Data File System must be dragged onto a

Project Timeline. Once on the Timeline, the Data File System may be referenced in Write and Read operations.

If the Project includes a Load DynamiX Server and requires a fully defined file system on that server, the lower pane of the Data File System Editor is used to create the structure of the Data File System. Creating the structure can be done in one of two ways: by importing a structure and contents from a filesystem accessible from the Load DynamiX workstation where the TDE is installed (using Import Folder or Import File) or by hand crafting the structure (Add Folder, Add File or Add Files). The structure of a Data File System is controlled by the lower pane. The contents of a Data File System is controlled by associating file contents from the upper pane with filesystem entries in the lower pane. The content files (upper pan) may be mapped into the filesystem entries that is created in the lower pane by highlighting a file within the created structure, clicking the Content Name column entry and selecting from the choices provided there.

After the structure and contents are created, the data contents, size, permissions, or time stamps of the files or folders in the file system can be fine-tuned by clicking on the column data that is to be updated. Importing Files or Folders into a file system structure also causes a parallel entry for the data contained in the files to be created in the Data Content window (the top portion of the Data File System Editor). Data Content files (::DataContent(X)) can be used to specify file content that can be written to servers (Load DynamiX or external) and then potentially verified during a Read operation.

Data File System objects must be associated with (dragged and dropped onto) Project Resources such as a Network Profile, Scenario or Logical Port to be used at runtime. Each physical port on the Load DynamiX Appliance has approximately 2GB of memory in which to store DFS. Creating a DFS with more than 2GB of files will cause problems at runtime.

Writing Custom Data Content

Load DynamiX CIFS-SMB, SMB2, SCSI, NFSv3, NFSv4/4.1 and HTTP write-related (**File Write**, **Write File**, **Put**, **LUN Write**, etc) Actions include a "Data Content" property group for these Actions. This group includes a property "Data Source" that allows the Tester to specify the data (e.g. Data Content) to be written by all instances of this Action. The value of the "Data Source" property is specified in the "::DataContent(n)" format or by specifying a type of ::DataContent file such as ::Sequential(), SeededRandom(0), ::Random(), ::Compressible(Seed,Percentage) or SeededRandom(Seed). If specified in the ::DataContent(n) form, the value for "Data Source" property must match one of the elements in the Data File System that is associated with (dragged and dropped onto) this Project (Logical Port, Network Profile or Scenario). References of the form ::SeededRandom(Seed), ::Sequential, ::Random or ::Compressible(Seed, Percentage) do not require that a Data File System Resource be present on the Project Timeline.

The screenshot shows a software interface for managing a sequence of operations. On the left, a table lists 18 operations:

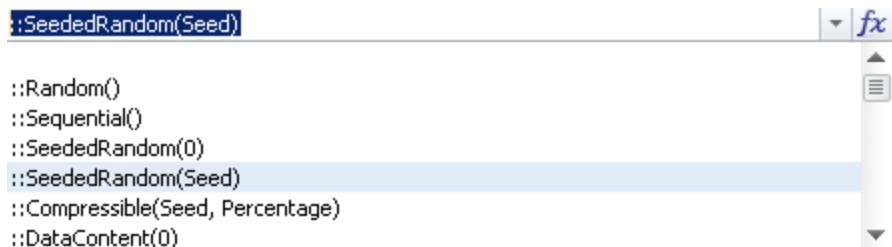
#	Protocol	Name
1	SWT	Create Variable
2	SWT	Create Variable
3	SWT	Create Variable
4	SMB2	Open SMB2 TCP Connection
5	SMB2	Negotiate
6	SMB2	Session Setup
7	SMB2	Tree Connect
8	SWT	Wait for Event
9	SMB2	Create File
10	SMB2	File Write
11	SMB2	File Close
12	SMB2	Create File
13	SMB2	File Read
14	SMB2	File Close
15	SWT	Raise Event
16	SMB2	Tree Disconnect
17	SMB2	Logoff
18	SMB2	Close SMB2 TCP Connection

On the right, the configuration for operation 10 (File Write) is shown in a detailed view:

Name	Value
Input	
File Handle	Default
Credits Charged	0
Credits Requested	1
Automatic Offset	False
File Offset	0
Bytes Per Block	1,024
Remaining Bytes to Write	0
Bytes Total	4KB
Block Sequence	Forward
Data Content	
Data Source	::DataContent(7)
Data Source Offset	=@UP(11,B)
Output	
Status Code	
Response Handlers	
Completion Status	
Scenario Impact	

At the bottom right are 'OK' and 'Cancel' buttons.

If ::SeededRandom(Seed) is selected in the Data Source field then an additional input field named Seed is provided for the Tester to provide a Seed value for the ::SeededRandom file. SeededRandom DataContent type produces pseudo random data streams with the period of 2 to the power 77 bytes for each of the given seed, which in turn is a 64 bit integer. In other words, approximately eighteen quintillion unique streams, each stream is 128 Zebibytes in length.



The Seed input field accepts constants, @Variable, @UP() or any Function like @RANDOM() that produces an integer as output, as input.

A configuration dialog is shown with the following settings:

Data Content	::SeededRandom(Seed)
Data Source	= @VARIABLE(11)
Seed	

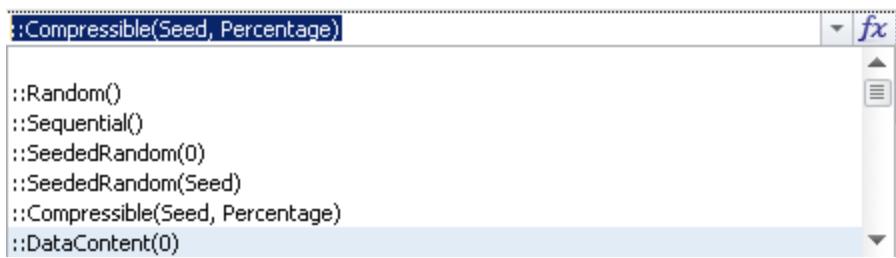
If ::Compressible(Seed,Percentage) is selected in the Data Source field then two additional input fields present:

Seed is provided for the Tester to provide a Seed value for the data that will exist in the ::Compressible file.

Percentage is provided for the Tester to provide a Percentage of the ::Compressible file that cannot be compressed (e.g. cannot be eliminated during a deduplication operation). Percentage = 100% means that the file is NOT compressible at all.

Compressible DataContent type produces a data stream in which each 1 kilobyte chunk consists of the incompressible portion and a sequence of zeros. The incompressible part is pseudo random data stream generated by an algorithm similar to the one used in SeededRandom type. The length of the

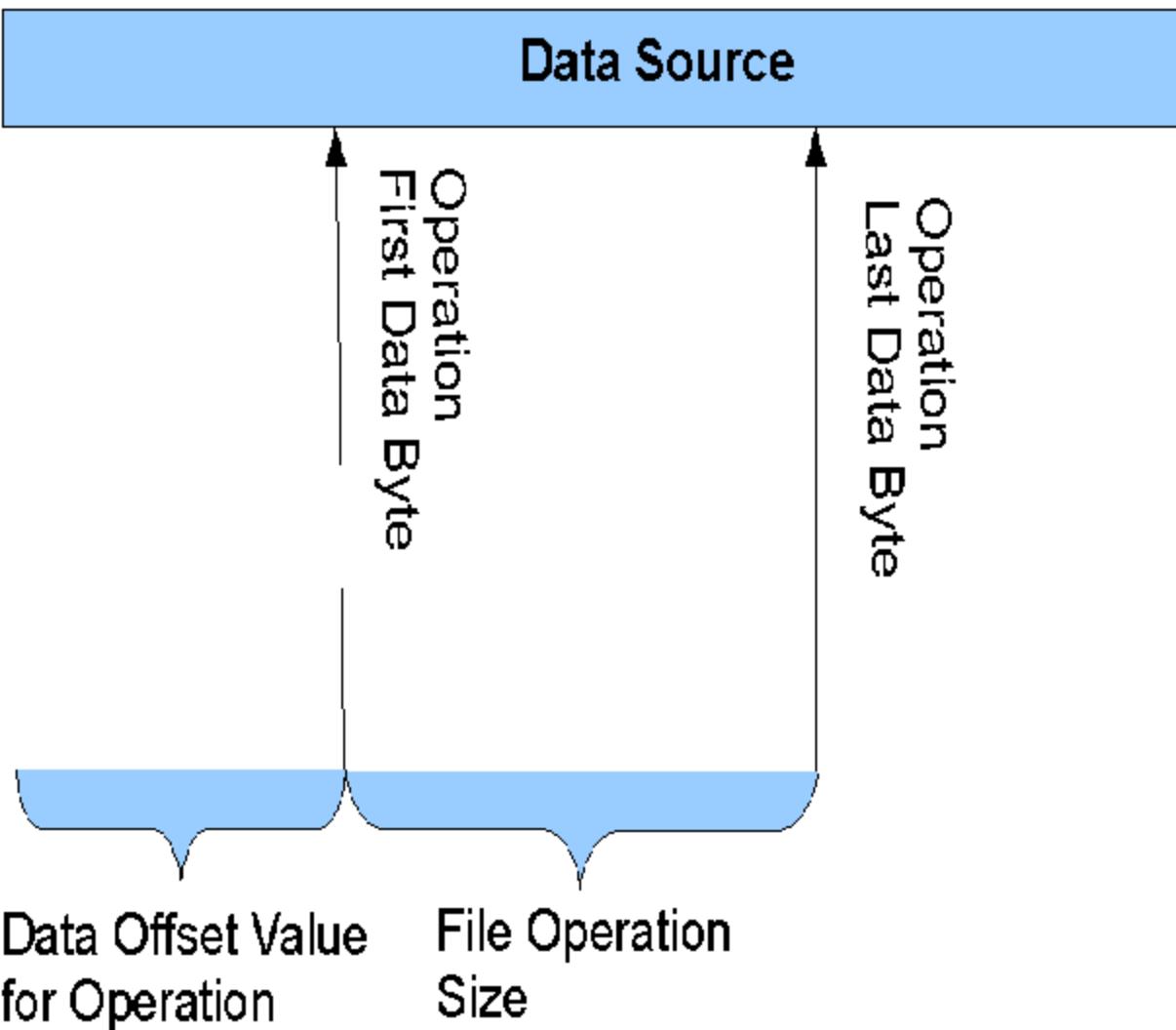
zero sequence depends on the chosen Percentage parameter and is chosen in such a way that on average most compression algorithms will give the desired percentage (ratio of compressed to original).



Seed and Percentage will accept constants, @Variable, @UP() or any Function like @RANDOM() that produces an integer as output, as input.

Data Content	
Data Source	= :Compressible(5eed, Percentage)
Seed	= @VARIABLE(11)
Percentage	= @UP(0,A)

The second property of this group is Data Source Offset. This value is the offset into the Data Source at which the data to be written begins. The value of the Data Source Offset field will be added to the value of the Initial File Offset value to determine where in the Data Source the Write operation starts taking data. A Data Offset value of 0 keeps the byte locations in the Data Source aligned with the file targeted by the Write operation.



Data Offset Behavior

In a CIFS-SMB File Write Action, there is a Data Content input area. In the Data Source field, insert a ::DataContent(X) reference and the data that is present in this file will be used for all of the Write operations.

Note that "Total Bytes To Write" property of these same Actions does not have to match the length of data identified by "Data Source" property. If "Total Bytes To Write" specifies the length that is greater than the length of Data Content item, then the Data Content item is repeated as required to match the length specified by "Total Bytes to Write" property. If "Total Bytes To Write" specifies the length that is less than the length of Data Content item, then only the required fragment of Data Content item is actually transmitted.

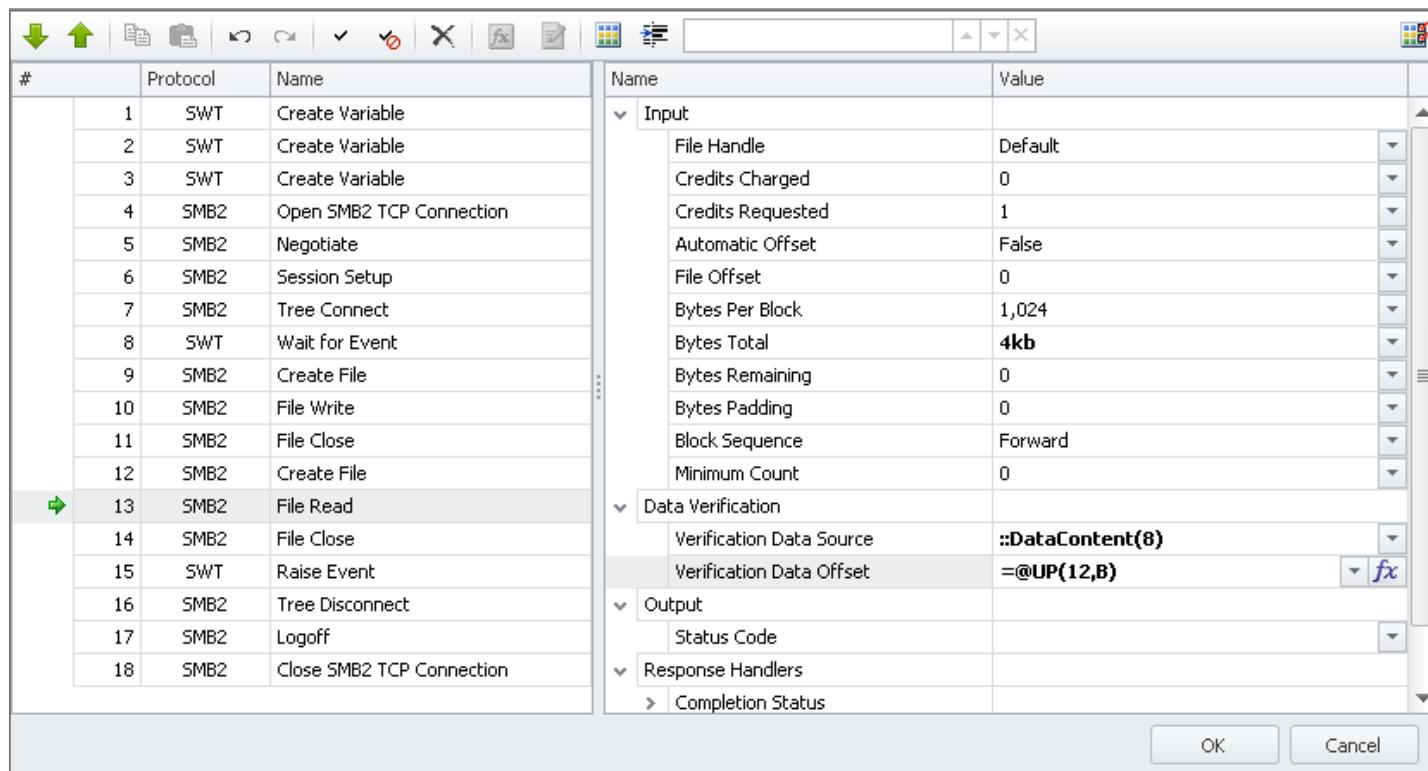
Note also that the use of large-length Data Content items may affect the time it takes a Project to start executing, since the Data Content items may require uploading to the appliance, especially if the data in Data Content items has changed, or is added for the first time.

Verifying Received Data Content

Load DynamiX CIFS-SMB, SMB2, SCSI, NFSv3, NFSv4/4.1 and HTTP read-related Actions include a "Data Verification" property group. This group includes a property "Data Verification" that allows the user to specify the Data File System Content to be used to verify the data read by these Actions. The value of the "Verification Data Source" property is the Data File System identity in the "::DataContent(n)" format. The user may type in or select the Data File System identity from the preset

values in the associated combo box.

If specified, the value for "Verification Data Source" property must match one of the identities in the associated Data File System. See "Data File System Configuration" section of this document for more information. The value of "Verify with" property defaults to blank, indicating the verification of the received data will not happen. If the value of "Verify with" property is specified, then each execution of this Action verifies the read data against data retrieved from the specified element of the Data File System. Note that the "Verification Data Source" property in the **File Read** Action should correspond to the Data Content property in the Write Action that created the data that is being read. The "Verification Data Offset" property (when present) behaves as described above in the Writing Custom Data Content discussion - it specifies where in the "Verification Data Source" data the comparison begins.



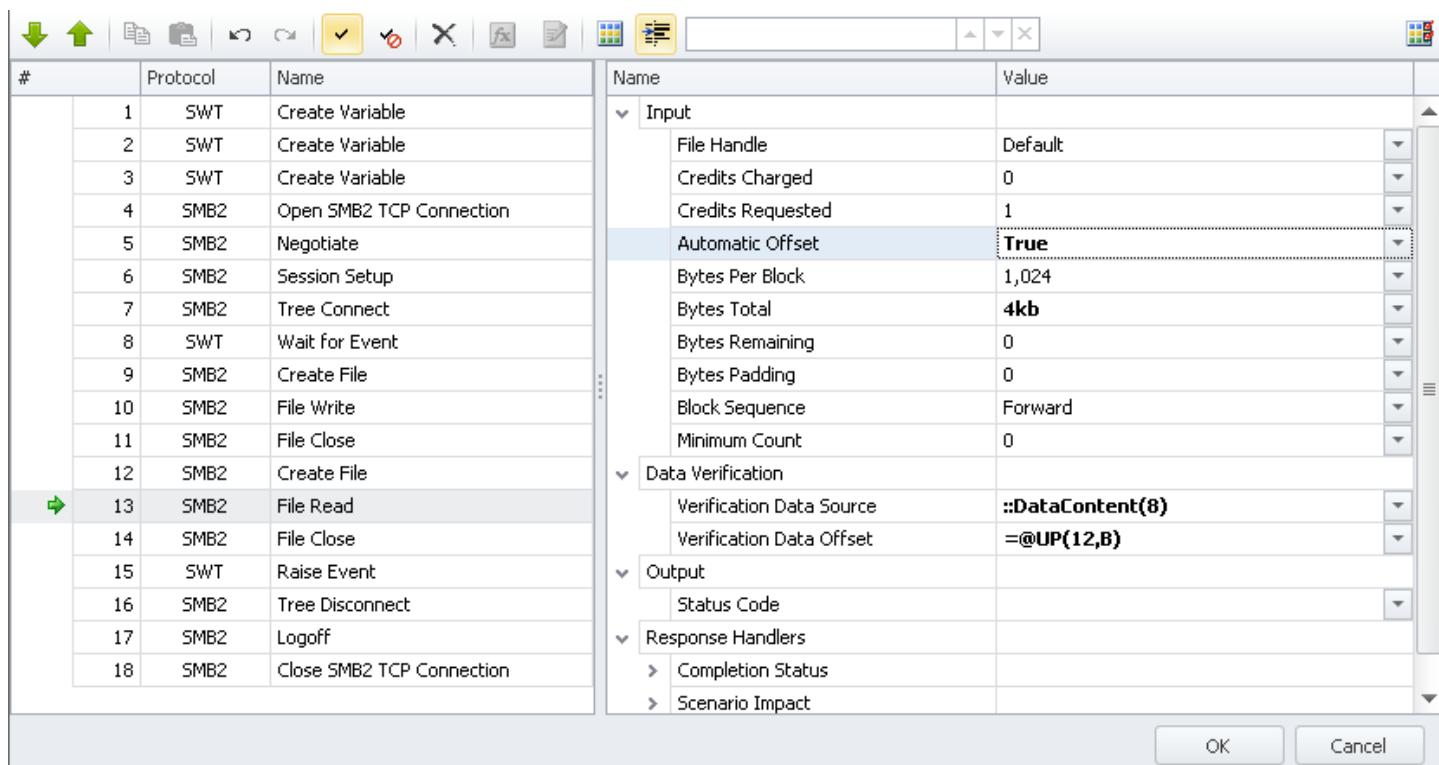
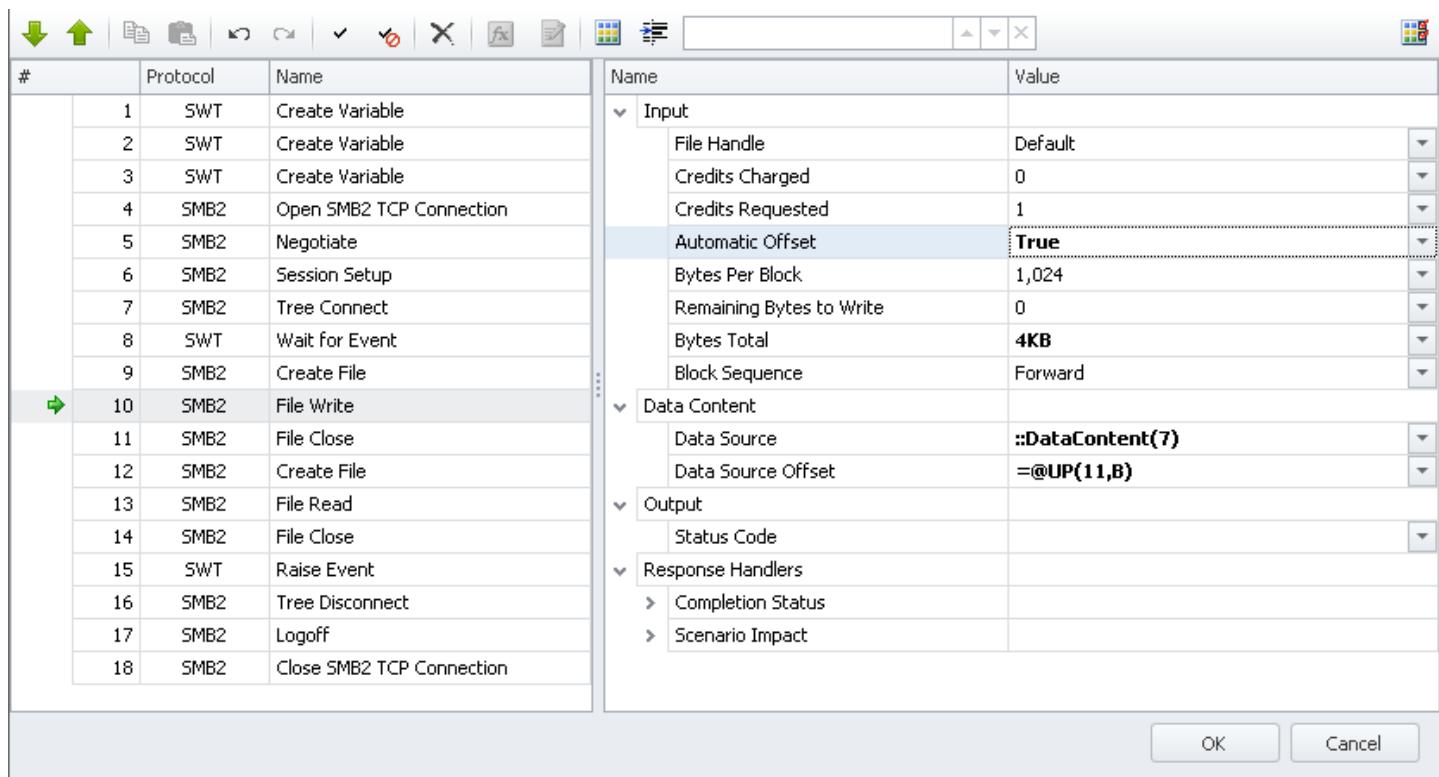
Verification of the received data succeeds only if the received data is in exact match to the data retrieved from the Data Content Profile, and fails otherwise. The appropriate total and per-command data verification statistics are accumulated for the duration of the Project execution and, in this revision, reported in the Results Log file in Results Explorer upon completion of Project's execution.

Note that if a **File Read** Action with the "Verification Data Source" property filled in, is preceded by a **File Write** Action with Verification Data Source" property set to the same Data Content identity, then verification shall succeed only if File Read "Bytes Total" is less than or equal to File Write "Bytes Total".

Automatic Offset

Load DynamiX CIFS-SMB, SMB2, SCSI (iSCSI and Fibre Channel) and NFS **Read File** and **Write File** Actions support a feature called Automatic Offset which forces the Load DynamiX Appliance Client software to automatically increment the offset for file read and write operations. For example, if a Scenario is doing a multiple reads and writes within a scenario, the Automatic Offset feature increments the value of the Offset field of the read or write Action so that the reads or writes are contiguous in the file. In the Scenario screenshots shown below, there are a pair of SMB2 **File Write** Actions followed by a pair of SMB2 **File Read** Actions and the Automatic Offset field is set to TRUE (default value for this field is FALSE). The writes and reads are both 4096 bytes long so the writes will write bytes 0 - 4095

and 4096 - 8191 and the reads will read bytes 0 - 4095 and 4096 - 8191.



This feature makes it easier for the Scenario developer to create contiguous data in an CIFS-SMB, SMB2 or NFS file.

The offset that Automatic Offset starts from does not always have to be zero. By default if the first Read/Write operation in a scenario has Auto Offset enabled, the beginning offset is 0. If Read or Write offsets are established prior to Auto Offset being used, Auto Offset will start from the current offset. For example, that the first **File Read** Action in a Scenario uses offset == 2048 and reads 4096 bytes

and the second **File Read** Action in the Scenario has Auto Offset == True. The starting offset for the second Read Action will be the ending offset of the first Read Action ($2048 + 4096 = 6144$ bytes).

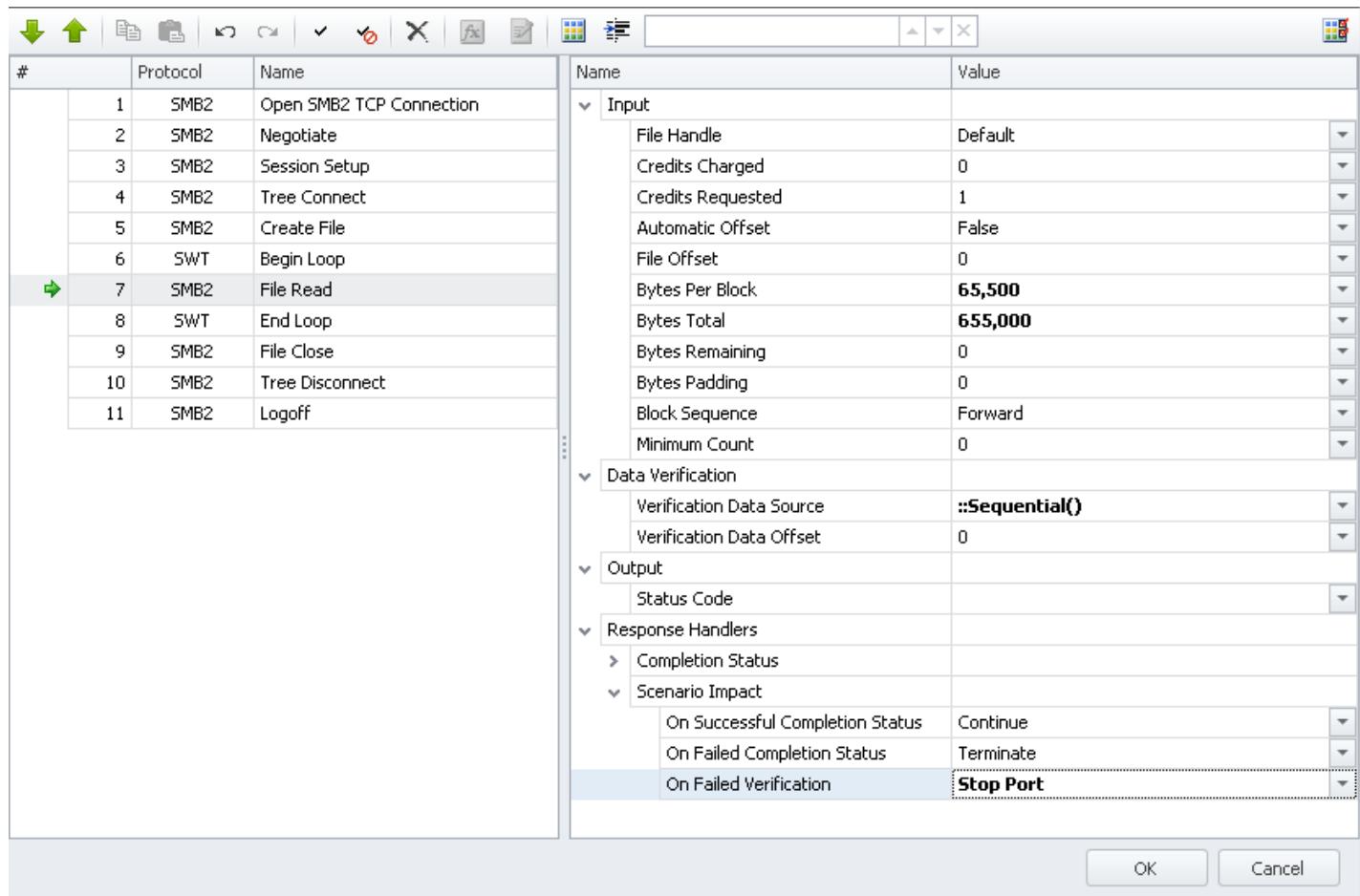
Verification of Data past End of File

Many of the storage protocols supported by Load DynamiX do not have an "End of File" error or condition - they simply return an "OK" status and 0 bytes when a Read past End of File is attempted. CIFS-SMB, NFSv3, NFSv4 and NFSv4.1 all have this behavior. SMB2 does have an End of File error. So, when Reading past the End of File, Load DynamiX Projects can behave differently both for Success or Failure of Protocol commands as well as Data Verification results.

For those protocols that do not support an End of File error (CIFS-SMB, NFSv3 and NFSv4), Read operations will show Success but Verification operations on those reads will show Failures. SMB2 will show Read command Failures but no Verification Failures.

Exit on Failed Verification

SMB Projects (CIFS-SMB and SMB2) and HTTP/HTTP Storage Projects can force a Scenario or Project to exit by using the On Failed Verification Completion Status field. In the SMB2 Scenario below, the On Failed Verification field is set to Stop Port (e.g. kill the entire Project). It can also be set equal to Terminate which will kill the Scenario. The Tester can control Scenario and Project behavior this way based on Data Verification results for CIFS-SMB and SMB2 and HTTP/HTTP Storage Projects.



Verification Data Failure files

When Data verification is enabled by the presence of a Verification Data Source in an CIFS-SMB, SMB2, NFS, HTTP or SCSI (iSCSI or Fibre Channel) Read Action and there are verification errors (as seen in the Data Verification graphs seen in the Results Explorer), the Load DynamiX Appliance

software will produce a <Protocol> Data Verification Failures file when the Project completes (stops on its own or the Stop button is pressed). These files are a .csv format file even though the file name in folder does not show the .csv extension.

The screenshot below shows what a <Protocol> Data Verification Failures file looks like in a Results Explorer folder - in this case the Data Verification results from a mutiprotocol test in which there are NFS and iSCSI data verification failures. The presence of this file indicates the Protocol that generated the errors (in this case iSCSI and NFS). These files are stored in the Port specific results folder (in this case Client Port 0) and not in the Totals or anywhere else in the results folder. The Data Verification failures will show in the Client Totals and Client Port # graphs.

If there are a lot of Data Verification Errors in a Project execution, this file can get quite large. The maximum size a Data Verification file can grow to is 2GB.



The contents of an SMB2 Data Verification file is as follows

Client Port 0(172.17.1.49 port 1) SMB2 Data Verification Failures.csv

Row Index	Time(micros)	Name	Offset	Seed	Invalid Byte	Expected B..
1	1001028	TESTFILE.062825.txt	0		0x1	0xFE
2	1001669	TESTFILE.049688.txt	0		0x1	0xFE
3	1002340	TESTFILE.080464.txt	0		0x1	0xFE
4	1002936	TESTFILE.010520.txt	0		0x1	0xFE
5	1003510	TESTFILE.003472.txt	0		0x1	0xFE
6	1004138	TESTFILE.037309.txt	0		0x1	0xFE
7	1004720	TESTFILE.037025.txt	0		0x1	0xFE
8	1005292	TESTFILE.059673.txt	0		0x1	0xFE
9	1005954	TESTFILE.099857.txt	0		0x1	0xFE
10	1006529	TESTFILE.058675.txt	0		0x1	0xFE
11	1007107	TESTFILE.093515.txt	0		0x1	0xFE
12	1007696	TESTFILE.033359.txt	0		0x1	0xFE

The contents of an iSCSI Data Verification file is as follows

Client Port 0(172.17.1.49 port 0) ISCSI Data Verification Failures.csv

Row Index	Time(micros)	Client	LUN	Offset	Seed	Invalid Byte	Expected Byte
1	2028664	172.16.160.5:512	1	0x2800		0x0	0x1
2	2029114	172.16.160.5:512	1	0x0		0x0	0x1
3	2029496	172.16.160.5:512	1	0x7800		0x0	0x1
4	2029866	172.16.160.5:512	1	0x5000		0x0	0x1
5	2030238	172.16.160.5:512	1	0xc800		0x0	0x1
6	2030630	172.16.160.5:512	1	0xa000		0x0	0x1
7	2031191	172.16.160.5:512	1	0x11800		0x0	0x1
8	2031570	172.16.160.5:512	1	0xf000		0x0	0x1
9	2032088	172.16.160.5:512	1	0x16800		0x0	0x1
10	2032866	172.16.160.5:512	1	0x14000		0x0	0x1
11	2035074	172.16.160.5:512	1	0x19000		0x0	0x1
12	2718583	172.16.160.26:512	1	0x2800		0x0	0x1
13	2719255	172.16.160.26:512	1	0x0		0x0	0x1
14	2720155	172.16.160.26:512	1	0x7800		0x0	0x1
15	2720767	172.16.160.26:512	1	0x5000		0x0	0x1

Data Verification file column headers:

Column Header	Value
Row Index	A basic row number for easier reference
Time(micros)	Time offset from the start of the Project execution in MilliSeconds
Name (CIFS-SMB/SMB2/NFS commands)	Name of the file being read in which the verification failures occurred
Path (HTTP-based protocols)	The URI, Bucket/Object, Container/Object path
Client (SCSI commands)	SCSI Client IP address and Port
LUN ((SCSI commands)	SCSI LUN being accessed
Offset	The offset into the data being read at which the verification error occurred
Seed	The Seed Value used for the DataContent source
Invalid Byte	Byte read from the specified Name or Path or LUN
Expected Byte	The byte read from the specified DataContent source

Data Verification Completion Modes

- Success - the comparison of the Expected data vs Read data is successful (Expected = Read)
- Failure - the comparison of the Expected data vs Read data is NOT successful (Expected Not = Read)
- Abort - the Data Verification operation did not complete (the operation ran out of time, server or client closed the connection, etc.)

Load DynamiX Server Scenarios

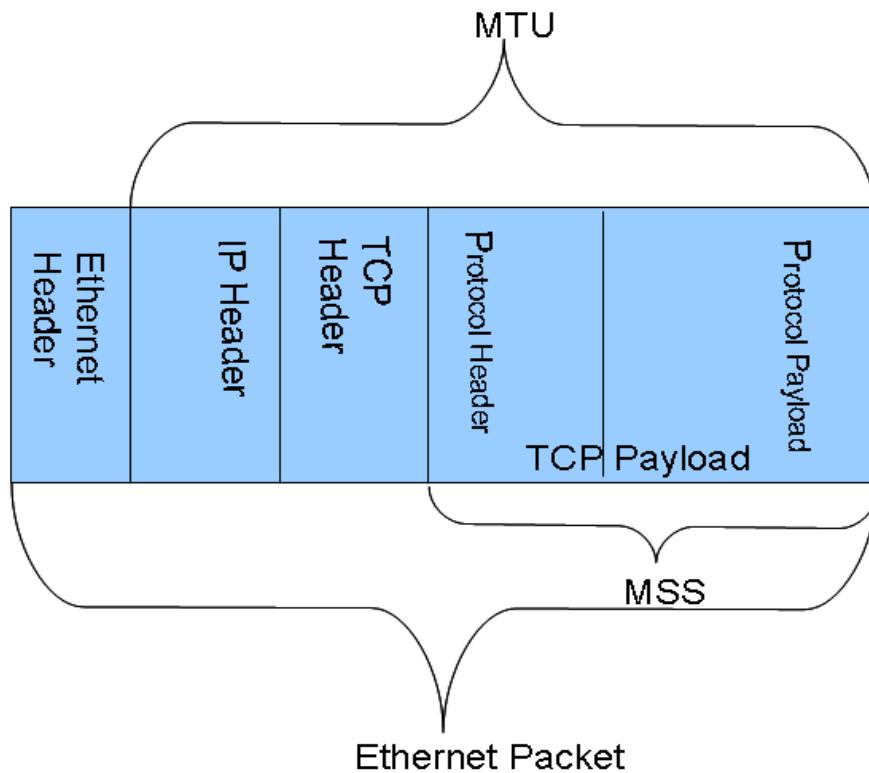
LoadDynamiX Appliance ports when acting as "Server" (e.g. SMB, SMB2, NFSv3, or HTTP Server) have 1GB of ram memory to be used as virtual disk space. When files are created and written to in this virtual disk space by a Project, the files will consume the virtual disk space unless deleted during the Project's execution. Files created by one Scenario can be accessed by future instances of that Scenario during the Project's execution. Files created on Port X cannot be accessed by a Scenario running on Port Y. All Server files are erased when the Server Scenario terminates.

Appendix: Jumbo Frames and Delayed ACK

Appendix: Jumbo Frames and Delayed ACK

TCP Communications:

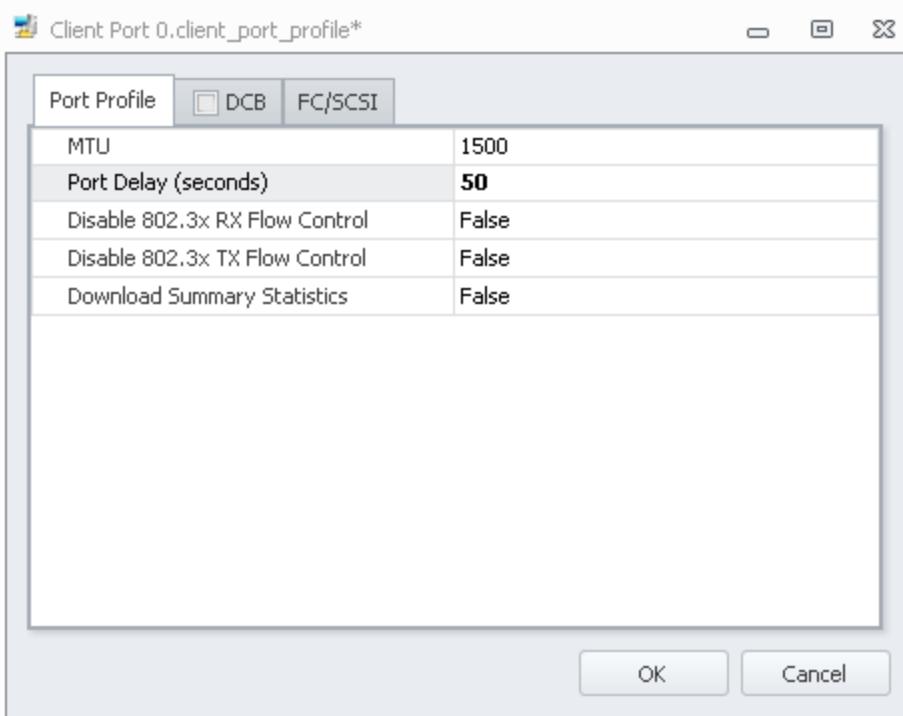
In TCP/IP over Ethernet communications, the Ethernet packet payload contains the IP and TCP headers that are necessary for routing via the Internet as well as TCP session support. The MTU (Maximum Transmission Unit) term corresponds to the maximum payload that can be transmitted in a single Ethernet packet in a given network. MSS (Maximum Segment Size) is the maximum size of a single TCP Payload. The TCP Payload section is typically further segmented into Protocol Header and Protocol Payload sections. The Protocol Header and Payload segments vary in size based on the protocol that is being sent. In the case of Load DynamiX developed tests, this section may contain HTTP headers and payload or SMB headers and payload or NFS headers and payload or iSCSI headers and payload.



Load DynamiX software allows the Tester to specify the MTU on Logical Ports. During the TCP connection startup process, each peer announces the Maximum Segment Size (MSS) that it can accept. MSS is typically equal to MTU - 40 (40 bytes for the IP and TCP headers). The default value for MTU is 1500 byte making the default MSS value 1460.

Jumbo Frames:

Client and Server Logical Ports on Load DynamiX 1G, 10G and Unified Series Appliances (models 3000, 3108, 5000, 5102, 5108S, 5108T, U1022 and U1044) support Jumbo Frames through an adjustable MTU parameter in the Logical Port properties. This property is accessible by double clicking the Logical Port object in a test timeline.



This property allows the test developer to define the MTU (Maximum Transmission Unit) for the interface that this Logical Port is associated with. The default value for this field is 1500 bytes (the standard MTU for Ethernet interfaces). Any MTU setting above 1500 allows the use of Jumbo Frames (TCP packets with a payload of more than 1460 bytes) during communications. The allowable range for the MTU property is 256 <=MTU <= 16128. For Load DynamiX 3000s with 1Gbps ports, the maximum MTU is 9216. It is possible to set the MTU higher but the Load DynamiX Client or Server software running on the Appliance will only use a maximum of 9216. On Load DynamiX Appliances with 10Gbps test ports, the maximum MTU setting is 16128.

Warning: On Load DynamiX Appliances with 1000BASE-T test ports, an MTU size of >= 8131 will cause performance degradation. It is recommended that for Projects striving for maximum performance with large packet sizes (e.g. reads or writes of > 8K bytes), that MTU size not be set to be greater than 8000. MTU size set to greater than 8131 will cause warning messages to be issued to remind the Tester of the impact of this setting.

Regardless of the MTU setting on a Logical Port the number of bytes transmitted in an Ethernet packet depends on the size of the message being transmitted and the interface at the other end of the connection. TCP uses the special MSS (Maximum Segment Size) option to announce, at the time the connection is established, the maximum amount of data that can be transmitted in a single TCP packet. Looking at a PCAP file entry for a typical TCP session, you will see something like this at the top of the PCAP file

Destination	Protocol	Info
172.16.1.23	TCP	cisco-sccp > microsoft-ds [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=0
172.16.240.1	TCP	microsoft-ds > cisco-sccp [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1460 WS=0

The two sides of this TCP connection are both announcing that they can accept up to 1460 bytes in the TCP packet payload. The MTU for this interaction is 1500 bytes, there are 40 bytes of header information so that leaves 1460 bytes of data.

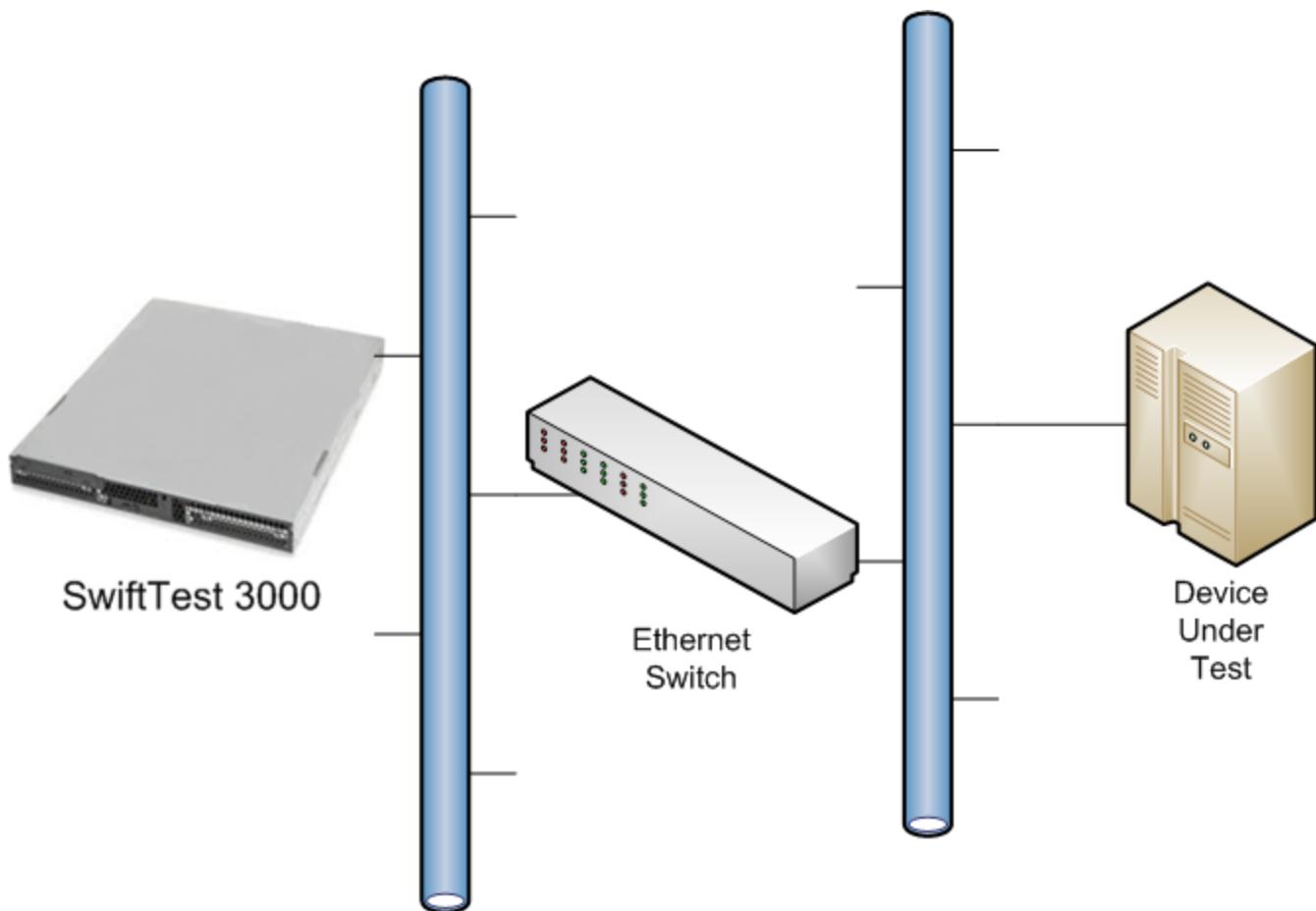
When the MTU value is increased to 1900 on one side of the connection and the other left at the default 1500 byte setting, the PCAP entry would look like this

Protocol	Info
TCP	cisco-sccp > microsoft-ds [SYN] Seq=0 Win=65535 Len=0 MSS=1860 WS=0
TCP	cisco-sccp > microsoft-ds [SYN] Seq=0 Win=65535 Len=0 MSS=1860 WS=0
TCP	microsoft-ds > cisco-sccp [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=0
SMB	Negotiate Protocol Request
TCP	microsoft-ds > cisco-sccp [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=0

We see in the above example that it is possible for the two interfaces in a TCP connection to support different size MTUs and thus to transmit different size packets to each other.

When data content of the Payload area of a packet is less than the MSS size, announced during the TCP connection handshake, only the bytes necessary are transmitted between the two interfaces (e.g. if a message only requires 138 bytes of Payload then only 138 bytes are transmitted regardless of the MSS size setting). When the Payload content for a packet is larger than the MSS size (e.g. a filesystem Read or Write) then that data is partitioned into multiple MSS-sized transmissions. At a default 1500 byte MTU setting, a 4096 byte read requires three packets to deliver 4096 bytes of data. On an Logical Port with a 5000 byte MTU setting, only a single packet is required to deliver the data. So, a larger MTU size can improve network performance by reducing the number of packets that must be transmitted and received. If network test traffic is predominantly operations that result in the transfer of data greater than 1460 bytes in length, setting the MTU size for the Logical Ports involved in those tests to a value that allows reads or writes to be delivered in a fewer packets can be beneficial.

For Ethernet communications using Payloads of greater than the default MTU size to operate correctly, it must be possible to transmit or receive packets of the specified size throughout the entire network. For example, in the following network configuration,



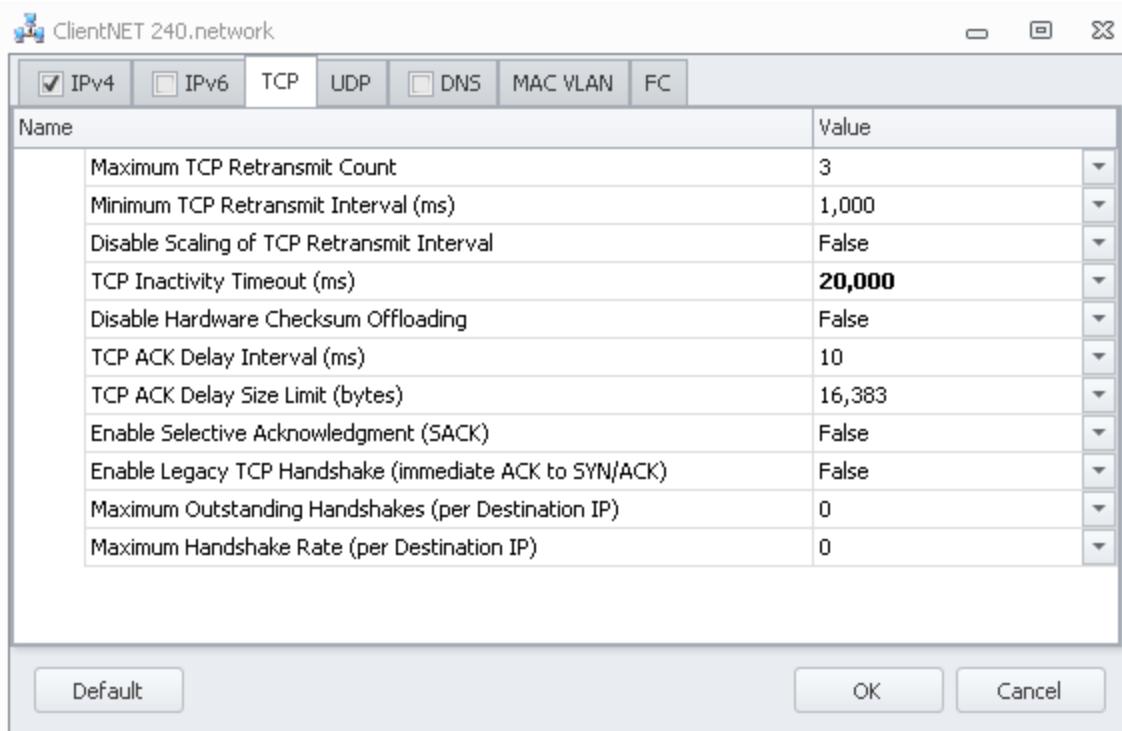
for both the Load DynamiX 1G Series Appliance and the Device under Test to use Payloads of 5000

bytes, all of the elements of the network, the Load DynamiX 1G Series Appliance, the Ethernet Switch, and the Device Under Test must support individual Ethernet packets of greater than 5014 bytes (14 bytes for the Ethernet header + 5000 for the Payload).

Currently, MTU size must be set manually for all Logical Ports involved in a test. MTU size set outside the allowable range of 256 <= MTU <= 16128 will result in either 256 or 16128 in the MTU field.

Delayed ACK:

Delayed ACKs allow the TCP stack to delay sending ACK messages for every packet they receive. This allows the TCP stack to focus more on data packets rather than overhead. The ACK interval is set in the Load DynamiX Network Profile property box for a test component. Each Network Profile associated with a Logical Port can have its own TCP ACK Delay Interval. Double click on a Network Profile object in a Timeline or in the Project Explorer and you will see



Test developers can set the TCP ACK Delay Interval value which can range from 0 to 50% of the Minimum TCP Retransmit Interval value (in this case shown above, the maximum TCP ACK Delay Interval would be 500 ms).

The default value of 10 is a slight performance improvement over the default value from previous Load DynamiX TDE releases (0). Changes in this value can improve or degrade performance. Viewing the impact of TCP Delay ACK changes can be accomplished via PCAP packet dump display tools such as Wireshark.

TCP Delay ACK Behavior:

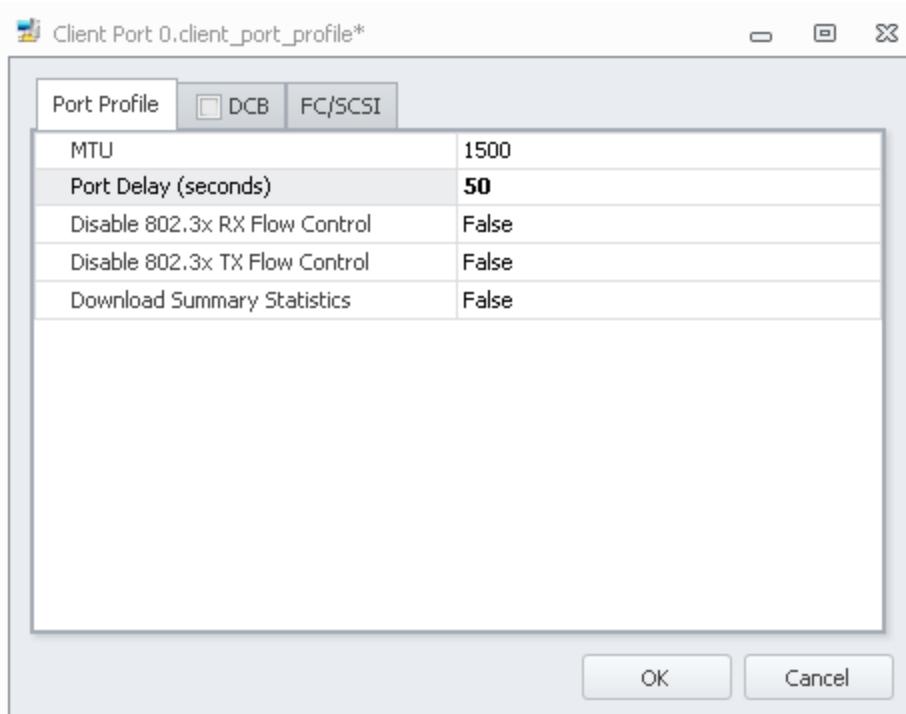
The TCP ACK Delay Interval controls how Load DynamiX will send ACK messages when responding to received data. When TCP Delayed ACK is configured in a Load DynamiX Network Profile (TCP ACK Delay Interval set to any value > 0), each TCP Connection within that profile will delay sending TCP ACKs for received data until one of the following happens:

1. Load DynamiX needs to transmit data on this connection
2. The TCP Delay ACK Interval expires
3. Received and unacknowledged bytes of data, equal to the TCP ACK Delay Size Limit (default = 16383) in the Network Profile for this network, is accumulated.

Setting TCP ACK Delay Interval to 0 will disable the TCP Delay ACK feature and all received data will be ACK'ed immediately after it has been received.

Summary Statistics:

In previous releases of the Load DynamiX Appliance software, a summary statistics file (all statistics captured during Project execution) was uploaded to the Load DynamiX TDE at the completion of Project execution. For very long Project execution runs (nonstop for days or weeks), this file grew to be very large and would delay getting access to Project execution results while it was being uploaded from the Appliance. This information is currently unused by the Load DynamiX TDE. So, the summary statistics are not uploaded at the end of Project execution. To force the upload of the Summary Statistics information set Download Summary Statistics in the Logical Port Resource to True and click OK. With this setting in place, every time this Project is executed, the Summary Statistics file will be downloaded to the Load DynamiX Management Station at the completion of Project execution.



At this time there is no reason to ever check that box.

Advanced Concepts: Response Handling

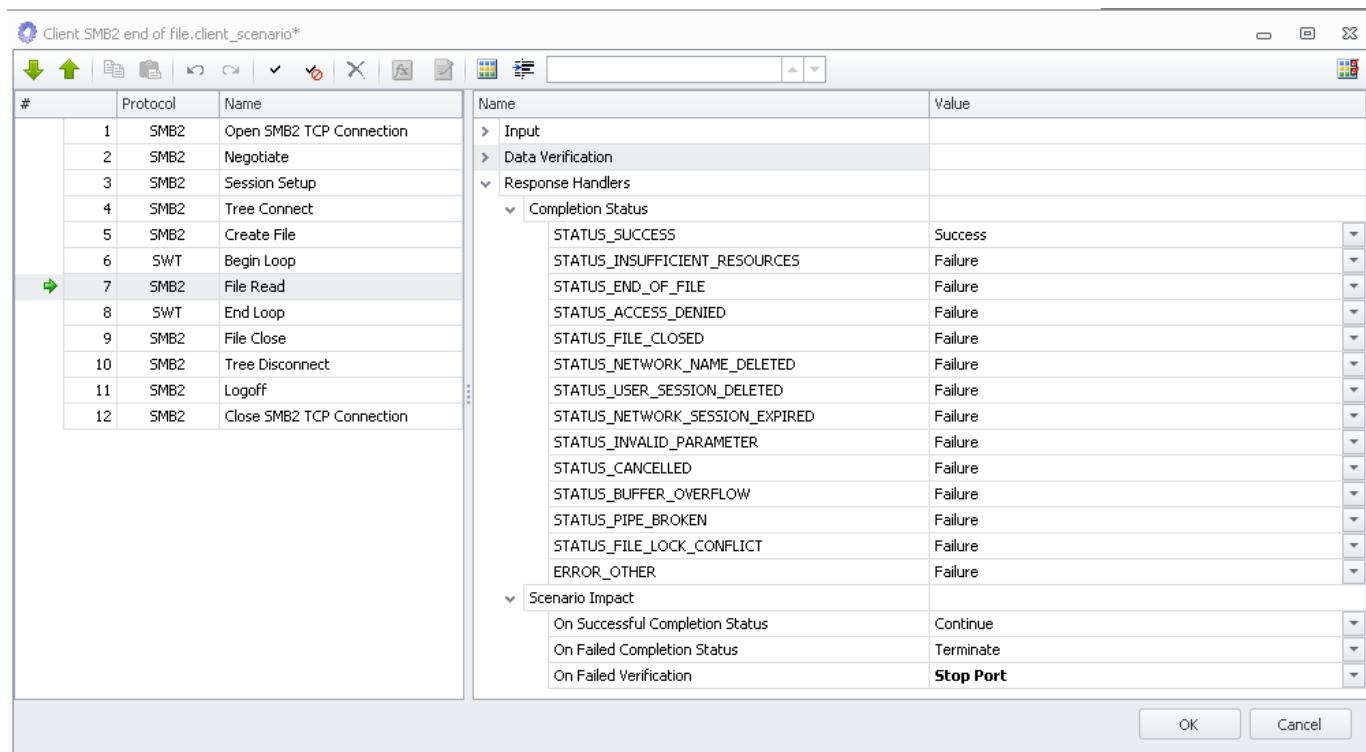
Advanced Concepts: Response Handling

All Load DynamiX SMB, NFS, HTTP/HTTP Storage and SCSI Actions contain a Response Handlers property. The purpose of this property is to give the Tester a means to determine the next step to be taken based on the Response Code that is returned as a result of executing that Action. The set of Response Codes that can be returned vary on the Action that is being executed but the general processing that is performed is consistent across SMB (CIFS-SMB, SMB2, MSRPC, SMB3), NFS (NFSv3, NFSv4, NFSv4.1), HTTP (HTTP, HTTPS), HTTP Storage (Amazon S3, OpenStack Swift, OpenStack Cinder, Keystone Authentication, and CDMI) and SCSI (iSCSI, Fibre Channel) Actions.

Response Handling is not currently supported by Kerberos Actions.

Completion Status

In the example below (SMB2 File Read), the Completion Status property lists the known Response Codes that can be returned as the result of a **File Read**. When a returned Response Code matches an item in the list, the state of Completion Status is set to either Success or Failure. If the Response Code returned is none of the specified Response Codes under Completion Status then the state is set to whatever **ERROR_OTHER** is set to be.



See [Appendix: Scenario Control Actions](#) for the use of the Completion Status values (seen by doing a mouse-over of the elements in the left column) in conjunction with the If/Else/Else If/End If Actions for Branching Control within Scenarios.

Scenario Impact

The last aspect of Response Handling is Scenario Impact - e.g. what should the Scenario do based on the state of Completion Status. Load DynamiX TDE gives the Tester the ability to determine what the next step for the Scenario should be. The On Successful Completion Status and On Failed Completion Status properties specify what the current Scenario should do if the state of the Completion Status is Success or Failure. The Tester can specify

On Successful Completion Status

Either **Continue** or **Terminate** the current Scenario

On Failed Completion Status

Either **Continue** or **Terminate** the current Scenario

On Failed Verification (only applicable to Read Actions that have the Data Verification property set)

Either **Continue** or **Terminate** the current Scenario or the entire Project (**Stop Port**) based on Data Verification results

Continue = Execute the next Action in this Scenario

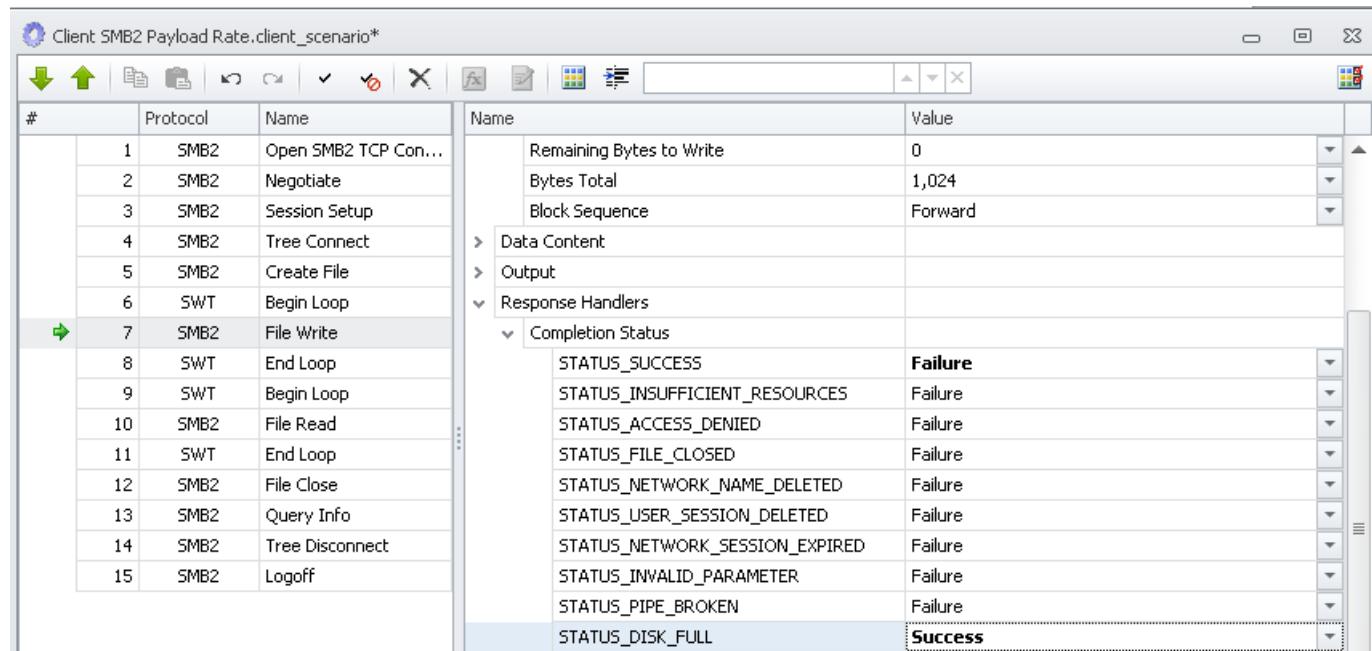
Terminate = Abort this Scenario (stop executing this scenario at this Action).

Stop Port = Abort the Project

Negative Condition Testing

Response Handler processing allows Testers to develop tests that can intentionally create and test for negative conditions (return codes that indicate some kind of negative status) such as a "Disk Full" error when attempting to write to a full disk or a "File Closed" error when attempting to write to a file that was never opened or was closed unintentionally. In the SMB2 examples below, the File Write Action's Completion Status inputs have been changed to expect (set to Success) the return codes that should be returned when attempting to write to a full disk or write to a file that was not open. Note that STATUS_SUCCESS has been changed to Failure since a "success" in these cases is a failure.

Disk Full



The screenshot shows a software interface for testing SMB2 operations. On the left, a list of actions is shown in a table:

#	Protocol	Name
1	SMB2	Open SMB2 TCP Con...
2	SMB2	Negotiate
3	SMB2	Session Setup
4	SMB2	Tree Connect
5	SMB2	Create File
6	SWT	Begin Loop
7	SMB2	File Write
8	SWT	End Loop
9	SWT	Begin Loop
10	SMB2	File Read
11	SWT	End Loop
12	SMB2	File Close
13	SMB2	Query Info
14	SMB2	Tree Disconnect
15	SMB2	Logoff

On the right, a detailed view of the 'File Write' action is shown, specifically focusing on the 'Completion Status' settings:

Name	Value
Remaining Bytes to Write	0
Bytes Total	1,024
Block Sequence	Forward
Data Content	
Output	
Response Handlers	
Completion Status	
STATUS_SUCCESS	Failure
STATUS_INSUFFICIENT_RESOURCES	Failure
STATUS_ACCESS_DENIED	Failure
STATUS_FILE_CLOSED	Failure
STATUS_NETWORK_NAME_DELETED	Failure
STATUS_USER_SESSION_DELETED	Failure
STATUS_NETWORK_SESSION_EXPIRED	Failure
STATUS_INVALID_PARAMETER	Failure
STATUS_PIPE_BROKEN	Failure
STATUS_DISK_FULL	Success

File Closed

#	Protocol	Name	
1	SMB2	Open SMB2 TCP Con...	
2	SMB2	Negotiate	
3	SMB2	Session Setup	
4	SMB2	Tree Connect	
5	SMB2	Create File	
6	SWT	Begin Loop	
7	SMB2	File Write	
8	SWT	End Loop	
9	SWT	Begin Loop	
10	SMB2	File Read	
11	SWT	End Loop	
12	SMB2	File Close	
13	SMB2	Query Info	
14	SMB2	Tree Disconnect	
15	SMB2	Logoff	

Completion Status

Status	Action
STATUS_SUCCESS	Failure
STATUS_INSUFFICIENT_RESOURCES	Failure
STATUS_ACCESS_DENIED	Failure
STATUS_FILE_CLOSED	Success
STATUS_NETWORK_NAME_DELETED	Failure
STATUS_USER_SESSION_DELETED	Failure
STATUS_NETWORK_SESSION_EXPIRED	Failure
STATUS_INVALID_PARAMETER	Failure

"Don't Care" Actions

Response Handler processing allows Testers to define Actions in a Scenario in which they do not care about the success or failure of that particular Action. To make an Action "Don't Care", simply set both On Successful Completion Status and On Failed Completion Status to "Continue".

Viewing Response Handler Results

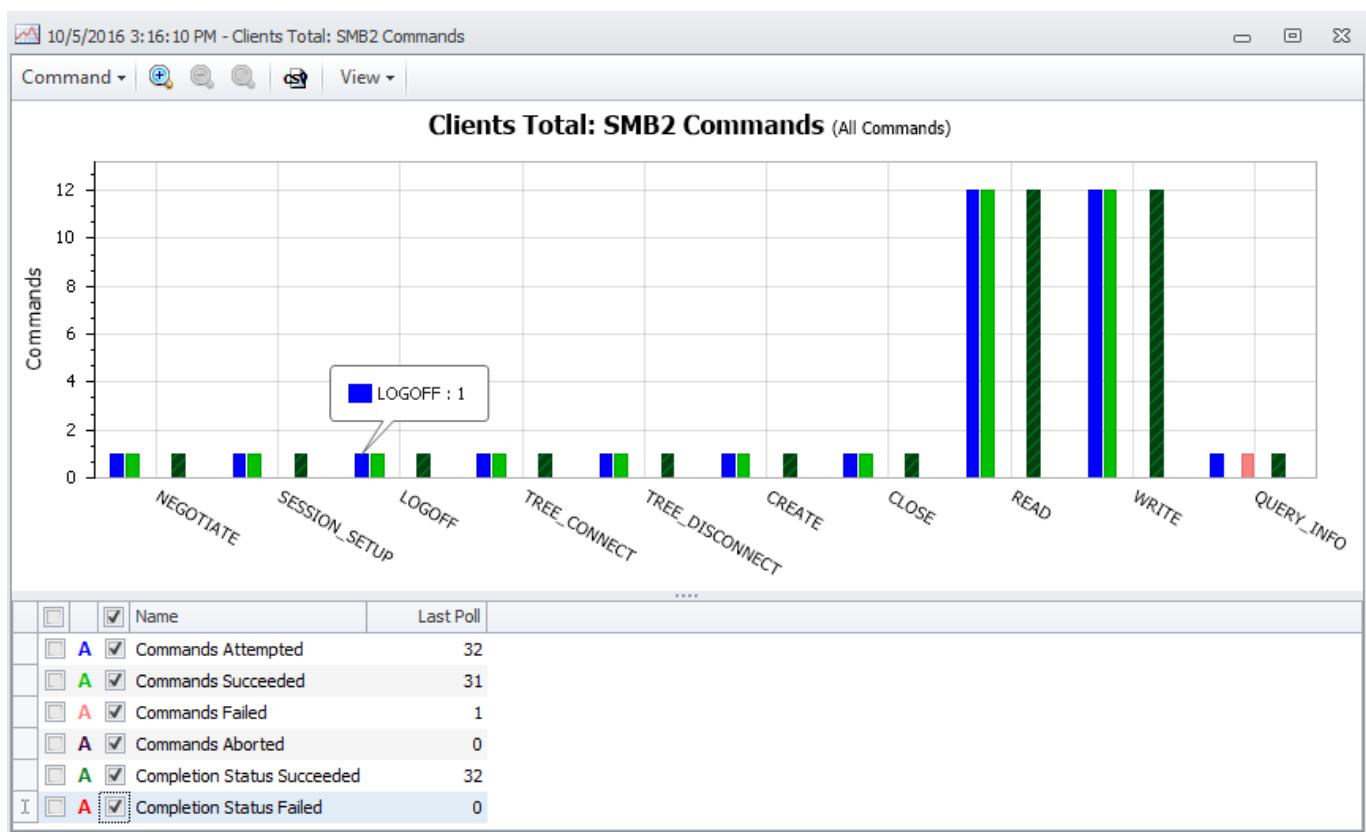
Response Handling data can be viewed in the Results Folder in two areas:

- Command Graphs
- Log Files

Command Graphs

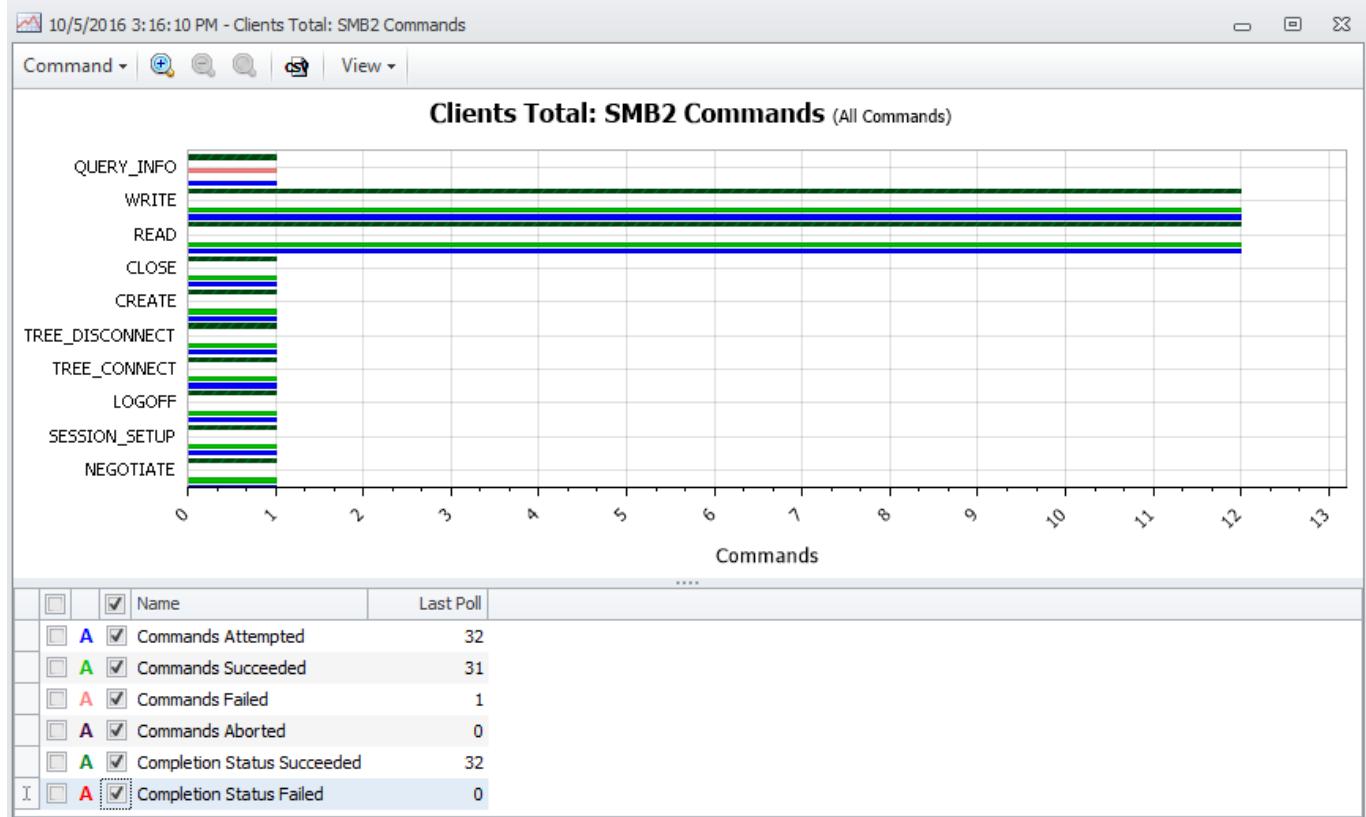
Response Handler data can be displayed along with Command data so, for example, the SMB Commands graph for a read and write Scenario with ten commands might look like this with the Response Handler data (Success and Failure) enabled. This graph shows the Response Handler data aligned with the Action execution data (Blue bar is count of Action Attempts, Green bar is count of Action Successful Completions, the green crosshatch bar (e.g. Write command) is the count of Success behaviors and the red crosshatch bar (QUERY_INFO Action) is the count of Failure behaviors. The Project that generated these graphs forced the QUERY_INFO Action to fail but set the Scenario Impact On Failed Completion Status to **Continue** so that the Scenario would continue executing even though the **QUERY_INFO** Action failed each time it was executed.

The default view of Command Graphs is Vertical bars as shown below but with the Response Handler data enabled (Completion Status Succeeded and Completion Status Failed checked).

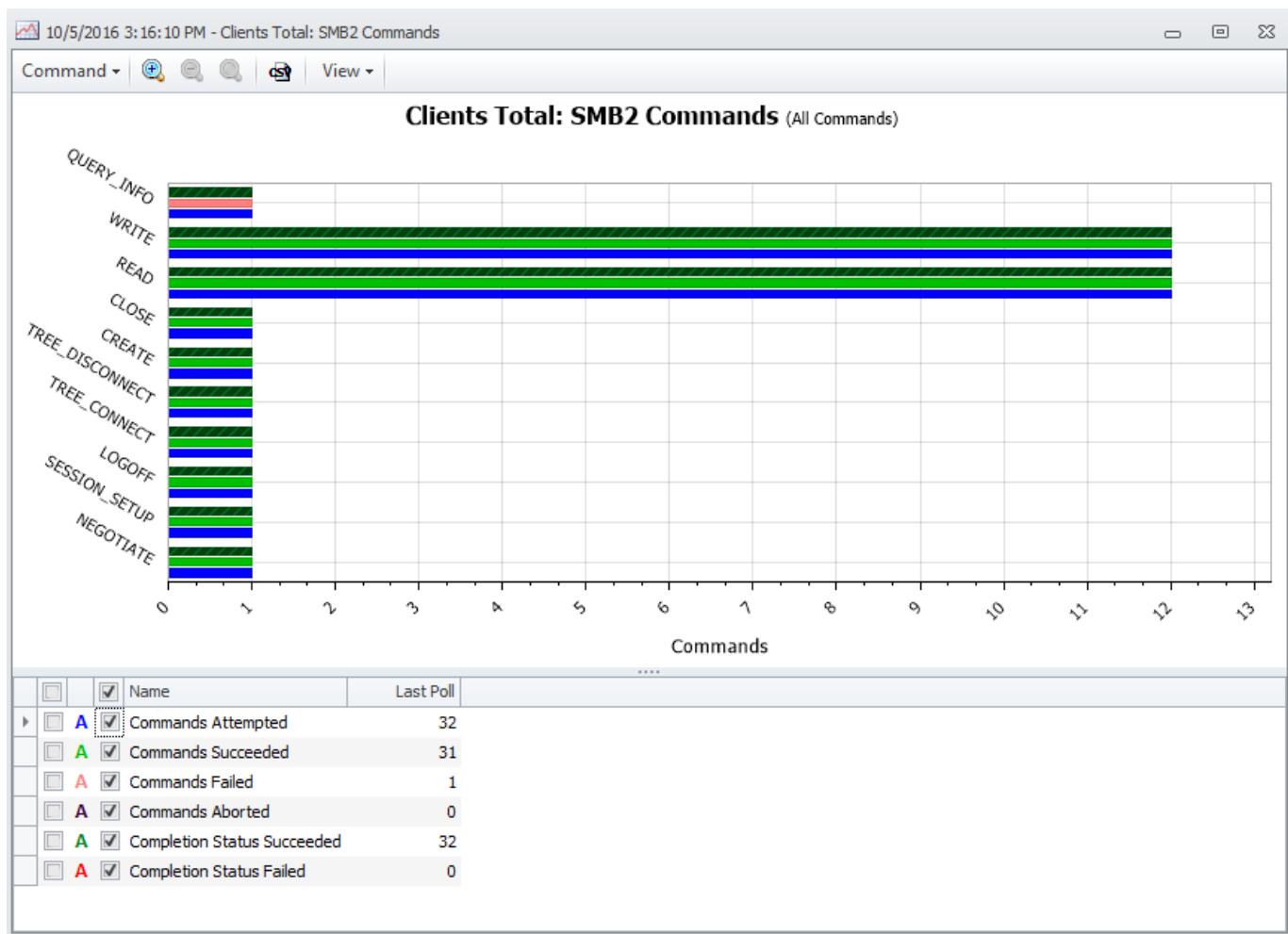


These Command Graphs can be viewed in alternate ways using the View menu function:

ROTATED (HORIZONTAL)



STACKED (where like data is displayed in same bar- in this case the statistics are Rotated and Stacked):



In many command graphs, the Response Handler data (counts) will be the same as the Command data - that is, for commands that execute without Aborts, Attempts = (Succeeds+Fails) = (Success + Failure). If there are Aborts present then there will be a disparity between the Command data (Attempts, Succeeds, Fails and Aborts) and Response Handler data (Success and Failure) because Aborts are not recorded and graphed in the Response Handler data.

HTTP (HTTP, HTTPS, HTTP Storage) Response Handling

See [Advanced Concepts: HTTP/HTTPS](#) for details regarding HTTP Response Handling.

FC/iSCSI/SCSI Response Handling

See [Reference: FC/SCSI/iSCSI Commands and Behaviors](#) for details regarding FC/iSCSI/SCSI Response Handling.

Client Log Files

Another way to view the response Handler data is in the Results Folder Client Log file. See below for an example where the Response Handler data mirrors the command data. Notice the identical Action and Responses Handled data for all of the CIFS-SMB commands shown. Had there been any Aborts in the Actions table, the Responses Handled table would show slightly different counts because Aborts are not counted by Response Handlers.

Line	Type	Date / Time	Text				
			SMB_COM Actions:	Attempted	Succeeded	Failed	Aborted
17	Info	3/31/2011 1:02:42 PM	SMB_COM Actions:				
18	Info	3/31/2011 1:02:42 PM	=====				
19	Info	3/31/2011 1:02:42 PM	Total:	182651	179359	3275	17
20	Info	3/31/2011 1:02:42 PM	-----				
21	Info	3/31/2011 1:02:42 PM	SMB_COM_CLOSE	6637	6637	0	0
22	Info	3/31/2011 1:02:42 PM	SMB_COM_READ_ANDX	67304	67300	1	3
23	Info	3/31/2011 1:02:42 PM	SMB_COM_WRITE_ANDX	65557	65548	0	9
24	Info	3/31/2011 1:02:42 PM	SMB_COM_TREE_DISCONNECT	6637	6637	0	0
25	Info	3/31/2011 1:02:42 PM	SMB_COM_NEGOTIATE	6653	6651	0	2
26	Info	3/31/2011 1:02:42 PM	SMB_COM_SESSION_SETUP_ANDX	6651	6651	0	0
27	Info	3/31/2011 1:02:42 PM	SMB_COM_LOGOFF_ANDX	6637	6635	0	2
28	Info	3/31/2011 1:02:42 PM	SMB_COM_TREE_CONNECT_ANDX	6651	6650	0	1
29	Info	3/31/2011 1:02:42 PM	SMB_COM_NT_CREATE_ANDX	6650	6650	0	0
30	Info	3/31/2011 1:02:42 PM	TRANS2_QUERY_PATH_INFORMATION	3274	0	3274	0
31	Info	3/31/2011 1:02:42 PM	=====				
32	Info	3/31/2011 1:02:42 PM					
33	Info	3/31/2011 1:02:42 PM	=====				
34	Info	3/31/2011 1:02:42 PM	SMB_COM Responses Handled:	Attempted	Succeeded	Failed	Aborted
35	Info	3/31/2011 1:02:42 PM	=====				
36	Info	3/31/2011 1:02:42 PM	Total:	182651	179360	3274	17
37	Info	3/31/2011 1:02:42 PM	-----				
38	Info	3/31/2011 1:02:42 PM	SMB_COM_CLOSE	6637	6637	0	0
39	Info	3/31/2011 1:02:42 PM	SMB_COM_READ_ANDX	67304	67301	0	3
40	Info	3/31/2011 1:02:42 PM	SMB_COM_WRITE_ANDX	65557	65548	0	9
41	Info	3/31/2011 1:02:42 PM	SMB_COM_TREE_DISCONNECT	6637	6637	0	0
42	Info	3/31/2011 1:02:42 PM	SMB_COM_NEGOTIATE	6653	6651	0	2
43	Info	3/31/2011 1:02:42 PM	SMB_COM_SESSION_SETUP_ANDX	6651	6651	0	0
44	Info	3/31/2011 1:02:42 PM	SMB_COM_LOGOFF_ANDX	6637	6635	0	2
45	Info	3/31/2011 1:02:42 PM	SMB_COM_TREE_CONNECT_ANDX	6651	6650	0	1
46	Info	3/31/2011 1:02:42 PM	SMB_COM_NT_CREATE_ANDX	6650	6650	0	0
47	Info	3/31/2011 1:02:42 PM	TRANS2_QUERY_PATH_INFORMATION	3274	0	3274	0

Advanced Concepts: Variables and Aliases

Advanced Concepts: Variables and Aliases

Variables

The input to Load DynamiX Action input fields may be Strings (text) or Integers (numbers). Some Action input fields allow the input to be defined by the output of a Function (see the [Appendix: Functions and Formula](#) for a description of allowable Function components). One of the allowed components of a Function are Variables. Variables are created by the Create Variable Action and contain either Strings or Numbers. A Variable retains the value set at creation time unless that Variable is Updated at some later point in a Scenario. An example of the use of a Variable might be to identify a file name that is to be used more than once in a Scenario.

Creating and Using Variables

A sample CIFS-SMB Scenario that creates and uses a Variable would be:

Name	Value
Alias	filename
Type	General
Value	=@UP(9, FILE_NAME)

Name	Value
Variable	8: Variable

This screenshot shows the **Create Variable** Action that creates a Variable with the Alias "filename" with the contents of the [Global User Parameter](#) file at index 9, column alias FILE_NAME. This Scenario will maintain a value for this Variable throughout the life of the Scenario. It may be used wherever a Function input is allowed by referring to the Action that Creates the Variable using the @VARIABLE Function. From this point on, unless Updated, the Variable referred to as @VARIABLE(8) or @VARIABLE(filename) or @VAR(filename) or @VAR(8) will maintain this value. It could be used repeatedly always providing the same result to the input field in which it is used.

The next screen shot shows the Variable being used in a CIFS-SMB Create Or Open File Action.

Updating Variables

Variables maintain the value set at Create time for the duration of a Scenario unless they are Updated during Scenario execution. Any reference to the Variable is always by the line number in which it is created. So, in the Scenario below, the Variable created in line #3 is referred to later by @VARIABLE(6) or @VAR(readyfor).

In this Scenario, a Variable with Alias "readyfor" is created in line #6 with the contents of the string "ReadyFor" and updated in line #7.

The Update Variable Action in line #7 changes the contents of @VARIABLE(readyfor) from "ReadyFor" to "ReadyFor" appended with the current contents of the Global User Parameter File, index 0, column C. @VARIABLE(readyfor) will maintain this value for the duration of the Scenario execution.

The examples above show a Variable with string or text contents. Variables may also be assigned

numbers and used in fields that require numeric input.

For more details on Aliases, see the Aliases section below.

VARIABLE CAVEATS

- `@VARIABLE()` may not be used as inputs to Open <Prot> TCP Connection Actions where <Prot> is any of the following protocols: SMB, SMB2, NFS (v2,3,4,4.1), iSCSI, HTTP, KERBEROS.
- Math operations on Variables is not supported: `=@VARIABLE(3) + @VARIABLE(10)` does not produce an integer with a value of the sum of the two VARIABLES. Formulas may be used to perform math operations using VARIABLES. See [Appendix: Functions and Formula](#) for details.

Aliases

Generically, an Alias is a name for an Action or a User Parameter file column. They are used in the same way, as an alternate way to refer to the Action or Column. Without Aliases, references to Actions are by Line Number and User Parameters are by User Parameter File index and Column. The discussion below focuses on Aliases used as a means to refer to an Action. For a detailed description of User Parameter File Column Aliases see [Advanced Concepts: User Parameters](#).

Many of the Load DynamiX Scenario Control Actions may be given an Alias. Using an Alias in these commands is purely optional and intended to help the Load DynamiX Project developer provide more readable/understandable Scenarios. The Scenario Control commands that support Aliases are:

IOManager
Begin Thread
Begin Loop
Create Variable
Formula

The Project shown below demonstrates how Aliases might be used in a Scenario. The Scenario without Aliases looks like this:

The screenshot shows the Load DynamiX Scenario Editor interface. On the left is a list of actions in a table:

#	Protocol	Name
1	SMB2	Open SMB2 TCP Connection
2	SMB2	Negotiate
3	SMB2	Session Setup
4	SMB2	Tree Connect
5	SWT	Create Variable
6	SMB2	Create File
7	SWT	Create Variable
8	SWT	Begin Loop
9	SWT	Formula
10	SMB2	File Write
11	SWT	End Loop
12	SMB2	File Close
13	SMB2	Tree Disconnect
14	SMB2	Logoff

On the right, the properties for action #10 (File Write) are displayed in a tree view:

- Input**
 - File Handle: Default
 - Credits Charged: 0
 - Credits Requested: 1
 - Automatic Offset: False
 - File Offset: `=@FORMULA(9)`
 - Bytes Per Block: `=@VAR(5)`
 - Remaining Bytes to Write: 0
 - Bytes Total: `=@VARIABLE(5)`
 - Block Sequence: Forward
- Data Content**
 - Data Source
 - Data Source Offset: 0
- Output**
 - Status Code
- Response Handlers**

At the bottom right are **OK** and **Cancel** buttons.

but the Scenario with Aliases looks like this (with the use of Aliases highlighted):

While this is a relatively small and uncomplicated Scenario, a very large or complicated Scenario that does not use Aliases, as in the first Scenario presented, could be difficult to read and/or understand. In the second Scenario, Aliases document the contents of the Create Variable Actions in lines 5 and 7 and document the output of the Formulas in lines 9 and 12.

The File Write Action in line 10 shows the use of Aliases in defining the inputs to File Offset, Bytes Per Block and Bytes Total. The Aliases make the references much clearer than if the line # mode was used: @FORMULA(9), @VARIABLE(5) and @VAR(5).

In the Actions in lines 5 and 7, the creation of Variables with Aliases containing the values 1024 and 200, and the use of an Alias for the Begin Loop Action is demonstrated:

#	Protocol	Name
1	SMB2	Open SMB2 TCP Connection
2	SMB2	Negotiate
3	SMB2	Session Setup
4	SMB2	Tree Connect
5	SWT	Create Variable oneKB
6	SMB2	Create File
7	SWT	Create Variable twohundred

Name	Value
Input	
Alias	twohundred
Type	General
Value	200
Output	
Variable	7: Variable

In the Actions in lines 9 (the FORMULA for write_offset) and 12 (the FORMULA for total_bytes), these Aliases are used instead of the line # of the Action:

#	Protocol	Name
1	SMB2	Open SMB2 TCP Connection
2	SMB2	Negotiate
3	SMB2	Session Setup
4	SMB2	Tree Connect
5	SWT	Create Variable oneKB
6	SMB2	Create File
7	SWT	Create Variable twohundred
8	SWT	Begin Loop mainloop
9	SWT	Formula writeoffset

Name	Value
Input	
Alias	writeoffset
Description	write offset
Formula	blocksize * index
BLOCKSIZE	@VAR(oneKB)
INDEX	@LI(mainloop)
Output	
Output	9: Formula

#	Protocol	Name
1	SMB2	Open SMB2 TCP Connection
2	SMB2	Negotiate
3	SMB2	Session Setup
4	SMB2	Tree Connect
5	SWT	Create Variable oneKB
6	SMB2	Create File
7	SWT	Create Variable twohundred
8	SWT	Begin Loop mainloop
9	SWT	Formula writeoffset
10	SMB2	File Write
11	SWT	End Loop
12	SWT	Formula totalbyteswritten

Name	Value
Input	
Alias	totalbyteswritten
Description	total bytes written
Formula	VAR(oneKB) * VAR(twohundred)
Output	
Output	12: Formula

These Aliases are used by the Write and Read Actions:

The screenshot shows a sequence of 17 protocol steps. Step 10, 'File Write', is highlighted with a green arrow icon. The configuration panel on the right displays the following properties for 'File Write':

Name	Value
Input	
File Handle	Default
Credits Charged	0
Credits Requested	1
Automatic Offset	False
File Offset	=@FORMULA(WriteOffset)
Bytes Per Block	=@VAR(oneKB)
Remaining Bytes to Write	0
Bytes Total	=@VARIABLE(oneKB)
Block Sequence	Forward
Data Content	
Data Source	::SeededRandom(0)
Data Source Offset	0
Output	
Status Code	
Response Handlers	
Completion Status	
Scenario Impact	

The screenshot shows the same sequence of 17 protocol steps. Step 10, 'File Write', is highlighted with a green arrow icon. The configuration panel on the right displays the following properties for 'File Write':

Name	Value
Input	
File Handle	Default
Credits Charged	0
Credits Requested	1
Automatic Offset	False
File Offset	0
Bytes Per Block	=@VARIABLE(oneKB)
Bytes Total	=@FORMULA(TotalBytesWritten)
Bytes Remaining	0
Bytes Padding	0
Block Sequence	Forward
Minimum Count	0
Data Verification	
Verification Data Source	
Verification Data Offset	0
Output	
Status Code	
Response Handlers	
Completion Status	

Producing a much more readable Scenario.

For more information on Formulas and Functions see [Appendix: Functions and Formula](#).

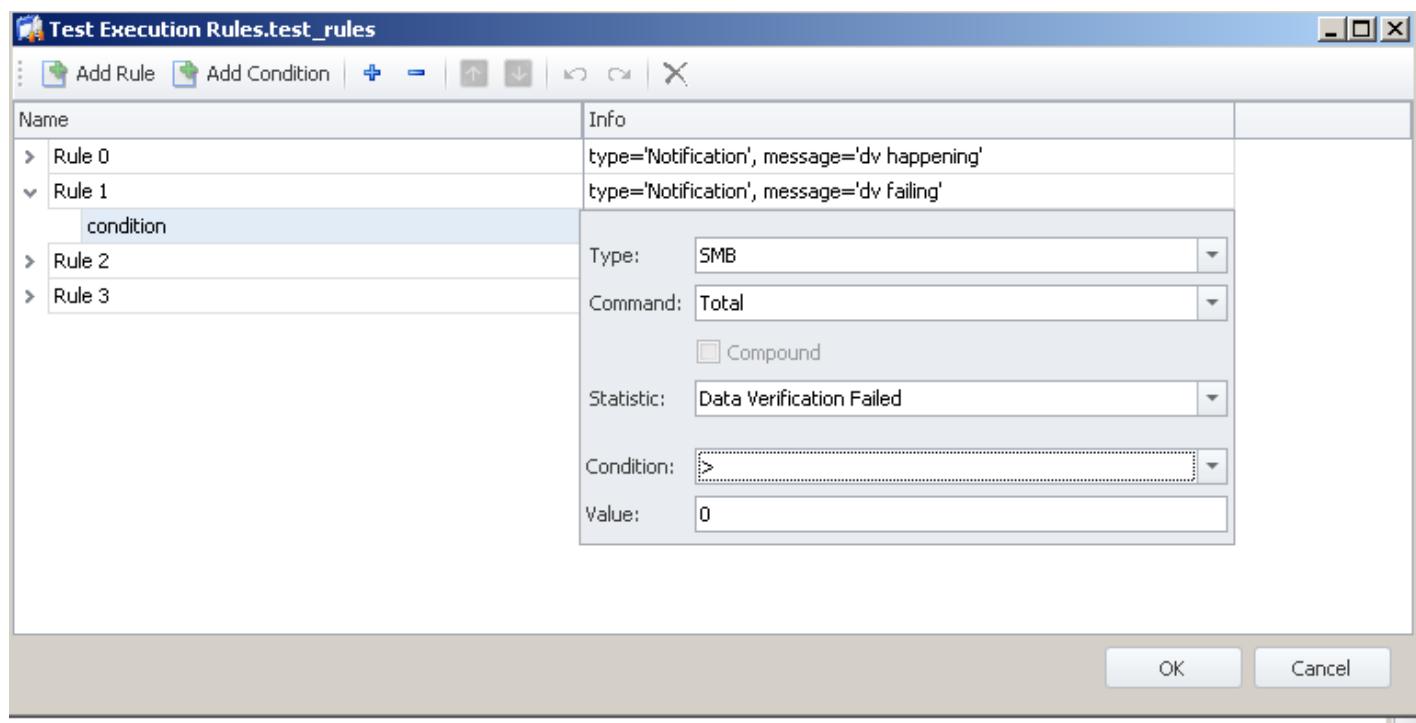
Advanced Concepts: Test Execution Rules

Advanced Concepts: Test Execution Rules

Test Execution Rules allow a Tester to specify conditions to be monitored during the execution of Scenarios and actions to be taken if/when a monitored condition is matched. If a condition of "Error" type is matched during the Scenario execution, the Project is immediately terminated, with a user-specified message. If a condition of "Warning" type or "Notification" type is matched during the Scenario execution, then Scenario execution continues after logging a user-specified message into the Output window (or to STDOUT if the test is being executed using LdxCmd.exe). Test Execution rules take effect by being associated with (dragged and dropped onto) a Timeline resource. Scenario execution via the TDE (see [Executing Tests and Assessing Results](#)) and Automation (see [Appendix: Test Automation](#)) monitor for condition matching in the same way.

Using Test Execution Rules

Test Execution Rules can be created by using Project -> Add New Item menu item or the Project Explorer window Add New item button . To add a new Rule, click the Add Rule button in the upper left corner of the Test Execution Rule property editor.



A Test Execution Rules component contains one or more "Rules". Each Rule is made up of:

- Rule Type (Error, Warning, Notification)
- User-defined message to be logged when the Rule's conditions are met
- A set of Conditions which are made up of
 - The Type of Statistic ((protocol or other statistics collection like Load, see below))
 - Command (scope of the Statistic - one command or Totals)
 - Statistic (which specific statistic to test)
 - A Condition (==, !=, <, <=, >, >=)
 - The Value to compare with

The compare Conditions are accessible via the drop down menu on the Condition field (see image above as an example).

See below for the Test Execution Rule Types and Statistics (counters or calculated measures) that can be evaluated.

Decide if the Rule is an Error (causes the Scenario to issue an Error message and exit) or a Warning (causes the Scenario to issue a Warning message and continue executing). Also decide what the message is going to be if Error or Warning (possibly the same message in either case).

Then select the Statistic that is to be monitored (e.g. SMB: Actions attempted) and what numeric condition to check for (e.g. \geq) and the numeric value (e.g. 100).

Rule Types are:

- **Error:** if any OR-condition in the rule is met, the Rule is matched, the Project execution is terminated with the user-defined **Error** message written to the Output Window and the Error count incremented.
- **Warning:** if any OR-condition in the rule is met, the Rule is matched, the Project execution continues after the user-defined **Warning** message is written to the Output Window and the Warning count incremented.
- **Notification:** if any OR-condition in the rule is met, the Rule is matched, the Test Project execution continues after writing the user-defined **Notification** message to the Output Window.

Rule Guidelines:

- Rules may occur in any order within a Test Execution Rule.
- Each Rule includes one or more OR-conditions. The Rule is considered matched if any of the OR-conditions are matched.
- Each OR-condition may be a stand-alone condition, or may include one or more AND-conditions.
- If OR-condition includes one or more AND-conditions, the OR-condition is considered met only if the OR-condition itself and all AND-conditions are met.
- AND-conditions may include OR-conditions.
- OR-condition and AND-condition include the following properties:
 - Statistics name to monitor the value of;
 - Statistics value to monitor for;
 - Condition to evaluate ("==", "!="), ">", "<", ">=" and "<=")
- Within the "Test Execution Rules" Editor, OR-conditions and AND-conditions are identified by indentation and the fact that:
 - All OR-conditions are siblings (same-level nodes) to each other in the conditions tree
 - All AND-conditions are children (lower-level nodes) of the OR-condition

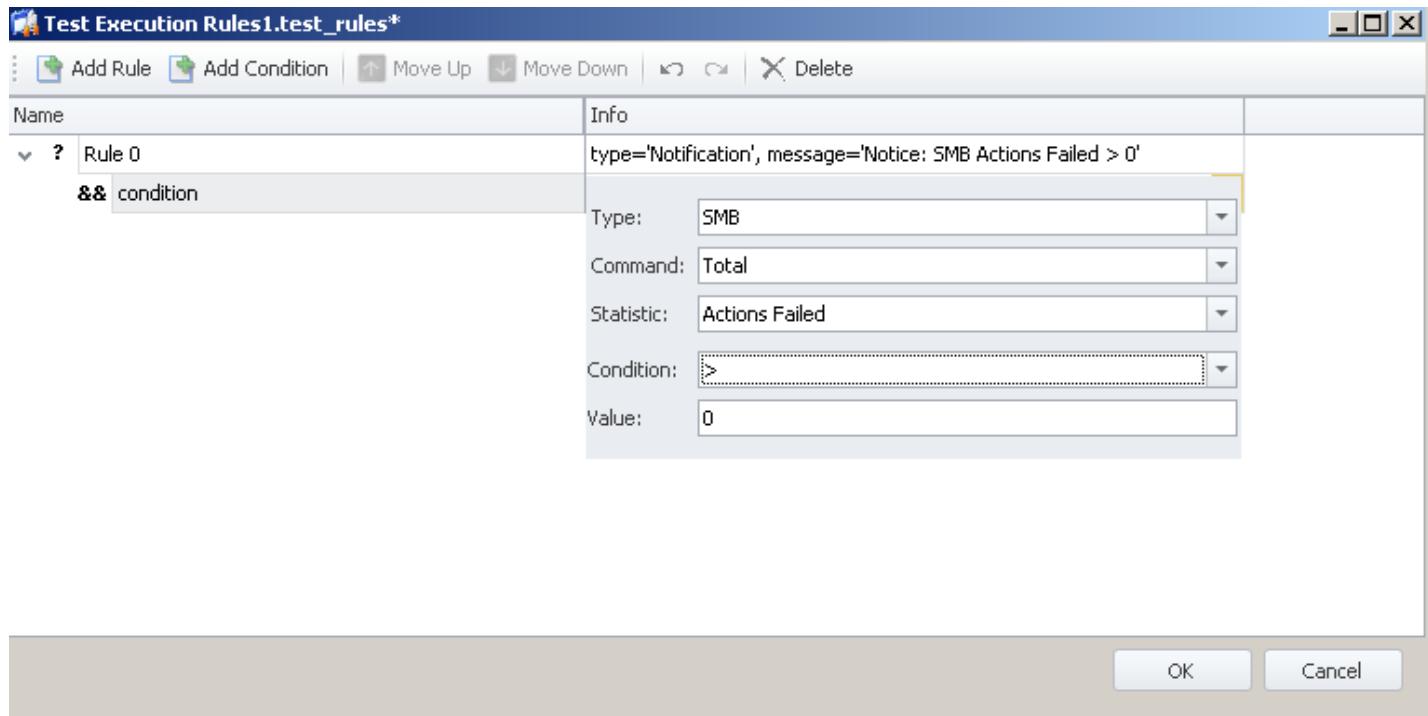
Sample Rule Logic:

```
-- AAA
-- BBB
-- CCC
-- DDD
-- EEE
-- FFF
```

The rule is matched if:

condition AAA is met by itself, or

condition BBB is met by itself, and
 condition CCC is met by itself, or
 condition DDD is met by itself, or
 condition EEE is met by itself, and
 condition FFF is met by itself



Once the Test Execution Rules have been defined, drag and drop the Test Execution Rule onto the desired Logical Port. Test Execution Rules on Network Profiles and Scenarios are not supported and allowed.

The above Test Execution Rules would check for Failed SMB Actions during Project execution , write a Notification message to the event log and continue executing the Project.

Test Execution Rules Types and Statistics

The following tables list the Statistics that are generated by the various Protocols that are used in Load DynamiX Projects. Some Statistics are generated on a per Action or Total Actions basis (e.g. SMB File Read Actions Attempted), some are generated on a Session or Connection basis (e.g. TCP Connections Opened Per Second, Timed Out Per Second), and some are generated based on the artifacts of a Load DynamiX Project (e.g. Scenarios Failed Per Second).

Load Status

Actions Attempted
Actions Succeeded
Actions Failed
Actions Aborted
Scenarios Attempted
Scenarios Succeeded
Scenarios Failed
Scenarios Aborted
Connections Attempted
Connections Succeeded

Connections Failed
Connections Aborted
Scenarios Attempted per Second
Scenarios Succeeded per Second
Scenarios Failed per Second
Scenarios Aborted per Second
Actions Attempted per Second
Actions Succeeded per Second
Actions Failed per Second
Actions Aborted per Second
Connections Attempted per Second
Connections Succeeded per Second
Connections Failed per Second
Connections Aborted per Second

Network Status

RX OK Packets
TX OK Packets
RX OK Bytes
TX OK Bytes
RX OK Packets per Second
TX OK Packets per Second
RX OK Bytes per Second
TX OK Bytes per Second
RX Broadcast Packets
TX Broadcast Packets
RX Multicast Packets
TX Multicast Packets
TX Failed Packets
TX Failed Bytes
RX Failed Packets
RX Failed Bytes
RX CRC/Symbol/Sequence/Carrier/Data Error Count
RX CRC/Symbol/Sequence/Carrier/Data Error Bytes
RX Out of Buffer Space Error Count
RX Fragmented Packet Error Count
RX Oversized Packet Error Count
RX Jabber Packet Error Count
RX Flow Control Error Count
RX Receive Length Invalid Error Count
RX Total Error Count
CRC Error Count
Alignment Error Count
Symbol Error Count
Missed Packet Error Count
Single Collisions Error Count
Multiple Collisions Error Count
Late Collisions Error Count
Total Collisions Error Count
Defer Count
Sequence Error Count
Carrier Extensions Error Count
RX Dropped Packets - stack throttled
RX Dropped Bytes - stack throttled
TX Dropped Packets - stack throttled
TX Dropped Bytes - stack throttled
Total Time of stack throttled
Maximum Count of Concurrent Packets
Current Bytes of User Memory Allocated
Maximum Bytes of User Memory Allocated

Total Time Process Out of User Memory
Maximum Number of Concurrent Scenarios
RX Undersized Packet Error Count
Excessive_Collisions_Error_Count

ARP

RX OK Request Packets
RX OK Request Bytes
RX OK Reply Packets
RX OK Reply Bytes
New Address Resolution Count
Overriding Address Resolution Count
Duplicated Address Resolution Count
RX Length Error Packets
RX Length Error Bytes
RX Dropped Packets
RX Dropped Bytes
RX Illegal Packets
RX Illegal Bytes
RX Trailer Bytes, cut off
TX Request Packets
TX Request Bytes
TX Gratuitous Packets
TX Gratuitous Bytes
TX Reply Packets
TX Reply Bytes
RX Packets - Requests Replied
RX Bytes - Requests Replied
RX Packets - Requests Ignored
RX Bytes - Requests Ignored

Ethernet

RX OK Packets
RX OK Payload Bytes
RX OK Ethernet Header Bytes
RX OK Vlan Header Bytes
TX OK Packets
TX OK Payload Bytes
TX Ethernet Header Bytes
TX Vlan Header Bytes
RX Type OK Packets - Supported Data Types
RX Type OK Bytes - Supported Data Types
RX Dropped Packets - Unsupported Data Types
RX Dropped Bytes - Unsupported Data Types
RX Length Error Packets
RX Length Error Bytes
TX Length Error Packets
TX Length Error Bytes

IPv4

RX OK Packets
RX OK Header Bytes
RX OK Data Bytes, Supported Protocols
RX OK Data Bytes, Unsupported Protocols
TX OK Packets
TX OK Header Bytes
TX OK Data Bytes
RX Header Length Error Packets
RX Header Length Error Bytes

RX Version Error Packets

RX Version Error Bytes

RX Data Length Error Packets

RX Data Length Error Bytes

RX Checksum Errors

IPv6

RX Checksum Errors

RX Data Length Errors Bytes

RX Data Length Errors Packets

RX Header Length Errors Bytes

RX Header Length Errors Packets

RX OK Data Bytes, Supported Protocol

RX OK Data Bytes, Unsupported Protocol
--

RX Version Error Bytes

RX Version Error Packets

TX Error Data Bytes

TX Error Data Packets

TX OK Data Bytes

TX OK Header Bytes

TX OK Packets

ICMP

Address Mask Reply

Address Mask Request

Alternate Host Address

Datagram Conversion Error

Destination Unreachable

Domain Name Reply

Domain Name Request

Echo Reply

Echo Request

Experimental Protocols

Information Reply

Information Request

Total

Traceroute

ICMPv6

Certification Path Advertisement Message
--

Certification Path Solicitation Message

Destination Unreachable

Duplicate Address Confirmation

Echo Reply

Echo Request

Experimental Protocols

FMIPv6 Messages

ICMP Node Information Query

ICMP Node Information Response

ILNPv6 Locator Update Message

Router Advertisement

Router Renumbering

Router Solicitation

Total

DCBX

DCBX TLV

CEE_APP_PRIO

CEE_PFC

CEE_PRIO_GROUP
IEEE_APP_PRIO
IEEE_ETS_CONN
IEEE_ETS_REC
IEEE_PFC
Total

Statistics

Resolved via Local
Resolved via Local Per Second
Resolved via Remote
Resolved via Remote Per Second
Rx TLV
Rx TLV Error Bit
Rx TLV Error Bit Per Second
Rx TLV Per Second
Terminated by Peer Change
Terminated by Peer Change Per Second
Terminated by TTL Timeout
Terminated by TTL Timeout Per Second
Tx TLV
Tx TLV Error Bit
Tx TLV Error Bit Per Second
Tx TLV Per Second

DNS Query

Aborted
Aborted Per Second
Attempted
Attempted Per Second
Empty Answer
Empty Answer Per Second
Failed
Failed Per Second
NXDOMAIN
NXDOMAIN Per Second
Other Aborted
Other Aborted Per Second
Other Failed
Other Failed Per Second
REFUSED
REFUSED Per Second
Response Time Average
Response Time Maximum
Response Time Minimum
Retries
Retries Per Second
Succeeded
Succeeded Per Second
Transport Reset
Transport Reset Per Second
Transport Timeout
Transport Timeout Per Second
Truncated
Truncated Per Second

DNS Resolution

Cache Entries Created
Cache Entries Created Per Second

Cache Hits
Cache Hits Per Second
Cache Misses
Cache Misses Per Second
Cache Negative Hits
Cache Negative Hits Per Second
Cache Negative Updates
Cache Negative Updates Per Second
Resolution Time Average
Resolution Time Maximum
Resolution Time Minimum
Resolutions Aborted
Resolutions Aborted Per Second
Resolutions Attempted
Resolutions Attempted Per Second
Resolutions Failed
Resolutions Failed Per Second
Resolutions Succeeded
Resolutions Succeeded Per Second

Priority Flow Control

Priority

PRIORITY_0
PRIORITY_1
PRIORITY_2
PRIORITY_3
PRIORITY_4
PRIORITY_5
PRIORITY_6
PRIORITY_7
Total

Statistics

RX PFC Pause OFF
RX PFC Pause OFF Per Second
RX PFC Pause ON
RX PFC Pause ON Per Second
TX PFC Pause OFF
TX PFC Pause OFF Per Second
TX PFC Pause ON
TX PFC Pause ON Per Second

TCP/TCPv6

RX Bytes Per Second
RX Checksum Errors
RX Data Length Errors Bytes
RX Data Length Errors Packets
RX Dropped Data Bytes
RX Dropped Header Bytes
RX Dropped Packets
RX Duplicate Bytes Per Second
RX Duplicate Data Bytes
RX Duplicate Header Bytes
RX Duplicate Packets
RX Duplicate Packets Per Second
RX Header Length Errors Bytes
RX Header Length Errors Packets
RX Invalid Address/Port Bytes Per Second
RX Invalid Address/Port Packets Per Second

RX Invalid Destination Bytes Per Second
RX Invalid Destination Data Bytes
RX Invalid Destination Header Bytes
RX Invalid Destination Packets
RX Invalid Destination Packets Per Second
RX Length Error Bytes Per Second
RX Length Error Packets Per Second
RX OK Bytes Per Second
RX OK Data Bytes
RX OK Header Bytes
RX OK Packets
RX OK Packets Per Second
RX Out Of Sequence Bytes Per Second
RX Out Of Sequence Data Bytes
RX Out Of Sequence Dropped Buckets
RX Out Of Sequence Dropped Packets
RX Out Of Sequence Enqueued Buckets
RX Out Of Sequence Enqueued Packets
RX Out Of Sequence Header Bytes
RX Out Of Sequence Packets
RX Out Of Sequence Packets Per Second
RX Out Of Sequence Recovered Buckets
RX Out Of Sequence Recovered Packets
RX Packets Per Second
RX Rejected Bytes Per Second
RX Rejected Data Bytes
RX Rejected Header Bytes
RX Rejected Packets
RX Rejected Packets Per Second
RX Throttled Bytes Per Second
RX Throttled Data Bytes
RX Throttled Header Bytes
RX Throttled Packets
RX Throttled Packets Per Second
TX Bytes Per Second
TX Canceled Data Bytes
TX Canceled Header Bytes
TX Canceled Packets
TX OK Bytes Per Second
TX OK Data Bytes
TX OK Header Bytes
TX OK Packets
TX OK Packets Per Second
TX Out Of Sequence Bytes Per Second
TX Out Of Sequence Packets Per Second
TX Packets Per Second
TX Retransmitted Bytes Per Second
TX Retransmitted Data Bytes
TX Retransmitted Header Bytes
TX Retransmitted Packets
TX Retransmitted Packets Per Second

TCP Connections

ARP/NDP Resolutions Attempted
ARP/NDP Resolutions Attempted Per Second
ARP/NDP Resolutions Failed
ARP/NDP Resolutions Failed Per Second
ARP/NDP Resolutions Succeeded
ARP/NDP Resolutions Succeeded Per Second
Attempted

Attempted Per Second
Average Connection Duration (microseconds)
Closed
Closed Per Second
Closing Time (microseconds)
Data Timeout
Data Timeout Per Second
DNS Resolutions Attempted
DNS Resolutions Attempted Per Second
DNS Resolutions Failed
DNS Resolutions Failed Per Second
DNS Resolutions Succeeded
DNS Resolutions Succeeded Per Second
Failed
Failed Per Second
Failed to Open
Failed to Open Per Second
Inactivity Timeout
Inactivity Timeout Per Second
Max Closing Time (microseconds)
Max Connection Time (microseconds)
Max Time-to-1st-byte Time (microseconds)
Min Closing Time (microseconds)
Min Connection Time (microseconds)
Min Time-to-1st-byte Time (microseconds)
Open Timeout
Open Timeout Per Second
Opened
Opened Per Second
Reset
Reset Per Second
Reset By Drop Due to Throttle
SYN Handshake Attempted
SYN Handshake Attempted Per Second
SYN Handshake Rejected
SYN Handshake Rejected Per Second
SYN Handshake Timeout
SYN Handshake Timeout Per Second
Time-to-1st-byte Time (microseconds)

TCP/TCPv6 State

Closed to Listen
Closed to Syn-tx'd
Closed to Syn-rx'd
Syn-rx'd to Established
Syn-tx'd to Syn-rx'd
Syn-tx'd to Established
Established to Fin Wait 1
Established to Fin Wait 2
Established to Closing
Fin Wait 1 to Fin Wait 2
Closing to Last Ack
Syn-tx'd to Closed
Syn-rx'd to Closed
Established to Closed
Fin-rx'd to Closed
Fin Wait 1 to Closed
Fin Wait 2 to Closed
Last Ack to Closed

TCP/TCPv6 Flags

RX OK RST
RX OK SYN
RX OK SYN/ACK
RX OK FIN/ACK
RX OK FIN/ACK/PUSH
RX OK ACK
RX OK ACK/PUSH
RX OK Invalid
RX Duplicate RST
RX Duplicate SYN
RX Duplicate SYN/ACK
RX Duplicate FIN/ACK
RX Duplicate FIN/ACK/PUSH
RX Duplicate ACK
RX Duplicate ACK/PUSH
RX Duplicate Invalid
RX Out of Sequence RST
RX Out of Sequence SYN
RX Out of Sequence SYN/ACK
RX Out of Sequence FIN/ACK
RX Out of Sequence FIN/ACK/PUSH
RX Out of Sequence ACK
RX Out of Sequence ACK/PUSH
RX Out of Sequence Invalid
TX OK RST
TX OK SYN
TX OK SYN/ACK
TX OK FIN/ACK
TX OK FIN/ACK/PUSH
TX OK ACK
TX OK ACK/PUSH
TX OK Invalid
TX Retransmit RST
TX Retransmit SYN
TX Retransmit SYN/ACK
TX Retransmit FIN/ACK
TX Retransmit FIN/ACK/PUSH
TX Retransmit ACK
TX Retransmit ACK/PUSH
TX Retransmit Invalid
RX Throttled RST
RX Throttled SYN
RX Throttled SYN/ACK
RX Throttled FIN/ACK
RX Throttled FIN/ACK/PUSH
RX Throttled ACK
RX Throttled ACK/PUSH
RX Throttled Invalid

TCP/TCPv6 Reconnects**TCP/TCPv6 Reconnect Commands**

Connection-level rounds
Reconnects due to address resolution timeout
Reconnects due to close by peer
Reconnects due to data timeout
Reconnects due to inactivity timeout
Reconnects due to open timeout
Reconnects due to reset by peer
Reconnects due to forced by action

Reconnects total

Statistics

Attempted
Succeeded
Failed
Aborted
Attempted Per Second
Succeeded Per Second
Failed Per Second
Aborted Per Second
OK Average Duration Excluding Delay (microseconds)
OK Minimum Time Excluding Delay (microseconds)
OK Maximum Time Excluding Delay (microseconds)
OK Average Duration Including Delay (microseconds)
OK Minimum Time Including Delay (microseconds)
OK Maximum Time Including Delay (microseconds)
Fail Average Duration Excluding Delay (microseconds)
Fail Minimum Time Excluding Delay (microseconds)
Fail Maximum Time Excluding Delay (microseconds)
Fail Average Duration Including Delay (microseconds)
Fail Minimum Time Including Delay (microseconds)
Fail Maximum Time Including Delay (microseconds)
Abort Average Duration Excluding Delay (microseconds)
Abort Minimum Time Excluding Delay (microseconds)
Abort Maximum Time Excluding Delay (microseconds)
Abort Average Duration Including Delay (microseconds)
Abort Minimum Time Including Delay (microseconds)
Abort Maximum Time Including Delay (microseconds)

SSL/TLS

RX Alert Bytes
RX Alert Bytes Per Second
RX Alert Packets
RX Alert Packets Per Second
RX Data Bytes
RX Data Bytes Per Second
RX Data Packets
RX Data Packets Per Second
RX Handshake Bytes
RX Handshake Bytes Per Second
RX Handshake
RX Handshake Packets Per Second
TX Alert Bytes
TX Alert Bytes Per Second
TX Alert Packets
TX Alert Packets Per Second
TX Data Bytes
TX Data Bytes Per Second
TX Data Packets
TX Data Packets Per Second
TX Handshake Bytes
TX Handshake Bytes Per Second
TX Handshake Packets
TX Handshake Packets Per Second

SSL/TLS Connections

Attempted
Attempted Per Second
Closed

Closed Per Second
Closing Time (microseconds)
Connection Average Duration Time (microseconds)
Established
Established Per Second
Handshake Aborted
Handshake Aborted Per Second
Handshake Failed
Handshake Failed Per Second
Max Closing Time (microseconds)
Max Connection Time (microseconds)
Max Time-to-open Time (microseconds)
Min Closing Time (microseconds)
Min Connection Time (microseconds)
Min Time-to-open Time (microseconds)
Rejected
Rejected Per Second
Reset Per Second
Reset
Time-to-open (microseconds)

HTTP Authentications

HTTP Authentication Commands

AMAZON
AWS2
AWS4
BASIC
DIGEST
NEGOTIATION_KERBEROS
NEGOTIATION_KERBEROS_NTLM
NEGOTIATION_NTLM
NEGOTIATION_NTLM_KERBEROS
NTLM_NTLM

Statistics

Access Forbidden
Access Forbidden Per Second
Authentication Aborted
Authentication Aborted Per Second
Authentication Attempted Per Second
Authentication Attempted
Authentication Attempted Passive
Authentication Attempted Passive Per Second
Authentication Attempted Preemptive
Authentication Attempted Preemptive Per Second
Authentication Failed
Authentication Failed Per Second
Authentication Ignored
Authentication Ignored Per Second
Disabled/Invalid Scheme Per Second
Maximum Time (microseconds)
Minimum Time (microseconds)
Scheme Reset By Server
Scheme Reset By Server Per Second
Server Error
Sever Error Per Second
Abort Maximum Time Including Delay (microseconds)

HTTP Encoding

Statistics

Chunked Rx HTTP Entity Bytes
Chunked Rx HTTP Entity Bytes Per Second
Chunked Rx HTTP Message Bytes
Chunked Rx HTTP Message Bytes Per Second
Chunked Rx Messages Aborted
Chunked Rx Messages Aborted Per Second
Chunked Rx Messages Attempted
Chunked Rx Messages Attempted Per Second
Chunked Rx Messages Failed
Chunked Rx Messages Failed Per Second
Chunked Rx Messages Succeeded
Chunked Rx Messages Succeeded Per Second
Chunked Total HTTP Entity Bytes
Chunked Total HTTP Entity Bytes Per Second
Chunked Total HTTP Message Bytes
Chunked Total HTTP Message Bytes Per Second
Chunked Tx HTTP Entity Bytes
Chunked Tx HTTP Entity Bytes Per Second
Deflate Rx Decoded Bytes
Deflate Rx Decoded Bytes Per Second
Deflate Rx Encoded Bytes
Deflate Rx Encoded Bytes Per Second

HTTP Response Parsing

Statistics

Body Entities Parsing Aborted
Body Entities Parsing Aborted Per Second
Body Entities Parsing Attempted
Body Entities Parsing Attempted Per Second
Body Entities Parsing Failed
Body Entities Parsing Failed Per Second
Body Entities Parsing Succeeded
Body Entities Parsing Succeeded Per Second
Body Values Extraction Aborted
Body Values Extraction Aborted Per Second
Body Values Extraction Attempted
Body Values Extraction Attempted Per Second
Body Values Extraction Succeeded
Body Values Extraction Succeeded Per Second
Header Entities Parsing Aborted
Header Entities Parsing Aborted Per Second
Header Entities Parsing Attempted
Header Entities Parsing Attempted Per Second
Header Values Extraction Aborted
Header Entities Extraction Aborted Per Second
Header Entities Extraction Attempted
Header Entities Extraction Attempted Per Second

HTTP/HTTPS Redirect

Statistics

HTTP Redirects w/o Reconnects Aborted
HTTP Redirects w/o Reconnects Aborted Per Second
HTTP Redirects w/o Reconnects Attempted
HTTP Redirects w/o Reconnects Attempted Per Second
HTTP Redirects w/o Reconnects Failed
HTTP Redirects w/o Reconnects Failed Per Second
HTTP Redirects w/o Reconnects Succeeded
HTTP Redirects w/o Reconnects Succeeded Per Second
HTTP Redirects with Reconnects Aborted
HTTP Redirects with Reconnects Aborted Per Second

HTTP Redirects with Reconnects Attempted
--

HTTP Redirects with Reconnects Attempted Per Second

HTTP Redirects with Reconnects Failed

HTTP Redirects with Reconnects Failed Per Second
--

HTTP Redirects with Reconnects Succeeded
--

HTTP Redirects with Reconnects Succeeded Per Second

Redirects Total Aborted

Redirects Total Aborted Per Second

Redirects Total Attempted

Redirects Total Attempted Per Second

Redirects Total Succeeded

Redirects Total Succeeded Per Second

FC Network Status

FC RX DISCARDS

FC RX OK BYTES

FC RX OK FRAMES

FC RX OK FRAMES PER SECOND

FC TX DISCARDS

FC TX OK BYTES

FC TX OK FRAMES

FC TX OK FRAMES PER SECOND

RX OK BYTES PER SECOND

TX OK BYTES PER SECOND

FCoE Network Status

FCoE RX DISCARDS

FCoE RX OK BYTE

FCoE RX OK FRAME

FCoE RX OK FRAMES PER SECOND

FCoE TX DISCARDS

FCoE TX OK BYTES

FCoE TX OK FRAMES

FCoE TX OK FRAMES PER SECOND

RX OK BYTES PER SECOND

TX OK BYTES PER SECOND

FCR

ABORTED

ABORTED PER SECOND

ATTEMPTED

ATTEMPTED PER SECOND

FAILED

FAILED PER SECOND

SUCCEEDED

SUCCEEDED PER SECOND

FC/iSCSI MPIO

ADDITIONAL TRANSITIONS

ADDITIONAL TRANSITIONS PER SECOND

FAIL BACKS

FAIL BACKS PER SECOND

FAIL OVER

FAIL OVER PER SECOND

RX BYTES

RX BYTES PER SECOND

RX PACKETS

RX PACKETS PER SECOND

TX BYTES

TX BYTES PER SECOND

TX PACKETS
TX PACKETS PER SECOND

FC Sessions

ABORTED
ABORTED PER SECOND
ATTEMPTED
ATTEMPTED PER SECOND
CLOSED
CLOSED PER SECOND
CONNECTION AVERAGE DURATION TIME (ms)
FAILED
FAILED PER SECOND
MAXIMUM CONNECTION TIME (ms)
MINIMUM CONNECTION TIME (ms)
SUCCEEDED
SUCCEEDED PER SECOND
TIMED-OUT
TIMED-OUT PER SECOND

UDP

RX Bytes Per Second
RX Checksum Errors
RX Data Length Errors Bytes
RX Data Length Errors Packets
RX Dropped Bytes Per Second
RX Dropped Data Bytes
RX Dropped Header Bytes
RX Dropped Packets
RX Dropped Packets Per Second
RX Header Length Errors Bytes
RX Header Length Errors Packets
RX Invalid Destination Bytes Per Second
RX Invalid Destination Data Bytes
RX Invalid Destination Header Bytes
RX Invalid Destination Packets
RX Invalid Destination Packets Per Second
RX Length Error Bytes Per Second
RX Length Error Packets Per Second
RX OK Bytes Per Second
RX OK Data Bytes
RX OK Header Bytes
RX OK Packets
RX OK Packets Per Second
RX Packets Per Second
RX Rejected Bytes Per Second
RX Rejected Data Bytes
RX Rejected Header Bytes
RX Rejected Packets
RX Rejected Packets Per Second
RX Throttled Bytes Per Second
RX Throttled Data Bytes
RX Throttled Header Bytes
RX Throttled Packets
RX Throttled Packets Per Second
TX Bytes Per Second
TX Canceled Data Bytes
TX Canceled Header Bytes
TX Canceled Packets
TX OK Bytes Per Second

TX OK Data Bytes
TX OK Header Bytes
TX OK Packets
TX OK Packets Per Second
TX Packets Per Second

UDP Transports

ARP/NDP Resolutions Attempted
ARP/NDP Resolutions Attempted Per Second
ARP/NDP Resolutions Failed
ARP/NDP Resolutions Failed Per Second
ARP/NDP Resolutions Succeeded
ARP/NDP Resolutions Succeeded Per Second
Average Transport Timespan (microseconds)
Data Timeout
Data Timeout Per Second
DNS Resolutions Attempted
DNS Resolutions Attempted Per Second
DNS Resolutions Failed
DNS Resolutions Failed Per Second
DNS Resolutions Succeeded
DNS Resolutions Succeeded Per Second
Inactivity Timeout
Inactivity Timeout Per Second
Max Connection Time (microseconds)
Min Connection Time (microseconds)
Rejected (invalid destination)
Rejected (invalid destination) Per Second
Rejected by Peer
Rejected by Peer Per Second
Reset
Reset Due to Throttle
Reset Due to Throttle Per Second
Reset Per Second
Transports Attempted
Transports Attempted Per Second
Transports Closed
Transports Closed Per Second
Transports Failed
Transports Failed Per Second
Transports Open Failed
Transports Open Failed Per Second
Transports Opened
Transports Opened Per Second

Action-Oriented Protocol Statistics

These Statistics below are generated during the execution of Action for Protocols such as CIFS-SMB, SMB2, KERBEROS RPC, SCSI, CDMI, OPENSTACK SWIFT, OPENSTACK CINDER, AMAZON S3, and HTTP/HTTPS.

See [Reference: CIFS-SMB Commands and Behaviors](#) for a list of the CIFS-SMB Actions (commands).

See [Reference: SMB2 Commands and Behaviors](#) for a list of the SMB2 Actions (commands).

See [Reference: NFSv2 Command List](#) for a list of the NFSv2 Actions (commands).

See [Reference: NFSv3 Commands and Behaviors](#) for a list of the NFSv3 Actions (commands).

See [Reference: NFSv4, v4.1 Command List](#) for a list of the NFSv4/v4.1 Actions (commands).

See [Reference: Kerberos v5 Command List](#) for a list of the Kerberos Actions (commands).

See [Reference: iSCSI Commands and Behaviors](#) for a list of the iSCSI Actions (commands).

See [Reference: FC/SCSI/iSCSI Commands and Behaviors](#) for a list of the Fibre Channel, SCSI and iSCSI Actions (commands).

See [Reference: HTTP/HTTPS Commands and Behaviors](#) for a list of the HTTP/HTTPS Actions (commands).

See [Reference: HTTP Storage Commands and Behaviors](#) for a list of the HTTP Storage (CDMI, Amazon S3, OpenStack Swift, Cinder, etc) Actions (commands).

Caveat:

Some of the statistics listed below do not apply to every Protocol.

- For example, Signing-related statistics do not apply to any Protocol that does not support packet signing (e.g. SCSI, HTTP, RPC, Amazon S3, etc)
- Statistics that do not apply for a particular protocol return <Null> and will never be triggered in a Test Execution Rule.

RX Bytes
RX Packets
TX Bytes
TX Packets
RX Bytes Per Second
RX Bytes Per Second
RX Packets Per Second
TX Bytes Per Second
TX Packets Per Second
Actions Attempted
Actions Succeeded
Actions Failed
Actions Aborted
Actions Attempted Per Second
Actions Succeeded Per Second
Actions Failed Per Second
Actions Aborted Per Second
Actions Retried
Actions Retried Per Second
Average Response/Latency Time (microseconds)
Minimum Response/Latency Time (microseconds)
Maximum Response/Latency Time (microseconds)
RX OK Bytes
RX OK Packets
TX OK Bytes
TX OK Packets
RX Failed Bytes
RX Failed Packets
TX Failed Bytes
TX Failed Packets
RX Dropped Bytes
RX Dropped Packets
TX Dropped Bytes
TX Dropped Packets
ResultOK Attempted
ResultOK Succeeded
ResultOK Failed
ResultOK Aborted
ResultOK Attempted Per Second
ResultOK Succeeded Per Second
ResultOK Failed Per Second
ResultOK Aborted Per Second
Data Verification Attempted
Data Verification Succeeded
Data Verification Failed

Data Verification Attempted Per Second
Data Verification Succeeded Per Second
Data Verification Failed Per Second
RX OK Bytes Per Second
RX OK Packets Per Second
TX OK Bytes Per Second
TX OK Packets Per Second
RX Failed Bytes Per Second
RX Failed Packets Per Second
TX Failed Bytes Per Second
TX Failed Packets Per Second
RX Dropped Bytes Per Second
RX Dropped Packets Per Second
TX Dropped Bytes Per Second
TX Dropped Packets Per Second
RX Signing Attempted
RX Signing Succeeded
RX Signing Failed
TX Signing Attempted
TX Signing Succeeded
TX Signing Failed
RX Signing Attempted Per Second
RX Signing Succeeded Per Second
RX Signing Failed Per Second
TX Signing Attempted Per Second
TX Signing Succeeded Per Second
TX Signing Failed Per Second
Total Errors

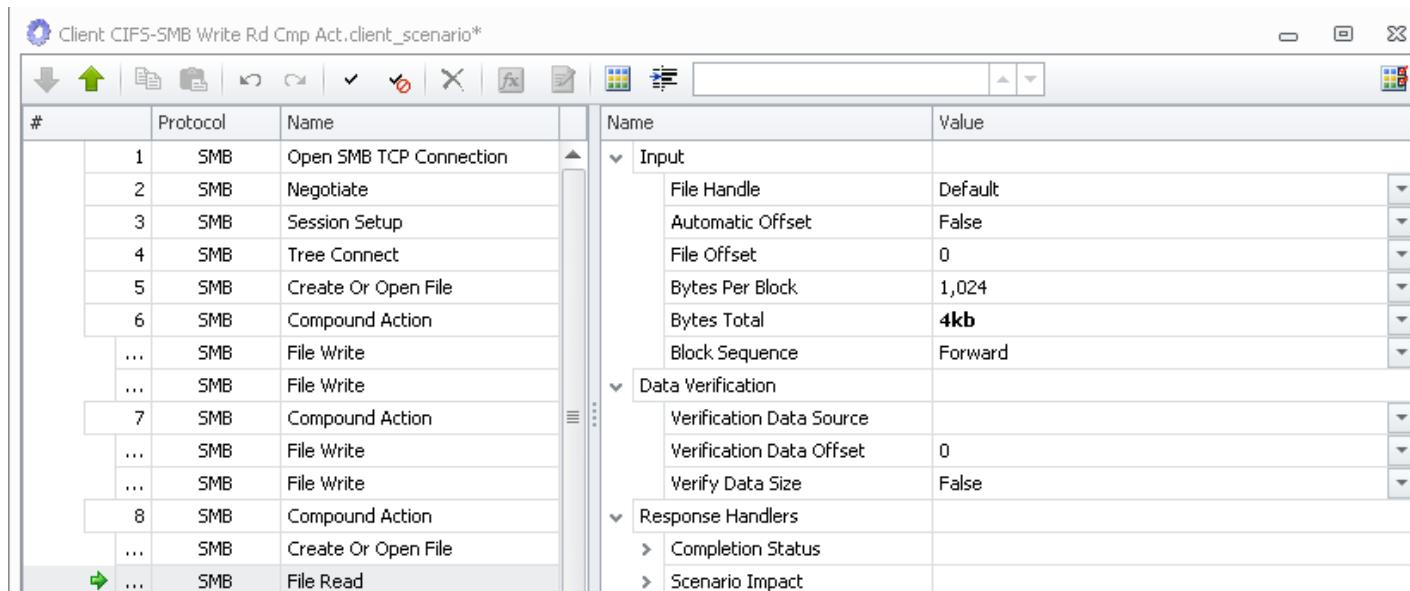
Advanced Concepts: Chained Commands

Advanced Concepts: Chained Commands

The CIFS-SMB and SMB2 Protocols allow multiple commands to be sent to a server in one or more consecutive packets. In the CIFS-SMB Protocol this is called Chaining Commands. In the SMB2 Protocol this is called Compound Requests. Load DynamiX on-line help will refer to them collectively as Compound Actions. In either case, Compound Actions are a means of improving client-server performance by reducing the network overhead of individual command handshake by the client and server. The CIFS-SMB protocol is more constrained regarding which CIFS-SMB commands may be chained together but the mechanism to use them in a Scenario is the same for both:

- Drag a Compound Action Action into the CIFS-SMB or SMB2 Scenario from the CIFS-SMB or SMB2 Toolbox
- Drag Actions from the CIFS-SMB or SMB2 Toolbox onto the Compound Action

The following screenshot shows a CIFS-SMB Scenario that contains multiple Compound Actions. The first Compound Action writes the 2K bytes of data to the file created in line 7 using two **File Write** Actions. The second Compound Action writes the next 2K bytes of data to the file created in line 7 using two **File Write** Actions. The third Compound Action reads back the 4K bytes first opening the file and then using one **File Read** Action. The Scenario Editor indicates which Actions are part of a Compound Action by using "..." in the line number column instead of a line number.



When this Scenario is executed, the five CIFS-SMB commands that are defined to be part of a Compound Action are packaged into 2 Chained Commands and sent to the Server in the order specified. Each Chained Command is executed in its entirety on the Server before any results are returned to the Client.

Compound Action Responses

When a Compound Action is sent to a server, the Client Scenario that sent the Compound Action will wait for the server to return a response to all of the actions in that Compound Action before continuing to the next Action in the Scenario.

CIFS-SMB: Only one response code (error code) is returned for the entire set of Actions sent in the

Compound Action.

SMB2: Each Action sent as part of a Compound Action will get its own response code.

For example, the following two PCAP segments show a CIFS-SMB Compound Action request and response

Compound Request containing two CIFS-SMB File Read Actions and one Trans2 Get File Information Action

```

[+] NetBIOS Session Service
[+] SMB (Server Message Block Protocol)
  [+ SMB Header
  [+ Read AndX Request (0x2e)
  [+ Read AndX Request (0x2e)
  [+ Trans2 Request (0x32)

```

Response from the Server for the above

```

NetBIOS Session Service
SMB (Server Message Block Protocol)
[-] SMB Header
  Server Component: SMB
  [Response to: 30]
  [Time from request: 0.000194000 seconds]
  SMB Command: Read AndX (0x2e)
  NT Status: STATUS_SUCCESS (0x00000000)
  [+ Flags: 0x98
  [+ Flags2: 0xc043
    Process ID High: 0
    Signature: 0000000000000000
    Reserved: 0000
  [+ Tree ID: 1 (\\\172.16.244.1\\)
    Process ID: 2
    User ID: 2048
    Multiplex ID: 6
  [+ Read AndX Response (0x2e)
  [+ Read AndX Response (0x2e)
  [+ Trans2 Response (0x32)

```

Note the single SMB Header portion of the response which has a single STATUS_SUCCESS for all three Actions. If for some reason, any one of the three Actions had failed, the status for that failure would be shown in the NT Status field.

Compound Action Inputs

CIFS-SMB

The input to the CIFS-SMB Compound Action is:

- Unicode (Enabled or Disabled)

Unicode = Enabled means that Unicode strings will be enabled for all of the Actions contained in this Compound Action. Unicode = Disabled means the opposite, Unicode strings are disabled for all the Actions in this Compound Action. Several CIFS-SMB Actions contain an input field that lets the Tester specify whether Unicode strings are to be used or not (e.g. Negotiate, Session Setup, Tree Connect, Create or Open File, ...). When these Actions are dragged into a

Compound Action, the input field used to specify Unicode or not is disabled and these commands will get their Unicode setting from the Compound Action input field.

SMB2

The input to the SMB2 Compound Action is:

Related Operations

the Actions in a CIFS-SMB Compound Action must all be related (i.e. the Actions inside a CIFS-SMB Compound Action must all operate on the same open file, share or server connection). SMB2 Compound Actions are allowed to have Actions that are not related (i.e. are not operating on the same open file, share or server connection) but the Related .vs. Non-Related nature of the Actions must be specified by the Tester. There is an input to the SMB2 Compound Action called Related Operation. That input set to True (the default), means that all Actions in the SMB2 Compound Action are Related. False means that they are not Related.

Caveats

RESPONSE HANDLERS

Compound Actions have Response Handlers with a limited selection of Completion Status values that can be acted upon. Only STATUS_SUCCESS and ERROR_OTHER can be detected and so Scenario Impact decisions must be made solely on those two choices.

CIFS-SMB Unicode Strings

CIFS-SMB Compound Actions have a single CIFS-SMB Header which contains the Unicode setting for the entire CIFS-SMB Compound Action. It is not possible to have some Actions with Unicode Enabled and some without in the same Compound Action.

SCENARIO CONTROL ACTIONS

All of the Load DynamiX Scenario Control Actions (e.g. Begin - End Loop, Advance User Parameters, Create Variable) are not allowed in Compound Actions. Dragging a Scenario Control Action into a CIFS-SMB Compound Action or SMB2 Compound Action will cause the Compound Action to be flagged as an error.

ASYNCHRONOUS ACTIONS (e.g. Change Notify, etc)

CIFS-SMB NT Transact Notify Change is not allowed in CIFS-SMB Chained Commands. In SMB2 Projects the Async and Sync Change Notify behavior is supported.

AUTO OFFSET

The Auto Offset feature of CIFS-SMB and SMB2 File Write and File Read Actions is supported in Compound Actions.

TEST EXECUTION RULES

Test Execution Rules for Actions embedded in Compound Actions will apply to the aggregate of all Actions of that type. For example, if a Project has two Compound Actions and each Compound Action contains an Open Action, it will be possible to create Test Execution Rules for all the Open Actions in Compound Actions but it will not be possible to set Test Execution Rules on the Open Actions separately.

PROTOCOLS

Only one Protocol (either CIFS-SMB or SMB2) is allowed within a single Compound Action. Dragging an SMB2 Action into a CIFS-SMB Compound Action or dragging an CIFS-SMB Action into a SMB2 Compound Action will cause the Compound Action to be flagged as an error.

DATA VERIFICATION

Data Verification will operate as expected inside of a Compound Action. Data written to a file inside a Compound Action may be read back and verified by Read Actions inside or outside of a Compound Action.

USER PARAMETERS & VARIABLES

User Parameters and Variables may be referred to within input fields of Actions inside of a Compound Action but the Actions that Advance or Reset User Parameter files, or Create or Update Variables are not allowed within Compound Actions.

DISALLOWED ACTIONS

The Load DynamiX TDE has no limit on the number of CIFS-SMB or SMB2 actions that can be dragged into a CIFS-SMB Compound Action or an SMB2 Compound Action. However, not all CIFS-SMB and SMB2 Actions that the TDE allows to be dragged into a Compound Action will be executed by a CIFS-SMB or SMB2 server as part of a Compound Action. The actions dragged into Compound Actions will be executed as a series of Compound and non-Compound requests to the target device based on CIFS-SMB and SMB2 protocol requirements. The TDE specifically allows most Actions to be dragged into Compound Actions to allow for negative testing scenarios.

CIFS-SMB and SMB2 Session Setup Actions should not be in Compound Actions if Packet Signing, Kerberos and NTLM Extended Security is being used.

SERVER BEHAVIORS: WINDOWS .vs. Load DynamiX .vs. Others

Not all CIFS-SMB and SMB2 Servers respond to Compound Actions in the same way. For example, the CIFS-SMB Chained Command specification says that multiple File Read commands are NOT allowed in the same Chained Command. If more than one File Read Action is executed in a Compound Action the response from the CIFS_SMB server will be very Server-specific. A Microsoft Windows 2003 Server might respond with an error code. A Load DynamiX SMB server would accept the multiple Read File Actions. Another CIFS-SMB server (e.g. a SAMBA-based server) might have a different response all together.

Advanced Concepts: Threads and Async Operations

Advanced Concepts: Threads and Asynchronous Operations

THREADS

The Actions in Load DynamiX Scenarios execute sequentially until completion which generally means that the Action in Row N executes and completes before the Action in Row N+1 is executed. If it is necessary, collections of Actions can be executed in parallel in Sub-Scenarios. The Begin Thread...End Thread Actions allow the Tester to specify which Actions are executed as a Sub-Scenario.

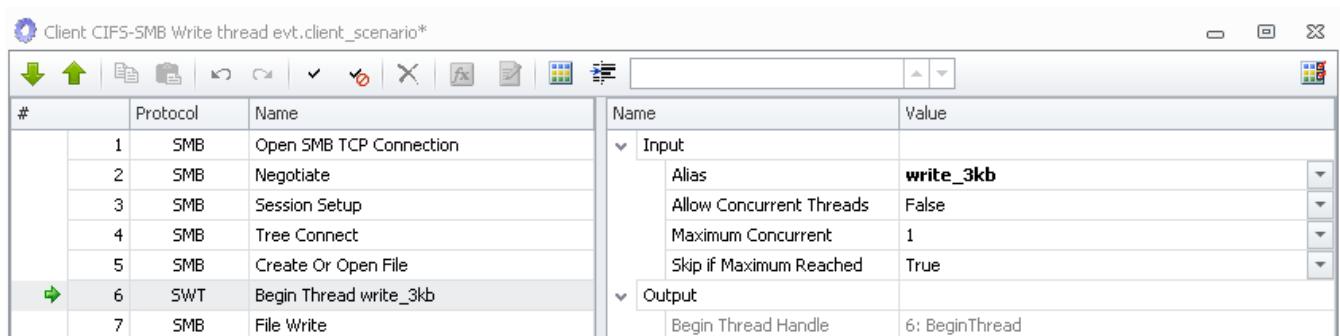
Conceptually, a Thread is a separate process that executes independently of the rest of the Scenario. The Scenario below is composed of two Threads, one that writes 3KB data to a file and the second that reads it back, 1KB at a time. Events are used to synchronize between the two Threads so that the Writes complete before the Reads start.

#	Protocol	Name
1	SMB	Open SMB TCP Connection
2	SMB	Negotiate
3	SMB	Session Setup
4	SMB	Tree Connect
5	SMB	Create Or Open File
6	SWT	Begin Thread write_3kb
7	SMB	File Write
8	SMB	File Write
9	SMB	File Write
10	SWT	Raise Event
11	SWT	End Thread
12	SWT	Wait for Event
13	SWT	Begin Loop
14	SWT	Begin Thread read_1kb_in_a_loop
15	SMB	File Read
16	SWT	End Thread
17	SWT	End Loop
18	SWT	Wait For All Threads
19	SMB	File Close

To wait for all Threads in a Scenario to complete, use the Wait For All Threads Action. In the Scenario above, there are two Threads. The two Threads each write 3KB to a file. The Wait For All Threads Action in line 18 will wait for all threads to complete before it will complete. If there are no Threads in the Scenario the Wait For All Threads Action is in effect a no-op.

Concurrent Thread Execution

The default settings of Thread Controls keep Threads from executing concurrently. To allow a Thread to execute more than once, it must be executed in a Begin Loop - End Loop block and the Thread Control settings must be set to non-default settings. Thread Control default settings are:

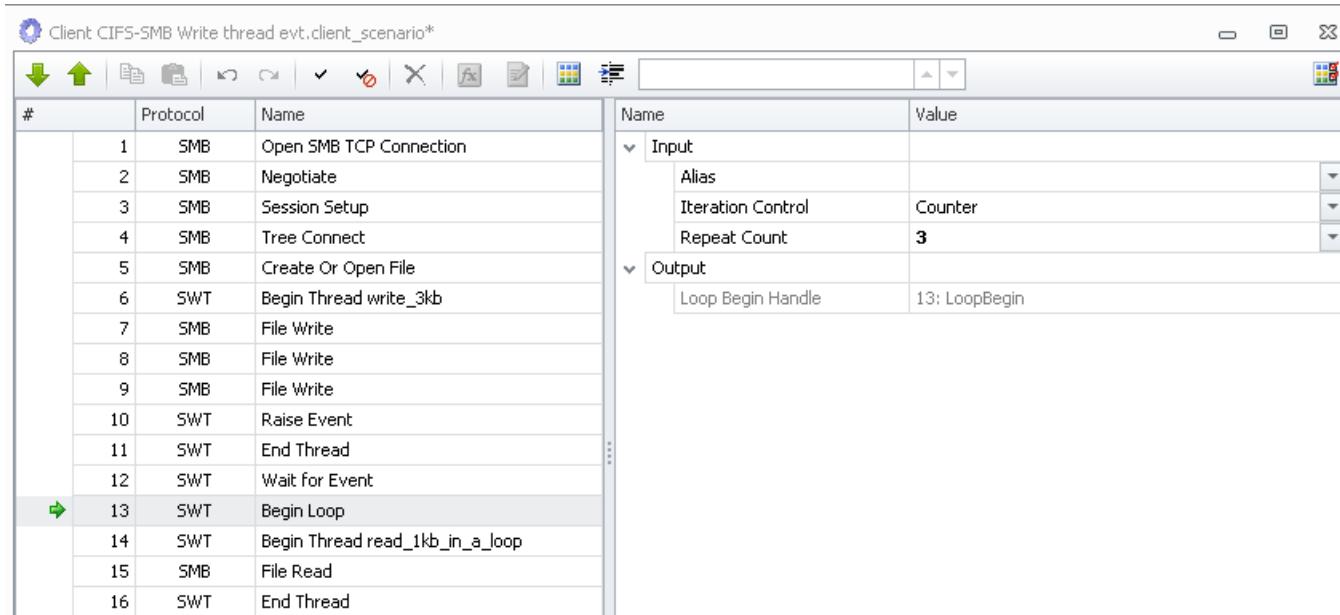


To allow a Thread to execute more than once at a time, put the Thread in a Begin Loop - End Loop block, set the Loop counter to the maximum number of Threads that are to be executed and set the Thread Controls to:

Allow Concurrent Threads = **TRUE** (to allow concurrent execution)

Maximum Concurrent = to the maximum number (**1...N**) of Threads to be executing at any particular time

Skip if Maximum Reached = **FALSE** (to prevent the loop from completing before all Threads have been executed)



In the example Project above, the Loop counter is 3 and the Thread Controls are set to:

In this Scenario, the Loop will get executed 3 times and during those 3 iterations, 3 Threads will be kept executing concurrently until the Loop counter expires. If the goal is for a specific number of Threads to be executed at the same time and no more, then set the Loop counter == Maximum Concurrent.

Nested Threads

Threads may be nested (Threads executed inside of Threads). In the example Scenario, below, the Thread that begins at line 6 contains a Begin Loop - End Loop block that contains another Thread (begins are line 8). If the Loop counter that begins at line 7 is 100 then the following behavior should be expected:

1 instance of the Thread beginning at line 6 will be executed. That Thread will spawn 100 Threads each containing three **File Write** Actions. So a total of 100 instances of the Thread that begins at line 8 will be executed and not until all have completed (due to the Wait for All Threads Action in line 15) will the Loop containing three **File Read** Actions be executed.

#	Protocol	Name
1	SMB	Open SMB TCP Connection
2	SMB	Negotiate
3	SMB	Session Setup
4	SMB	Tree Connect
5	SMB	Create Or Open File
6	SWT	Begin Thread main_thread_1
7	SWT	Begin Loop
8	SWT	Begin Thread nested_thread_100
9	SMB	File Write
10	SMB	File Write
11	SMB	File Write
12	SWT	End Thread
13	SWT	End Loop
14	SWT	End Thread
15	SWT	Wait For All Threads
16	SWT	Begin Loop
17	SMB	File Read
18	SMB	File Read
19	SMB	File Read
20	SWT	End Loop

Protocols Supporting Threads

The following Protocols support Threads:

- CIFS-SMB
- SMB2
- iSCSI (SCSI)
- Fibre Channel (SCSI)
- NFSv4.1
- HTTP/HTTPS
- HTTP Storage (CDMI, OpenStack Cinder, OpenStack Swift, Amazon S3)

ASYNCHRONOUS OPERATIONS

In the Scenario above, the **File Write** and **File Read** Actions are executed in their own Sub-Scenarios but they are still executed sequentially within each Sub-Scenario. If the Tester needs to have individual Actions executed in Parallel, then Asynchronous Operations can be used. To identify a set of Actions that are to be executed in parallel, use the Begin Async...End Async Actions around the SCSI, CIFS-SMB or SMB2, NFSv4.1, HTTP/HTTPS, HTTP Storage Actions that are to be executed in Parallel. Begin Async has only one input value - Wait For Completion with a default value of True which means that the Async collection of Actions will wait until all of the Actions have completed before the next Action is executed. If Wait For Completion is set to False then the Actions in the collection will be launched and Scenario execution will continue. In the Scenario below, the first Thread does all of the writing and the second Thread does all of the reading. Events are used to synchronize execution between the two threads and between the second Thread and the Tree Disconnect (we do not want to disconnect from the Share before the Reads are done). Each Thread contains a pair of **File Write** or **File Read** Actions in an Async block. The File Write and File Read Actions will be executed Asynchronously with the Thread.

#	Protocol	Name
1	SMB	Open SMB TCP Connection
2	SMB	Negotiate
3	SMB	Session Setup
4	SMB	Tree Connect
5	SWT	Begin Thread
6	#	Create file then All writes in parallel, wait for all to co...
7	SMB	Create Or Open File
8	SWT	Begin Async
9	SMB	File Write
10	SMB	File Write
11	SWT	End Async
12	SMB	File Close
13	SWT	Raise Event
14	SWT	End Thread
15	#	Next Thread reopens the previous file
16	SWT	Wait for Event
17	SWT	Begin Thread
18	SMB	Create Or Open File
19	#	All reads in parallel, wait for all to complete
20	SWT	Begin Async
21	SMB	File Read
22	SMB	File Read
23	SMB	File Read
24	SMB	File Read
25	SMB	File Read
26	SMB	File Read
27	SWT	End Async
28	SWT	Raise Event
29	SWT	End Thread
30	SWT	Wait for Event
31	SMB	Tree Disconnect
32	SMB	Session Logoff
33	SMB	Close SMB TCP Connection

Protocols Supporting Async Operations

The following Protocols support Async operations:

- CIFS-SMB
- SMB2
- iSCSI (SCSI)
- Fibre Channel (SCSI)
- NFSv4.1
- HTTP/HTTPS
- HTTP Storage (CDMI, OpenStack Cinder, OpenStack Swift, and Amazon S3)

NOTES

Actions Supported - Currently ,Thread blocks support all Load DynamiX Scenario Control Actions, NFSv4.1 (Write/Read) and SCSI (LUN Write/Read), CIFS-SMB, SMB2 HTTP and HTTP Storage protocols and Async blocks support SCSI (LUN Write/Read), NFSv4.1 (Write/Read),

CIFS-SMB/SMB2, HTTP/HTTPS (GET, HEAD, PUT, DELETE, OPTIONS, TRACE) and (see Caveats below) HTTP Storage protocols. If unsupported Actions are put inside Async or Thread blocks, a run-time message of the form **Begin Thread at position <24> is ignored - unsupported protocol** will be generated. The Thread or Async block will be ignored - Actions inside will be executed but only executed as normal (sequentially).

Response Handling - All Response Handling behaviors are the same inside of Thread or Async blocks. If the Scenario Impact of an Action executed inside of a Thread or Async block is Terminate then the Scenario will be Terminated regardless of the state of the other Actions executing inside that block.

Events - Events may be sent or received inside or outside of Thread or Async blocks.

Synchronization - It is up to the Tester to make sure that Actions executed within Thread or Async blocks are appropriately synchronized. The Begin Async Wait Until Completion == TRUE forces the operations inside an Async block to be synchronized but otherwise it is up to the Tester to guarantee the appropriate order of execution.

Caveats -

- Begin-End Loops: Async blocks only execute ONCE regardless of the loop count if Thread Controls maintain their Default settings (see section above on Thread Controls)
- Actions within a Begin Async ... End Async block must be independent of one another because order on the wire is not guaranteed when the Actions are executed in parallel.

Troubleshooting Projects

Troubleshooting Projects

Load DynamiX Projects often require some amount of debugging or troubleshooting to get them working the first time or to address issues that arise in working tests. This chapter identifies some of the tools and files that are available to help with the troubleshooting process and key areas to check. See [Tips and FAQ section](#) for some tips and answers to frequently asked questions.

Troubleshooting Considerations

Recommended considerations for troubleshooting and debugging test Projects developed in the Load DynamiX TDE:

Kerberos Authentication

When building Projects that require Kerberos authentication (discussed in the Test Creation chapter), it may be very useful to capture the Kerberos interactions between a PC and the device that is being tested (DUT) that requires Kerberos authentication to see exactly what Kerberos information is passed between them. The process capture this data is to have a sniffer (e.g. Wireshark running on a PC) capturing data on the network between the PC and the DUT and then turn on the PC that will use Kerberos to log in to the DUT. This will capture the packet flow between the PC and the DUT which will be helpful in building Kerberos Projects

Windows Authentication on a DUT

If it is necessary to create a Project that behaves the same as Windows does when it authenticates access to a DUT, follow these steps to capture that information:

Before starting

- Install Wireshark on your Windows PC.
- Get the IP address of the DUT and the name of a Share on that DUT.
- Get a valid user id and password for that user ID on the DUT.

Now:

- Open any Command window.
- Start Wireshark using the Windows GUI..
- In the Command window, type this command, but DO NOT type <Enter> yet:
 - Net use <UNUSED DRIVE LETTER>: \<DUT IP ADDR>\<SHARE NAME> /User:<VALID USER ID> <PASSWD>
- In Wireshark, click Capture -> Interfaces and select the NIC that you communicate through.
- Click Capture -> Start.
- In the Command window, press <Enter>.
- In the Command window, type <UNUSED DRIVE LETTER>:<Enter>.
- In the command window type dir<Enter>.
- In the command window, type type <ANY FILE NAME ON SHARE><Enter>.
- In Wireshark, Click Capture -> Stop.

Error and Warning Messages

Does the compiled Project produce 0 errors? Does an executed Project produce no Errors or Warnings? If not, examine the log in the Output window to see what is being complained about. Compilation Errors will prevent a Project from being executed and must be fixed before execution will begin. Compilation Error messages are of the form:

Action "<Action_Number> <Action_Name>" <Action_Message>

Parameter "<Action_Field_name>" <Parameter_Message>
 <Action_Sub_Field_Name>: <Sub_Field_Message>

Where

- <Action_Number> - line number within the scenario that is failing
- <Action_Name> - the name of the Action at the line number indicated
- <Action_Field_name> - the name of the input property group that has a problem
- <Action_Sub_Field_Name> - the sub field within that property group that has a problem

When Projects are executed, they can produce Errors (such as segmentation faults, or not being able to establish a connection to the target device) or Warnings. Messages in the Output window of the form

```
error: 172.17.1.43:7 - appliance error: <ERROR Device [7]: Port Link failure. Check port
wiring and/or switch port status.>
warning: 172.17.1.43:7 - appliance warning: No devices running, exiting...
error: 172.17.1.43:7 - error occurred. Execution stopped
```

indicate that the Project was unable to establish contact with between the client scenario and its target device. This could mean a typographical error in the Actions that establish the connection or that the target device is not on line. Using tools such as Packet Capture (see PCAP below) can help in deducing the cause of these kinds of errors.

Warnings should not prevent the Project from executing but understanding these Warnings and addressing the cause is appropriate. Warning messages are of the form

warning <Appliance_IP_Address> - <Warning_Message>

Example:

```
warning: 172.17.1.43 - software versions do not match: appliance software is 0.22.11495 TDE is 0.22.11673
```

In this case the warning has to do with the version of the TDE and Appliance not being the same. This situation might be of no consequence or it might be a cause of a project not executing correctly so even a Warning message should be investigated fully.

Test Failures

Developing Load DynamiX Projects is a programming task and requires the skills and due diligence that any programming effort requires. It is a task that requires the following kinds of knowledge on that part of the person developing the test Scenarios:

- Knowledge of the protocol being used in the test. Whether NFS or CIFS-SMB or iSCSI or HTTP or Kerberos, the developer must be aware of the role of the individual protocol commands and what the commands need as input. For example, CIFS-SMB File Write Block Size may not exceed 64K bytes.
- Knowledge of the configuration of the network and the devices under test. IP addresses, gateways, security, filesystems structure, end user permissions, etc - the state of all of these items are potentially critical to the success of the test.
- What is the expected outcome of the test? Knowledge of expected success and/or failure of the elements of the test are important to implementing the correct behavior in a test. For example, if a file being created is expected to fail then configuring the [Response Handler settings](#) for that specific operation to expect failure is the right programming step, however it is not the default behavior of the create options so it must be a conscious thought on the part of the person developing the test.

When tests fail unexpectedly, evaluate the

- Network configuration (Logical Port/Network Profile settings, input to connect actions, etc)
- Security and Permissions settings: user ids, passwords, security system interactions

- (Kerberos, NTLM, etc)
- Programming logic (expected errors, create before use, cleanup, etc)
- Filesystems structure (expected/created directory structure or files)
- Device connectivity - using the TCP or UDP Echo protocol to make sure that Load DynamiX Appliances have connectivity to devices that are to be used by a Project

Desired Results

There are a number of capabilities that the Load DynamiX product provides that are intended to help determine if the desired results are achieved.

Ping - Load DynamiX Appliance Test Ports are only active during test execution so Pinging them prior to running a test will not produce a positive response. If it is necessary to Ping the Appliance Test Ports during test execution to verify that the Test Port is active, please limit the frequency and size of Ping requests as it can impact Test Port performance during test execution. Load DynamiX suggests setting the Ping packet size to less than 32 bytes (-s option on the linux/unix ping command or -l option on the Windows ping command)

PCAP (Tracing Parameters) - packet capture and review is often one of the easiest ways to find issues with test programs including items such as:

- Source and Destination IP addresses
- Protocol specific data (e.g. CIFS-SMB or NFS file names, directory names, read and write buffers, return codes)
- Event order (in what sequence to different Actions take place)

Protocol	Info
SMB2	NegotiateProtocol Request
SMB2	NegotiateProtocol Response
SMB2	SessionSetup Request, NTLMSSP_NEGOTIATE
SMB2	SessionSetup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
SMB2	SessionSetup Request, NTLMSSP_AUTH, User: WORKGROUP\USER001
SMB2	SessionSetup Response
SMB2	SessionLogoff Request
SMB2	SessionLogoff Response
SMB2	NegotiateProtocol Request
SMB2	NegotiateProtocol Response
SMB2	SessionSetup Request, NTLMSSP_NEGOTIATE
SMB2	SessionSetup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
SMB2	SessionSetup Request, NTLMSSP_AUTH, User: WORKGROUP\USER002
SMB2	SessionSetup Response
SMB2	SessionLogoff Request
SMB2	SessionLogoff Response
SMB2	NegotiateProtocol Request
SMB2	NegotiateProtocol Response
SMB2	SessionSetup Request, NTLMSSP_NEGOTIATE
SMB2	SessionSetup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
SMB2	SessionSetup Request, NTLMSSP_AUTH, User: WORKGROUP\USER003
SMB2	SessionSetup Response
SMB2	SessionLogoff Request
SMB2	SessionLogoff Response

The WireShark view of a PCAP captured by a Load DynamiX SMB2 test shows (using the filtering capability of WireShark) how the Actions defined in a Load DynamiX test are translated into real SMB2 packets/Actions. In this case, the filter eliminates all packets except those that are an SMB2 protocol packet. The Scenario in question is relatively simple - establish contact with the SMB2 device, logon and logoff as various users in a loop.

#	Protocol	Name
1	SMB2	Open SMB2 TCP Connection
2	SMB2	Negotiate
3	SMB2	Session Setup
4	SMB2	Logoff
5	SMB2	Close SMB2 TCP Connection

Name

Name	Value
Input	
Connection Handle	Default
Credits Charged	0
Credits	1
Authentication	
GSSAPI	Disable
Authentication method	NTLM only
NTLM authentication o...	
Domain Name	WORKGROUP
Machine Name	
User Name	=@UP(USER)
Password	=@UP(PASSWORD)
NTLM Flags	0x00080007 (524295)

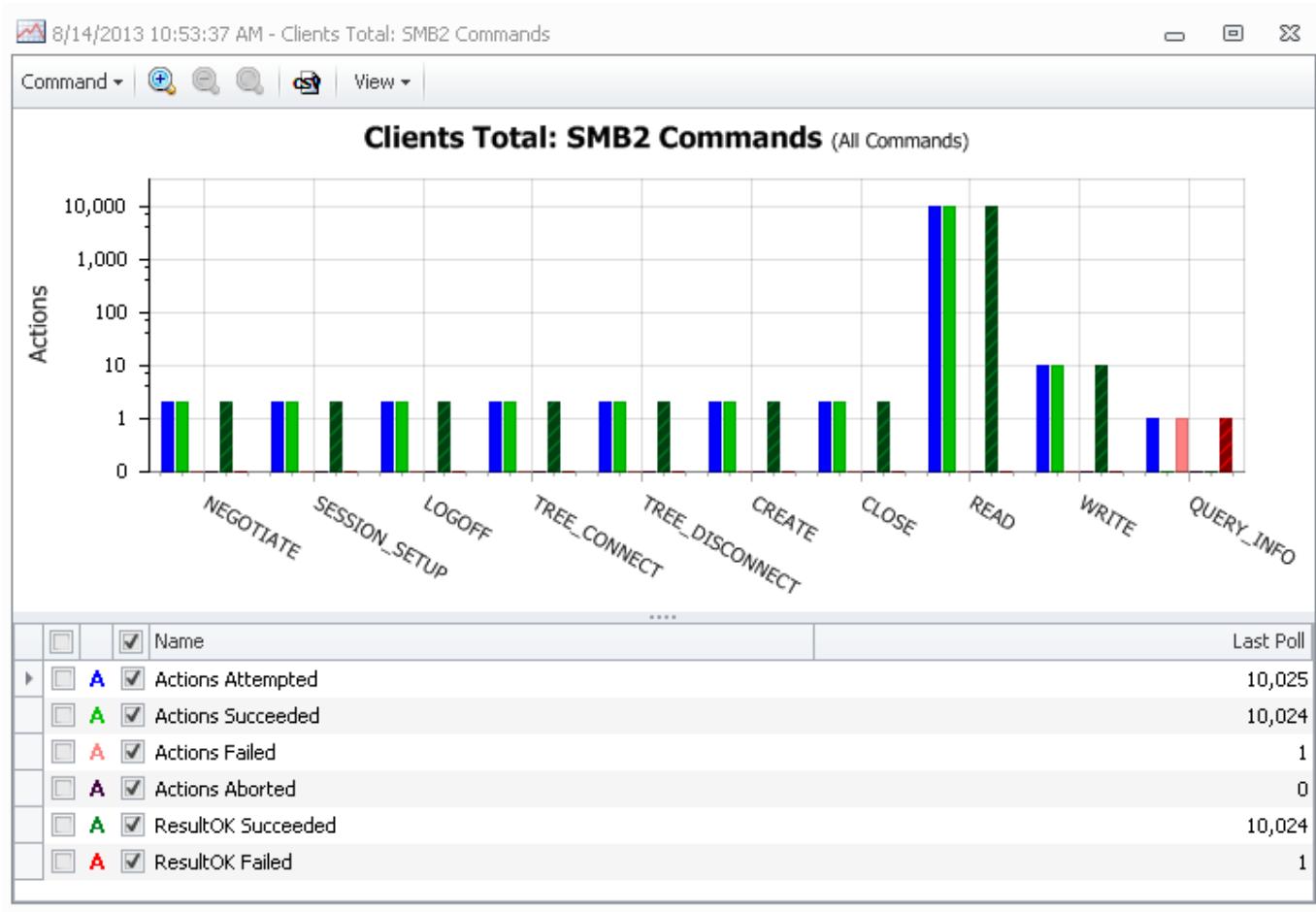
but it is easy to see how the information extracted from the PCAP file can be used to help identify and correct issues in the Load DynamiX test. See the [Test Creation section](#) for a discussion of how to configure Tracing Parameters.

NOTE: Tracing Parameters are not a performance tool. Using Tracing Parameters to capture PCAP data will reduce performance in the Load DynamiX Client and Server software.

Results Folder - the results folder contains the statistics and log files that are captured during the execution of a Load DynamiX test

- Statistics - statistics are captured real time and may be observed by opening the Results Folder during a test run and selecting the statistic that is interesting. For example,
 - SMB Commands - to see how the execution of SMB Actions are doing
 - Load Status - to see how the various Scenarios launched by the test are doing, etc.

While these statistics can be used for real time evaluation of the progress of a test, they are also excellent troubleshooting tools. For example, the SMB2 Command statistics output will indicate which SMB2 Commands are succeeding and which are failing. In the case below, all commands except the **QUERY_INFO** Action succeeded. Also shown in this graph are the Response Handler data (green cross-hatched data indicating the number of Actions that resulted in an Success completion status and the red cross-hatched data indicating the number of Actions that resulted in an Failure completion status).



Client and Server Port log files - at the end of a test run, the Load DynamiX appliance sends a log file to the TDE containing execution details for the test. There is a log file for each Logical Port in the test and this log file can be immensely helpful in determining the cause of test failures. In the example log file below, notice that, two of the Scenarios aborted. Looking further down the log file, all of the READ_ANDX commands are failing so that would indicate that those are the CIFS-SMB Actions that need troubleshooting. In the top left hand corner of the Client and Sever logs are information filters. By clicking on these filters, the user can eliminate or reinstate the information of this type. When the filter shows Yellow, the information is still visible in the log. Click the filter and it will change color to Gray and the information of this type will no longer be visible. Click it again, it will change color back to Yellow and the information will return to the log file.

Client Port 0(172.17.1.49 port 3).log				
	Line	Type	Date / Time	Text
	10	Status	9/29/2010 11:03:05 AM	Execution stopped.
11	Info		9/29/2010 11:03:06 AM	Finalizing statistics...
12	Debug		9/29/2010 11:03:06 AM	Total scenarios attempted: 204686
13	Debug		9/29/2010 11:03:06 AM	Total scenarios succeeded: 0
14	Debug		9/29/2010 11:03:06 AM	Total scenarios failed : 204684
15	Debug		9/29/2010 11:03:06 AM	Total scenarios aborted : 2
16	Info		9/29/2010 11:03:06 AM	=====
17	Info		9/29/2010 11:03:06 AM	=====
18	Info		9/29/2010 11:03:06 AM	SMB_COM Actions: Attempted Succeeded Failed ...
19	Info		9/29/2010 11:03:06 AM	=====
20	Info		9/29/2010 11:03:06 AM	Total: 3889024 3684338 204685 ...
21	Info		9/29/2010 11:03:06 AM	-----
22	Info		9/29/2010 11:03:06 AM	SMB_COM_CLOSE 204685 204685 0 ...
23	Info		9/29/2010 11:03:06 AM	SMB_COM_READ_ANDX 204685 0 204685 ...
24	Info		9/29/2010 11:03:06 AM	SMB_COM_WRITE_ANDX 2456224 2456223 0 ...
25	Info		9/29/2010 11:03:06 AM	SMB_COM_NEGOTIATE 204686 204686 0 ...
26	Info		9/29/2010 11:03:06 AM	SMB_COM_SESSION_SETUP_ANDX 204686 204686 0 ...
27	Info		9/29/2010 11:03:06 AM	SMB_COM_TREE_CONNECT_ANDX 204686 204686 0 ...
28	Info		9/29/2010 11:03:06 AM	SMB_COM_NT_CREATE_ANDX 409372 409372 0 ...
29	Info		9/29/2010 11:03:06 AM	=====
30	Info		9/29/2010 11:03:06 AM	=====
31	Info		9/29/2010 11:03:06 AM	=====
32	Info		9/29/2010 11:03:06 AM	SMB_COM Responses Handled: Attempted Succeeded Failed ...
33	Info		9/29/2010 11:03:06 AM	=====
34	Info		9/29/2010 11:03:06 AM	Total: 3889024 3684338 204685 ...

Network statistics are captured in Client and Server log files for debugging purposes only. These statistics can be used to debug issues at the lowest levels of the communications process. The following measurements with non-zero values may indicate that the Load DynamiX hardware and software are having difficulties sending and/or receiving packets.

rx packets dropped - indicates that the Load DynamiX Appliance received packets that it then was unable to process due to the packet's contents or status

tx packets dropped - indicates that the Load DynamiX Appliance software attempted to send more packets than the hardware could handle

rx packets out of buffer space - indicates that the Load DynamiX Appliance software is unable to receive packets at the rate that they are being received by the hardware.

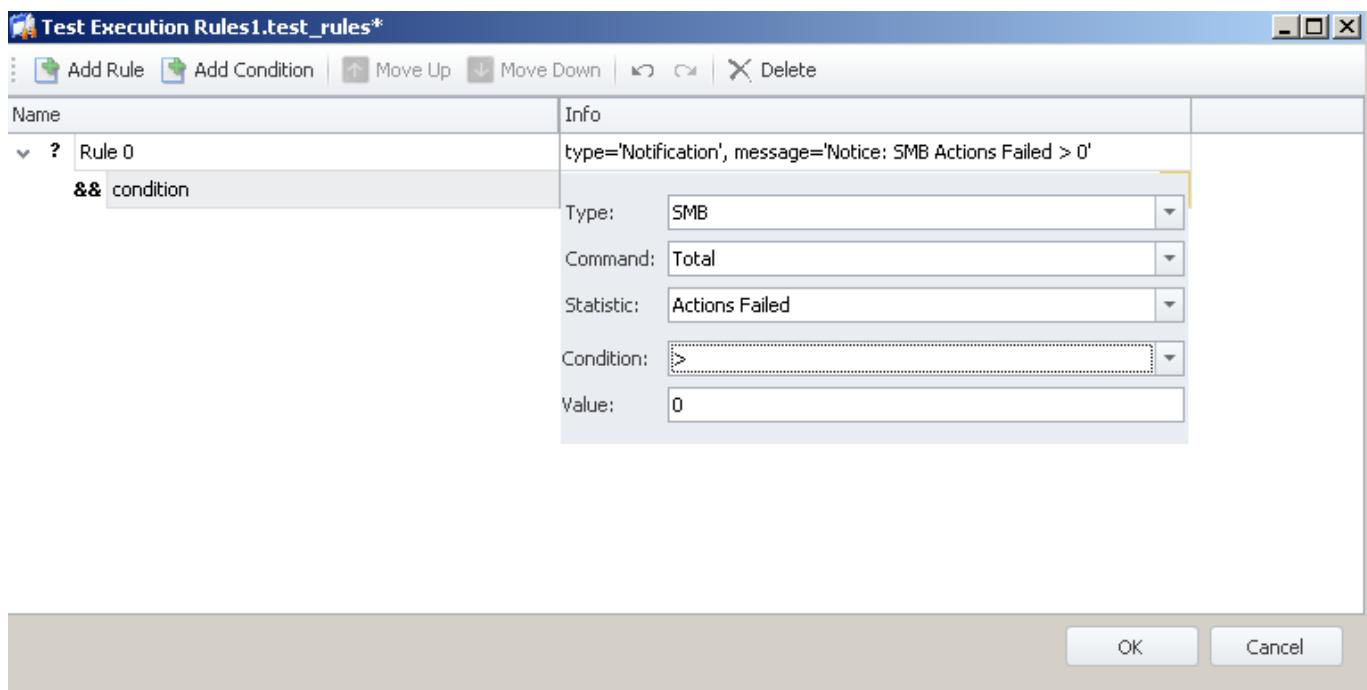
missed packet errors count - indicates that the Load DynamiX Appliance is unable to receive packets from the wire at the same speed in which they are arriving

These statistics are reported as Rx and Tx Errors in the Network Bandwidth graph..

=====
Network statistics: Port 0
=====

rx packets total:	88513
rx bytes total:	17333486
tx packets total:	88299
tx bytes total:	89144508
rx packets dropped:	0
rx bytes dropped:	0
tx packets dropped:	0
tx bytes dropped:	0
rx vlan tags count:	0
tx vlan tags count:	0
tx packets link failure:	0
tx bytes link failure:	0
tx packets duplicate:	0
tx bytes duplicate:	0
rx broadcast packets:	96
tx broadcast packets:	3
rx 802.3X PAUSE ON packets:	0
rx 802.3X PAUSE OFF packets:	0
tx 802.3X PAUSE ON packets:	0
tx 802.3X PAUSE OFF packets:	0
rx packets out of buffer space:	0
rx undersized packet errors:	0
rx fragmented packet errors:	0
rx oversized packet errors:	0
rx jabber packet errors:	0
rx flow control unsupported errors:	0
rx receive length errors:	0
rx errors:	0
CRC errors count:	0
alignment errors count:	0
symbol errors count:	0
missed packet errors count:	0
single collisions count:	0
excessive collisions count:	0

Test Execution Rules - The following Test Execution Rule set could be used to monitor for Actions failing a CIFS-SMB test. See the [Test Creation chapter](#) for more details regarding Test Execution Rules.



Events - If cooperating Scenarios are not seeing Events that have been Raised check the following:

- Load Profiles - are the Load Profiles for the cooperating Scenarios identical? If not, make them identical.
- Are the number of rows of Event Message Strings in User Parameter Files greater than the number of concurrently executing Scenarios? If not, add enough unique Event Key rows to the User Parameter files to be greater than the number of concurrently executing Scenarios.
- Are Event Message Strings coming from User Parameter Files? If so, are two identical User Parameter Files in the User Parameters Map or dragged onto the Scenario's timeline entries? If not, the Event Keys for the cooperating Scenarios must come from pairs of identical User Parameter Files.
- For more Event Troubleshooting information see [Appendix: Load DynamiX Scenario Control Actions](#).

Tips and FAQ

TIPS and FAQ

Development Environment

- HTTP Proxy. If the Load DynamiX TDE is installed on a PC that uses an HTTP Proxy, the Proxy must be made aware of the addresses of all Load DynamiX Appliances.
- If a Load DynamiX Appliance can be reached via a Ping (ICMP) request but not visible by the TDE then it is likely that one of the key programs required to run Load DynamiX projects is not functioning correctly. It will be necessary to reboot the Appliance using the button labeled Reset on the upper left corner of the front of the Appliance. Pressing this button will reboot the Appliance and cause all of the required Load DynamiX software to get reloaded.
- HTTP access. The Load DynamiX TDE and command line Automation both use the HTTP protocol to communicate with the Appliance. If the system running the TDE or Automation cannot communicate with the Appliance via HTTP, no Projects will be executable. A simple test of HTTP access to the Appliance from any system is to open a browser on that system, enter the Appliances IP address in the browser's address bar. What should be seen is the word Load DynamiX in the upper left hand corner of the browser.
- Installation directories for Windows XP, Server 2003 and Server 2008 are listed in the [Product Installation chapter](#) but are not officially supported by Load DynamiX.
- .ZIP files imported into the TDE by the [Project Import feature](#) must have been created by the [Project Export feature](#).
- TDE Unhandled Exceptions cause a log file to be generated in the Log folder (see [Product Installation](#) for the location).
- Downgrading an Appliance to a prior release will not undo any Activation or Licenses present on that Appliance.

Execution Behavior

- No PCAP or Client/Server log files are created if a test execution is Aborted (red Abort button). However if the test is let run to conclusion or is Stopped (blue Stop button), log and PCAP files are generated.
- Once a Project cycles through all of the IP addresses specified in a Network Profile, it will loop back and begin using them again.
- Spanning Tree should be turned off in the test bed due to the startup delay it introduces when the Load DynamiX Appliance ports come on line during a test execution. Load DynamiX physical test ports are not enabled until test execution begins and in a Spanning Tree environment there is a delay while Spanning Tree about the ports. This delay can cause Load DynamiX tests to fail. If Spanning Tree cannot be disabled, an appropriate (30 second) ramp up period or Delay Execution Action must be inserted.
- The Load DynamiX Change Notify Action. This SMB2 command creates a set of watch criteria on directories or trees. The Server that receives a Change_Notify request sends a Change_Notify response to the Client whenever the watched-for changes happen. Because the notification can occur at any time, the Load DynamiX Change_Notify Action implements two modes of operation - Synchronous (the default behavior, Async property = False) in which the Change Notify Action will issue the Change_Notify request and wait for the Server to return the Change_Notify response or Asynchronous (Async property = True) in which the Change Notify Action will issue the Change_Notify request and continue executing other Actions, allowing the Change_Notify response to be sent whenever the changes occur. The statistics kept for SMB2 Scenarios that contain the Change Notify Action take into account the extra messages that are received by this command. See [Reference: SMB2 Command List](#) for other details.
- The Load DynamiX SMB2 Set Info Action is an example of where a Load DynamiX Action does "more" than the Protocol specification dictates. The Set Info request has defined sets

of File Information Type that can be set (File, Filesystem, and Security). The SMB2 specification defines 15 File attributes that can be set/unset by the Set Info request. The Load DynamiX SMB2 Set Info Action allows the Tester to set/unset 30 File attributes when the File Information Type property is set to File. The 15 File attributes not defined by the SMB2 specification are sent with the Set Info request in an area of message defined to be Reserved. The reason for this behavior is to allow the possibility of Negative testing - what will an SMB2 server do when it encounters bits that it might not be anticipating.

- Ping - Load DynamiX Appliance Test Ports are only active during test execution so Pinging them prior to running a test will not produce a positive response. If it is necessary to Ping the Appliance Test Ports during test execution to verify that the Test Port is active, please limit the frequency and size of Ping requests as it can impact Test Port performance during test execution. Load DynamiX suggests setting the Ping packet size to less than 32 bytes (-s option on ping for linux/unix or -l option on the Windows ping command).
- Load DynamiX Actions that are executed on a closed server connection will appear in the Results command statistics but will not be present in PCAP files created during the Project execution. The commands will show as "Failing" but in effect will not be attempted due to the lack of an open connection over which to send them.
- Many of the storage protocols supported by Load DynamiX do not have an "End of File" error or condition - they simply return an "OK" status and 0 bytes when a Read past End of File is attempted. CIFS-SMB, NFSv3, NFSv4, and NFSv4.1 all have this behavior. SMB2 does have an End of File error. So, when reading past the end of file, Load DynamiX Projects can behave differently both for Success or Failure of protocol commands as well as Data Verification results. For those protocols that do not support an End of File error (CIFS-SMB, NFSv3 and NFSv4), Read operations will show Success but Verification operations on those reads will show Failures. SMB2 will show Read command Failures but no Verification Failures.
- iSCSI TCP Reconnect: Multiple iSCSI Open TCP Connection Actions per Scenario are not supported. Load DynamiX Scenario Control actions inside of iSCSI Reconnect Blocks produce unexpected results.
- See the [CIFS-SMB Command List](#) reference section for how to create directories using the CIFS-SMB protocol.
- Use of an incompatible SFP+ transceivers will cause an error message containing the text "**<ERROR Device[X] Generic Failure: Status Code [Y]>**" to be received from the 5000/5102/5108S Appliances. See the [Product Installation](#) chapter for more details on support for SFP+ transceivers for optical or DA connections.
- On the Load DynamiX 6202/6204/6208/6202E Appliances, only the virtual WWPNs that are provisioned on Fibre Channel Targets will be used by the Load DynamiX Fibre Channel driver (i.e. even though the Load DynamiX Appliance may provision 10 virtual interfaces and if the target device is only provisioned to accept 8 of those, then on the 8 provisioned on the target FC device will be used).
- To verify that an SFP+ cable/transceiver combination to work between the Load DynamiX 5000 Appliance and a DUT/10GbE switch, verify Link Status using the TDE Ports & Appliances > Appliances tab entry for the Load DynamiX 5000. If the Link Status messages indicate no Link then the SFP+ cable/transceiver combination is not compatible. However, successful Link Status messages do not guarantee that traffic can be sent over this connection. Run a test to verify a working connection. Include a Tracing Resource in the project. If the PCAP file that results from the Tracing Resource contains only ARP packets then the SFP+ cable/transceiver combination is incompatible with the target DUT/switch even if the link status appears OK. See the [Product Installation](#) chapter for more details on support for SFP+ transceivers for optical or DA connections.
- On the Load DynamiX 6202/6204/6208/6202E Appliances, only the virtual WWPNs that are provisioned on Fibre Channel Targets will be used by the Load DynamiX Fibre Channel driver (i.e. even though the Load DynamiX Appliance is configured for multiple virtual WWPNs, the virtual WWPN will only be used if the Target is provisioned to allow it).

- On the Load DynamiX 6202/6204/6208/6202E Appliances, If NPIV is enabled on a Fibre Channel Physical Port, only the virtual WWPNs will be used.
- Statistics updates are sent to the TDE using the HTTP protocol and if for some reason the TDE is not able to receive an update, the update is lost but because of the short update cycle, little if any information is lost. The most likely cause of this situation is because a PC running the TDE goes into Hibernation mode while a test was running. Although Windows hibernation is automatically disabled while a test is running, if the PC is sent to hibernation or powered off by the user, the TDE will not receive updates from the Appliance. If the TDE does not receive the statistics update that contains the "end of project" indication then the TDE will not know that the Project has terminated. If the Duration timer that the TDE keeps completes and the TDE has not received the "end of project" indication then it will wait for at least 5 minutes and then it will stop waiting and issue a "Test Expired" error message.
- If unsupported Actions are put inside Async or Thread blocks, a run-time message of the form **Begin Thread at position <24> is ignored - unsupported protocol** will be generated. The Thread or Async block will be ignored - Actions inside will be executed but only executed as normal (sequentially).
- Projects that contain un-Licensed Protocols will not execute.
- The Load DynamiX Fibre Channel Tracing Resource does not capture Fibre Channel FLOGI/PL:OGL exchanges.
- Certain SAN switches will shutdown a switch port if more Initiator WWPNs (NPIV) are used than are configured for that port on the switch.
- Fibre Channel processing of invalid requests. The lower level Fibre Channel interfaces on the Load DynamiX 6202/6204/6208/6202E Appliances will not send SCSI requests to FC devices that are determined by the low level interfaces to contain invalid information. A simple example is LUN number. If the LUN field of a SCSI Read Capacity or Test Unit Ready command (for example) contains 10 and the targeted FC device does not have LUN 10, the Fibre Channel interfaces will not send the command. The command will be marked with error reason "Transmission Failed" even though transmission was never really attempted. The Transmission Failed error condition cannot be handled by the SCSI Response Handling code so the scenario will always fail in this condition.
- Load DynamiX Server Scenario virtual disk space. LoadDynamiX Appliance ports when acting as "Server" (e.g. SMB, SMB2, NFSv3, or HTTP Server) have 1GB of ram memory to be used as virtual disk space. When files are created and written to in this virtual disk space by a Project, the files will consume the virtual disk space unless deleted during the Project's execution. Files created by one Scenario can be accessed by future instances of that Scenario during the Project's execution. Files created on Port X cannot be accessed by a Scenario running on Port Y. All Server files are erased when the Server Scenario terminates.
- SCSI Servers (Start SCSI Server Action) do not actually write data written to them to disk so data verification using the Start SCSI Server Action is not possible.

Project Configuration

- Checking the Disable Hardware Checksum box in the Network Profile dialog box will seriously impair network performance of the Load DynamiX Appliance because the Load DynamiX Appliance Firmware will now compute packet checksums.
- TCP Delay ACK Behavior: The TCP ACK Delay Interval controls how Load DynamiX will send ACK messages when responding to received data. When TCP Delayed ACK is configured in a Load DynamiX Network Profile (TCP ACK Delay Interval set to any value > 0), each TCP Connection within that profile will delay sending TCP ACKs for received data until one of the following happens: 1) Data is sent on this TCP Connection, 2) The TCP Delay ACK Interval expires, or 3) TCP ACK Delay Size Limit bytes of unacknowledged received data is accumulated. This value is set in the Network Profile property and defaults to 16383. See [Appendix: Jumbo Frames and Delayed ACK](#) for more detail on Delayed ACKs.
- Because packets sent by the Load DynamiX Appliance are stored in PCAP files before they are transmitted and the Appliance lets the network hardware calculate packet checksums,

these packets will show up in PCAP viewers as having invalid checksums. This issue can be ignored.

- Client Load Profile. The New Scenarios Per Second option should only be used for Scenarios that take less than a second to complete, otherwise the Scenario count will build up until the target server or device is overwhelmed.
- Server Load Profile. The Server Load Profile has no real impact on how Servers behave on the Load DynamiX Appliance. Ramp Up time and settings in the Load Profile are ignored. Ramp Down time is set to the smallest of the Client Ramp Down times.
- DO NOT copy resource files directly from one Project folder to another Project folder - this will cause confusion by the TDE. If sharing Project Resources is desired, copy the Resource to be shared into the Resource Library, My Resources folder and then copy it from this folder into the Project that needs it.
- CIFS-SMB Session Setup Action contains an input field named Virtual Circuit. The default value for Virtual Circuit == 0 which has the behavior of closing any open virtual circuits/sessions for the IP address requesting the new session. The default value behavior is impactful in tests where there are a small number of available IP addresses but a reasonable number of valid Users. If it is necessary for the same IP address to have multiple sessions open with a CIFS-SMB server then the value of the Virtual Circuit input must be > 0 and must be unique for each unique Virtual Circuit.
- CIFS-SMB File Write Action Block Size may not exceed 64K bytes.
- Maximum Open Handles (all types, connection, user-id, file, directory, etc) per Project theoretical limit is 40M open handles/physical port on the appliance. The achievable limit during a test run will be determined by the DUT.
- Maximum Open Handles (all types, connection, user-id, file, directory, etc) per Scenario theoretical limit is 16M open handles/scenario. The achievable limit during a test run will be determined by the DUT.
- Maximum Number of Open Files per Project theoretical limit is 40M open files/physical port on the appliance. The achievable limit during a test run will be determined by the DUT.
- Maximum File Size that can be written theoretical limit is 8 TB. The achievable limit during a test run will be determined by the DUT.
- Maximum Duration of a Project executing on a Physical Port is 1000 hours. Maximum Duration of a Project executing on a Virtual Appliance is 72 hours.
- Any TCP Open Action (CIFS-SMB, SMB2, NFSv3, NFSv4, NFSv4.1, HTTP and iSCSI) executed inside a loop without a corresponding TCP Close Action in that loop will get executed only a single time. If executing Open Actions in a loop is required, be sure to include the corresponding Close Action in that loop.
- When using User Parameter files to provide User IDs and Passwords to internal Load DynamiX servers in the Start Server Actions, the presence of blank (empty) entries in the User ID column should be noted: A single blank User ID cell with non-blank Password is allowed - this is the blank User ID definition. Subsequent blank User ID and valid Password entries will be ignored. A valid (non-blank) User ID entry and blank password will create a User Id with a blank password.
- When using User Parameter files to provide User IDs and Passwords to CIFS-SMB/SMB2 Session Setup Actions, the presence of blank (empty) entries in the User ID column should be noted: Blank User ID cell with non-blank Password is allowed and an attempt to establish a Session for the User ID == blank. If there is a blank User ID set up with this Password, the setup will succeed.
- SCSI Sequential device (SSC command set) Read and Write command Chunk Size limits vary depending on the target SCSI device. Observed maximum is 128MB. The value for any given SSC device can be seen in the response to the Read Block Limits command in a PCAP file.
- Fibre Channel Read and Write Chunk Size limits vary depending on the target FC device. Observed maximum is 128MB. Using Chunk Size values greater than 128MB will result in failed Read and Write commands.

- Use custom MAC addresses with custom IP addresses (and make the address pool size the same) when running Projects with large numbers of clients. Not doing so may introduce an undesired effect on some DUT, since the MAC address is often used as a hash key for client uniqueness, and having thousands of IPs with the same MAC could introduce performance issues not present in real-world scalability tests.

Project Design

- Maximum Project/Test Duration is 1000 Hours.
- Test success or failure will be impacted by the protocol-correctness of the test being executed. For example, in an CIFS-SMB test, the Negotiate Action always precedes the Session Setup which precedes the Tree Connect, etc. Likewise, an NFS protocol test must have well ordered Actions to succeed.
- When using multiple user ids and passwords to concurrently access a CIFS-SMB device, be sure to provision the User Parameter file with at least 20% more user ids and passwords than the test expects to use because there is some likelihood that two Scenarios could use the same user id and password at the same time.
- Client Load Profile. There is no rule of thumb for the optimal number of clients or connections per Logical Port. It is a function of what the clients or connections are doing. In no load condition (no reads or writes), no more than 1 million clients or 100K connections/second, per Logical Port but once reads and/or writes are added to the mix then predicting what the optimal configuration is problematic.
- The Load DynamiX Sample Projects delivered with each release provide excellent guidance as to how to structure the Actions in HTTP, iSCSI, CIFS-SMB, SMB2, NFSv2, NFSv3, and NFSv4 tests. See [Appendix: Load DynamiX Sample Projects](#) for a list of the available sample Projects.
- Understanding, in detail, what a Project is to accomplish and its requirements, before trying to execute it are key to limiting the time and energy required to get a Project executing correctly. In particular, understand the characteristics of the device under test. For example, if the test intends to log into this device with a 1000 users, do those 1000 users exist on the device, or if the test wants to create 100s of MB of file storage, is there 100s of MB available.
- Variables containing strings used in Action input fields that expect a numeric value will produce the numeric input of "0".
- Math operations are supported in Formula (see [Appendix Functions and Formula](#) for details).
- For a list of the kinds of information typically required to design a Load DynamiX Project, see the Information Typically Required for Project Design section in the [Introduction chapter](#).
- Local User Parameter references (e.g. \$(A)) are not allowed in Functions (i.e. $=@$(A)+@(B) is not allowed).
- There is a specific algorithm that determines how many IP Addresses a given IPv4 or IPv6 Network Profile will generate. This algorithm is described in detail in the [Test Creation chapter](#) of this Help but it is simply (smaller of ([Max Number possible calculated using IPv4 Netmask or IPv6 prefix] and [IPv4 or IPv6 Count]) divided by [IPv4 or IPv6 Step]). IPv4 example: **Netmask 255.255.255.0 = 256 Max possible IPv4 addresses; IPv4 Count = 1000; IPv4 Step = 2; Smaller of (256,1000) = 256; 256/2 = 128** IPv4 addresses will be generated by this Network Profile.
- Scenarios that share Events must be operating on the same Logical Port and must have the same Load Specification. They may have different Network Profiles.
- Start Server Actions in Server Scenarios only ever use the first row of UP Files that are used as input to the Start Server Actions (e.g. IPv4 Address, IPV6 Address, Machine Name, Domain Name, etc) except if UP File columns are used as input to the List of User Names and List of Passwords fields. In those two cases, all rows of the columns referenced are used. See [Advanced Concepts: User Parameters](#) for more details.
- Fibre Channel Projects cannot combine NPIV initiators and MPIO connections.
- Fibre Channel Projects with MPIO Actions will fail if MPIO Enabled = False in the Logical Port resource.

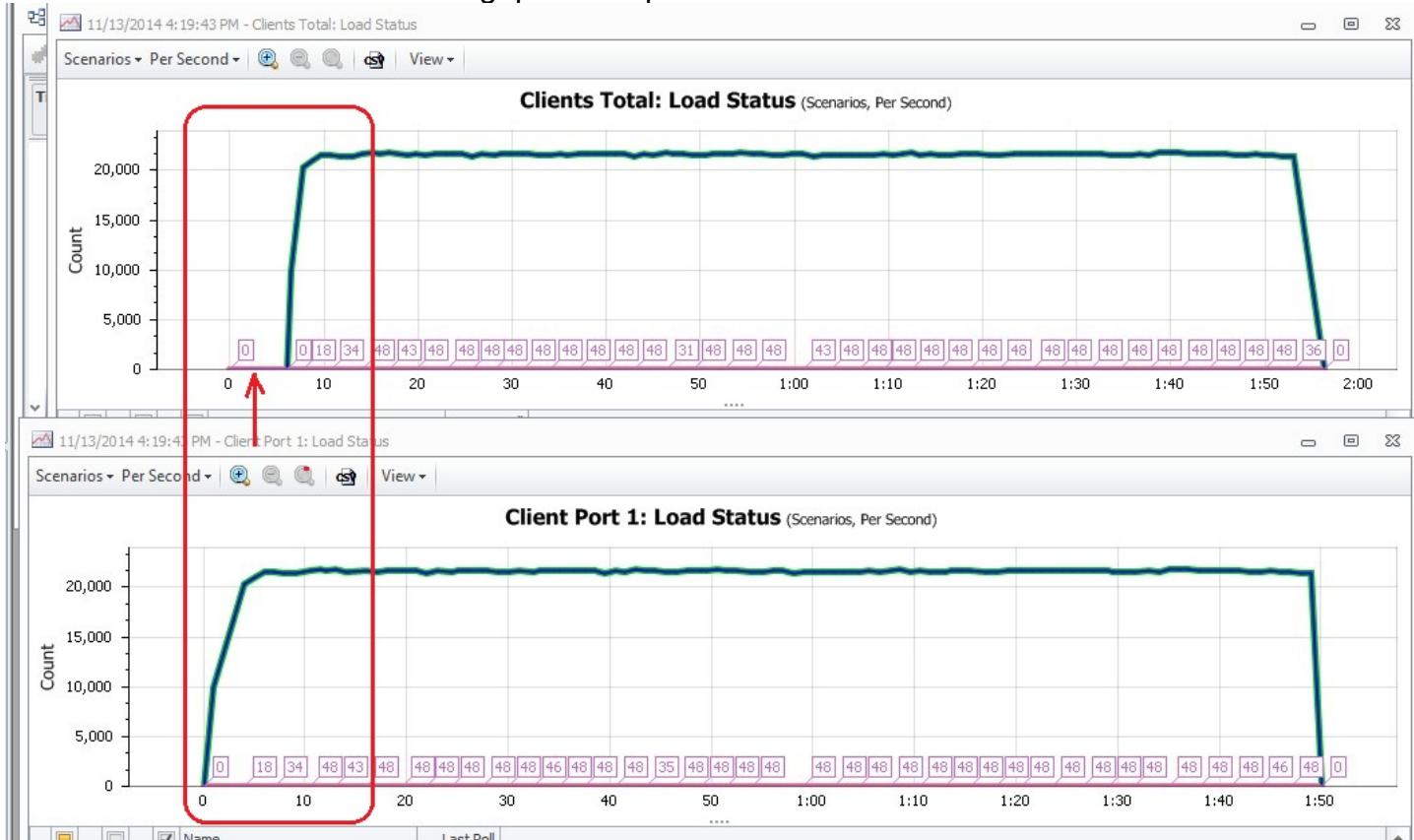
- Fibre Channel NPIV connections do not support Reconnect.
- The Load Specification Restart Scenario feature is ignored in Server Scenario Load Specifications.

Project Results

- Aborted Scenarios late in a Project run are often caused by a lack of time in the timeline for them to complete. Be sure that there is plenty of Ramp Down time for Scenarios that have been started to complete before the Project timeline expires. See [Executing Tests and Assessing Results](#) for a detailed discussion of Aborting Scenarios and Actions.
- Aborted Scenarios early in a Project run may indicate that a DUT's maximum number of connections has been reached.
- Aborted Scenarios in the middle of a Project run may indicate that a DUT is being pushed harder than it can effectively handle. See [Executing Tests and Assessing Results](#) for a detailed discussion of Aborting Scenarios and Actions.
- Aborted Scenario: Client is unable to open TCP connection after sending SYN. Triggered by "Maximum TCP Retransmit Count" and "Maximum TCP Retransmit Interval" combination. See [Executing Tests and Assessing Results](#) for a detailed discussion of Aborting Scenarios and Actions.
- Aborted Scenario: Client is not receiving ACK on a packet in an open TCP connection. Triggered by "Maximum TCP Retransmit Count" and "Maximum TCP Retransmit Interval" combination. See [Executing Tests and Assessing Results](#) for a detailed discussion of Aborting Scenarios and Actions.
- Aborted Scenario: Client is not receiving any packets from a server for a time longer than specified in "TCP Inactivity timeout". See [Executing Tests and Assessing Results](#) for a detailed discussion of Aborting Scenarios and Actions.
- Aborted Scenario: Server drops connection before client sends FIN. See [Executing Tests and Assessing Results](#) for a detailed discussion of Aborting Scenarios and Actions.
- Rx/Tx errors in the TCP Details graph indicate that the Load DynamiX Appliances physical ports are having Receive and/or Transmit issues.
- Increasing Response time as captured in the Client Port(s) protocol Response Time graphs may indicate that the DUT is being pushed harder than it can effectively handle.
- TCP connection failure at the beginning of a Scenario is often linked to the Ramp Up time being too short.
- The Output window Error and Warnings counts do not reflect the real number of failed Actions or aborted Scenarios in a test. See the [Load DynamiX Test Development Environment and GUI section](#) for more details of what the Output Window displays.
- Automation Results. Project statistics are captured and stored in a Results folder just as if the Project was executed by the TDE. If the Project is open in the TDE when the Automation is executed, the new Results folder will not appear in the Results Explorer until the Project has been closed and re-opened, or until the Refresh function (right-click on any Results folder in the Results Explorer) is executed.
- Results folders with no protocol specific graphs (e.g. no SMB Actions or SMB2 Commands or etc.) typically indicates that the Project Scenario(s) were never able to connect with the DUT. This could be because of a bad or invalid DUT IP address or an unreachable IDUT P address in the TCP Connect Action.
- PCAP files with nothing but ARP commands means that the DUT's IP address was not found in any of the ARP tables that were searched. This could be because of a bad or invalid DUT IP address or an unreachable IDUT P address in the TCP Connect Action.
- TCP Connections Results folder graphs (or log file information) showing 100% failures is another indicator that the TCP Connection Action is using a bad or invalid DUT IP address or an unreachable IDUT P address.
- Substituting scaled or custom fonts for standard Windows fonts may cause issues when graphs are displayed.
- In Fibre Channel Projects, Read Capacity 10 Actions take as input the value of the LUN to be

queried. If this value is invalid for the target WWPN then the Read Capacity 10 action will result in a Transmission Failed error. This condition is not handled by the SCSI command [Response Handlers](#).

- NetMon will not process the PCAP files produced by a Tracing Resource on a Fibre Channel Project.
- The X-Axis for a specific Port represents the time for that Port. The X-Axis for Client Total reflects TDE time elapsed so that in graphs, there is a gap in the time between Clients Total and Client Port X. This gap is as expected - see below.



Performance

- Tracing Parameters are a very useful tool but should only be used when debugging the behavior of a Project, due to the performance impact of collecting the packets during test execution. Using Tracing Parameters to capture PCAP data will reduce performance in the Load DynamiX Client and Server software.
- Data Verification in Read Actions can degrade the overall performance of a Project.
- Protocol level behaviors such as Signing in the CIFS-SMB and SMB2 protocols can negatively impact performance.
- Writing specific data patterns such as the contents of ::DataContent files can negatively impact performance.
- The use of Jumbo Frames in test networks (MTU size > 1500) where large packets are supported throughout the network can
 - performance for large writes and reads.
- On Load DynamiX 1G Series Model 3000/3108 1000BASE-T ports, an MTU size of ≥ 8131 will cause performance degradation. It is recommended that for Projects striving for maximum performance with large packet sizes (e.g. reads or writes of > 8K bytes), that MTU size not be set to be greater than 8000. See [Appendix: Jumbo Frames and Delayed ACK](#) for more detail on MTU size.
- Ping - Load DynamiX Appliance Test Ports are only active during test execution so Pinging them prior to running a test will not produce a positive response. If it is necessary to Ping the Appliance Test Ports during test execution to verify that the Test Port is active, please limit

the frequency and size of Ping requests as it can impact Test Port performance during test execution. Load DynamiX suggests setting the Ping packet size to less than 32 bytes (-s option on ping for linux/unix or -l option on the Windows ping command).

- Optimizing Projects for Throughput: To get maximum throughput for a given Project requires the balancing the number of active users (Scenarios) and the amount of information that is being written or read in the context of the maximum capabilities of the underlying Ethernet network (e.g. 1Gbps, 10Gbps, etc). Use the Load DynamiX sample Project CIFS-SMB Full Duplex Payload as an example of a Load DynamiX Project that has already been tuned to maximize throughput. In this Project there are 20 concurrent Scenarios (users) each writing or reading 1.3GB of data for a total of 26GB of data which is approaching the maximum of amount of data that can be pushed through a 1Gbps link in the Project's ~110 second Timeline.
- Response time results for SMB Read and Write operations: Read - response time is measured from the time the Client sends the Read request until a Server responds with the first Read response packet. So, in general, Read response time will be similar or identical for small and large Reads because the Server returns a Read response packet immediately after the Read request packet. Write - response time is also measured from the time the Client sends the first packet of the Write request until the Server sends the first Write response packet. In the case of Writes, the SMB Server will not send a Write response packet until all packets associated with the Write have been received. Thus, small and large Writes may have significantly different response times.

TCP Stack

- Rx Drops - statistic only seen in TCP Details graph. Rx Drop counter increments whenever
 - A packet is received from a closed TCP connection
 - A packet is received from a still-open connection that has received its Last ACK.
- Rx or Tx Errors captured in the Results TCP Details graph indicate that the Load DynamiX Appliance ports are having receive and/or transmit problems.
- The Load DynamiX TCP stack will allow malformed TCP packets (e.g. received packets with an invalid checksum) to be passed up the stack to higher level services like NFS, CIFS-SMB, SMB2 clients and servers. The presence of these packets will be captured in the TCP Details graph and the Client and/or Server log file.
- Pause Frames. TX PAUSE Frames of 802.3X are NOT currently included in the statistics counts of either transmitted or received packets in any way, TDE or otherwise. However:
 - All Load DynamiX's physical ports indicate to the link partner at initialization time the ability to both receive and transmit TX PAUSE frames.
 - All Load DynamiX's physical ports also recognize the link partner's ability to both receive and transmit TX PAUSE frames and the ports align their flow control configuration accordingly, as defined by the standards.
 - All Load DynamiX's physical ports operate on received TX PAUSE frames as indicated by the sender of the frame in the frame's content.
 - All Load DynamiX's physical ports do not transmit TX PAUSE frames, except that the very first packet transmitted by Load DynamiX's immediately after the physical link establishment may be a TX PAUSE frame.

Appliance Configuration and Events

The URL "<http://<APPLIANCE IP ADDRESS>/api/events>" returns event information from the Appliance. The event information includes Port start and stop times, busy status, etc (Port "utilization" information).

The URL "<http://<APPLIANCE IP ADDRESS>/api/appliance>" returns configuration metadata from the Appliance such as firmware version, serial number, MAC address, IP address and

information about each Port such as type, fabric, speed, etc. See below for an example of information returned by a Load DynamiX U1122 Appliance (2x10GbE and 2x16Gb Fibre Channel Test Ports) with Admin Port IP address of 172.17.74.10:

<http://172.17.74.10/api/events>

```
{"events": [
    {"time":"2016-10-28 16:16:49.144579261 -0700", "event":"logger stopped" },
    {"time":"2016-10-28 16:16:49.405336686 -0700", "event":"logger started" },
    {"time":"2016-10-28 16:17:14.357215512 -0700", "port":0, "busy":true },
    {"time":"2016-10-28 16:17:14.357458279 -0700", "port":1, "busy":true },
    {"time":"2016-10-28 16:18:57.566251602 -0700", "port":0, "busy":false },
    {"time":"2016-10-28 16:18:57.568155955 -0700", "port":1, "busy":false },
    {"time":"2016-10-29 10:01:47.004372602 -0700", "port":1, "busy":true },
    {"time":"2016-10-29 10:01:47.004612296 -0700", "port":0, "busy":true },
    {"time":"2016-10-29 10:01:51.805344927 -0700", "port":1, "busy":false },
    {"time":"2016-10-29 10:01:55.805748259 -0700", "port":0, "busy":false },
    null
  ]}
```

<http://172.17.74.10/api/appliance>

```
{"appliance": {
    "version": "1.53.36416-Internal_only",
    "serialNumber": "ST0153",
    "platformType": "X9DRD-7LN4F",
    "timeStarted": "2016-09-14 20:25:21 -0700",
    "macAddress": "00:25:90:C3:19:C7",
    "IPv4": "172.17.74.10",
    "ports": [
      {
        "portID": 0,
        "slot": "4.0",
        "typeID": 5,
        "typeName": "10GbE Fiber",
        "fabric": "Ethernet",
        "speed": 10
      },
      {
        "portID": 1,
        "slot": "4.1",
        "typeID": 5,
        "typeName": "10GbE Fiber",
        "fabric": "Ethernet",
        "speed": 10
      },
      {
        "portID": 2,
        "slot": "2.0",
        "typeID": 8,
        "typeName": "16Gb Fibre Channel",
        "fabric": "Fibre Channel",
        "speed": 16
      },
      {
        "portID": 3,
        "slot": "2.1",
        "typeID": 8,
        "typeName": "16Gb Fibre Channel",
        "fabric": "Fibre Channel",
        "speed": 16
      }
    ]
  }}
```

Appendix: Office

Appendix: Office

Office consists of two command-line utilities, SwiftExcel.exe and SwiftWord.exe, used to create Microsoft® Word-like and Excel-like files containing random characters. Both commands make use of the method “RNGCryptoServiceProvider.GetNonZeroBytes” from the Microsoft® .NET Framework Class Library. Per Microsoft, this method “fills an array of bytes with a cryptographically strong sequence of random nonzero values”. SwiftExcel and SwiftWord must be executed through a command prompt. The standard installation directory for SwiftExcel is C:\Program Files (x86)\Load DynamiX\Load DynamiX TDE\swiftexcel and SwiftWord is at C:\Program Files (x86)\Load DynamiX\Load DynamiX TDE\swiftword.

Note that the location of the executable binaries may vary according to the version of the Windows operating system that is being used. The [Product Installation chapter](#) provides guidance on the locations for various Windows versions.

In order to simplify entering Office commands, best practice is to copy the two files to a directory close to the root, for example, C:\Office. Then, when you execute the commands, open a command window and cd to the directory that was created for Office.

Files created by the Office utilities are named numerically. For example, if you specify the creation of 10 files with no prefix, files are named 01 though 10. You can include a file name prefix. For example, if you use the /Prefix:Excel parameter for the 10 files mentioned above, the SwiftExcel Utility generates file names Excel01.xls though Excel10.xls.

Type either of the commands with no parameters to view command-line help.

SwiftExcel

Syntax is: **SwiftExcel /Count:x /RowCount:x /ColCount:x [/Prefix:x] [/Path:x] [/O]**

/Count:x - count of new files to generate (1 - 1000)
 /RowCount:x - number of rows in new files (1 - 10000)
 /ColCount:x - number of columns in new files (1 - 10000)
 /Prefix:x - file name prefix
 /Path:x - destination path for new files
 /O - overwrite existing files

Example:

```
SwiftExcel /Count:10 /RowCount:10 /ColCount:10
SwiftExcel /Count:10 /Prefixabc /RowCount:10 /ColCount:10 /Path:c:\myfiles\ /O
```

SwiftWord

Syntax is: **SwiftWord /Count:x /PageCount:x [/Prefix:x] [/Path:x] [/O]**

/Count:x - count of new files to generate (1 - 1000)
 /PageCount:x - number of pages in new files (1 - 1000)
 /Prefix:x - file name prefix
 /Path:x - destination path for new files
 /O - overwrite existing files

Example:

```
SwiftWord /Count:10 /PageCount:10
SwiftWord /Count:10 /Prefixabc /PageCount:10 /Path:c:\myfiles\ /O
```

Copyright © 2008-2017 Virtual Instruments Inc.

Appendix: Test Automation and LDX-E Integration

Appendix: Test Automation and LDX-E Integration

In a testing environment, there is often the need to run a test repeatedly. Users may want to change the configuration of the device and run the test again or the user may want to change one parameter in the test (block size, file size, etc.) and run the test again. Load DynamiX Automation allows users to quickly and easily automate (execute as a script) their tests.

The recommended approach to follow for Automation is:

- Testers develop and get working the basic test in the Load DynamiX Test Development Environment (TDE).
- Generate the configuration files that are used to automate the test.
- Make a copy of the Generated configuration files to execute and expand the capabilities of the basic test because the Tester will want to modify the Generated files and it is good to have the original intact in case it is necessary to go back to the original state.

This appendix explains how to use Load DynamiX Automation but does not teach any scripting language. It assumes the user knows one or more scripting languages (TCL, Perl, PowerShell, etc.).

Automation

Automation allows Testers to execute Load DynamiX Projects created by the TDE or by hand using scripting languages such as TCL, PERL, PowerShell, etc.

- Easily execute Load DynamiX Projects developed within the TDE in scripting environments such as TCL and PERL,
- Update User Parameter File contents directly,
- Change Scenario Load Profile start and duration values within AutomationConfig.XML,
- Change/Extend Scenarios by editing AutomationClientScenario and AutomationServerScenario files,
- Change Tracing Parameter, Network Profile values within AutomationConfig.XML,
- Update Test Execution Rules directly,
- Create entire Scenarios from scratch by defining XML files that are content and structurally compatible with those generated by the TDE and execute them the same way as TDE-developed Projects.

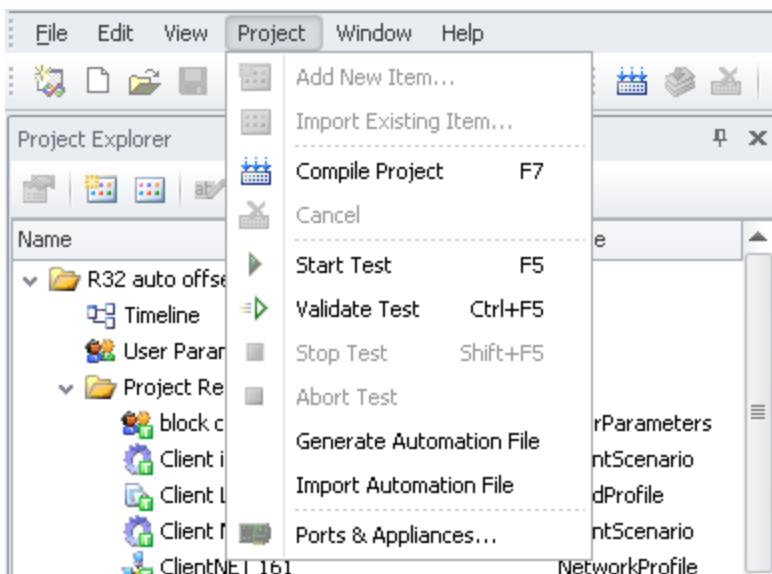
Fundamentals

The AutomationConfig folder contains all of the files necessary to Automate the execution of Load DynamiX projects. This folder is created using the:

- TDE main toolbar Project>Generate Automation File function,
- LdxCmd.exe /generate command line function.

Generating Automation Files

Automation files (AutomationConfig.XML, AutomationClientScenario-*.XML, AutomationServerScenario-*.XML, UserParameters-*.XML, TestExecutionRules-*.XML, DataFileSystem-*.XML.) are only generated when either the Project menu **Generate Automation File** menu item is selected or the LdxCmd.exe **/Generate** command line option is used.



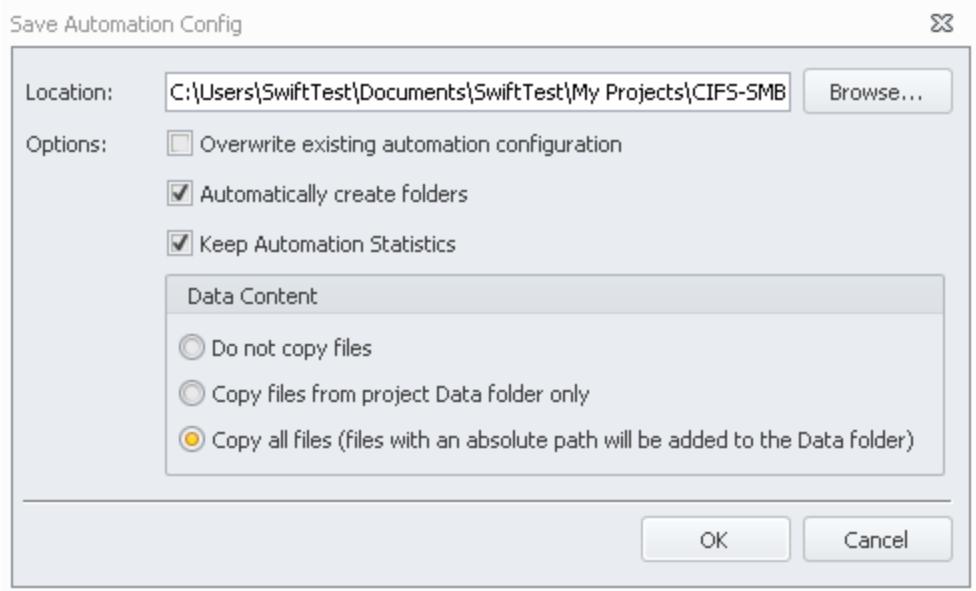
When the **Generate Automation File** menu item is clicked, the user is given the choice of where to have the Automation files stored. This allows the user to have multiple instances of the Automation files for a given Project in separate locations. When the **Generate Automation File** option is selected, the following dialog box appears. By default, the folder is ..\My Projects\<Test Name>\AutomationConfig. The user also can specify whether to **Overwrite existing automation configuration** files and whether to automatically create the named folder. If the **Automatically create folders** option is NOT checked and if the folder does NOT EXIST, the Generate operation will fail. If the **Overwrite existing automation configuration** option is selected, the end user will be asked if they want to overwrite the automation configuration files or not. If the end user answers No, they will be returned to this dialog box. If they answer Yes, the Generate option will proceed. If **Keep Automation Statistics** is checked, any existing Results folder contents will be left in place when the AutomationConfig folder is created.

Data Content options (allows Tester to manage the propagation of DataContent type files to the AutomationConfig folder):

Do not copy files: Ignore all DataContent type files (whether absolute path or in the project Data folder)

Copy files from project Data folder only: Only copy the files in the project Data folder to the AutomationConfig folder

Copy all files: Copy all files (absolute path as well as those in the project Data folder) to the AutomationConfig folder



Directory Structure, File Contents

Load DynamiX Projects are typically stored in the Projects Folder documented in the [Product Installation chapter](#). Each Project is stored in a folder with the same name as the Project. In the Projects Folder there are additional folders that are created when Automating Load DynamiX Projects. The following folders are created during the Automation process and will be discussed below:

- AutomationConfig Folder
- Automation Folder
- Automation/Results Folder
- Data (if DataFilesystem physical files are copied into the AutomationConfig folder)

In addition to these folders, the following .XML files are created during Automation and will be discussed below:

- AutomationConfig.XML
- Resource XML files

Folders:

Data - contains files copied from the Data folder in the Project folder and/or other folders if the Tester requests that during the AutomationConfig Generation process.

AutomationConfig Folder - this folder and its contents are created or updated when a Load DynamiX Project's Automation files are "Generated". See the Generating Automation Files topic below for a detailed discussion of the Generate process. The contents of this folder are the XML files necessary to execute the project using LdxCmd.exe.

There are two types of XML files contained in this folder:

- AutomationConfig.XML - the main, controlling XML file for Load DynamiX Automation. This file contains two types of information:
- Resource Definitions - the definitions of the settings for those Resources that do not reside in separate Resource XML files (see below). These Resources are defined in the top portion of the AutomationConfig.XML file (above the <!-- Project Timeline --> XML

tag). Examples of the Resources defined here are:

- Load Profile height and type (Load Profile duration is defined below the <!-- Project Timeline --> tag)
- Network Profile
- Tracing Parameters
- User Parameter File Map
- The XML statements that associate some of the Resource Files (User Parameter Files and Test Execution Rules) with their internal ID.

Assignment of Resource files to the Internal 4 digit IDs are used in by AutomationConfig.XML to refer to the external Resource files or the Resources defined in the AutomationConfig.XML file. The Internal ID is a 4 digit number of the form {1,2,5}XXX. These four digit IDs are assigned to the Resource XML files and used below the <!-- Project Timeline --> XML tag whenever that Resource is used in the Project. Logical Ports: client 1XXX, server 2XXX and all other resources are 5XXX.

- Timeline - Below the <!-- Project Timeline --> XML tag is the XML representation of the Project Timeline as created in the TDE. Below the <!-- Project Timeline --> XML tag are the:
 - Logical Port and definitions (Appliance IP address and Physical Test Port assignment)
 - Network Configurations (which Network Profiles are associated with which Logical Ports)
 - Client or Server Scenario definitions including reference to the Resource XML file that contains the Scenario XML and assignment of the Client or Server Scenario ID
 - Load Profile Start time and Duration (specified in Milliseconds) for each Client or Server Scenario
 - How the other Project Resources are deployed in the Project Timeline
- Resource XML files - these files are self-contained Load DynamiX XML files that contain Client/Server Scenario XML, User Parameter File XML, DataFileSystem.XML, and Test Execution Rules XML. These files are referenced in AutomationConfig.XML where the Project Timeline and Resource assignments are defined.

Automation Folder - this temporary folder is created or overwritten whenever a Load DynamiX Project is executed or validated using Automation (LdxCmd.exe /start or LdxCmd.exe /verify) and /project folder path is specified. When a project is executed/validated, the contents of the AutomationConfig folder are compiled and sent to the Appliance for execution. The compiled AutomationConfig folder contents are stored in the Automation folder. The contents of the Automation Folder are overwritten every time a Project is executed or validated.

Automation/Results Folder - also in the Automation folder is the temporary Results folder that is created and overwritten at execution time to hold the results statistics and logs that are retrieved from the Appliance during execution. The Automation folder and the Results sub-folder are overwritten every time the Project is executed or validated. The contents of the Results folder are copied back into the Project's Results Explorer folder to maintain the Results history of Automation executions.

Notes:

- Load Profile Ramp Up , Ramp Down and other Load Profile times are specified directly (see the <Loads> XML below for an example).

<ConstrainsConfig>

```

<ProjectName>12943.Load DynamiX.Automation</ProjectName>
<ProjectRevision>19</ProjectRevision>
<ProjectDuration>144000</ProjectDuration>
</ConstrainsConfig>
.....
<Loads>
  <Load>
    <Type>RampUp</Type>
    <Duration>9000</Duration>
  </Load>
  <Load>
    <Type>Steady</Type>
    <LoadID>5006</LoadID>
    <StartTime>9000</StartTime>
    <Duration>3351000</Duration>
  </Load>
  <Load>
    <Type>RampDown</Type>
    <StartTime>3360000</StartTime>
    <Duration>240000</Duration>
  </Load>
</Loads>

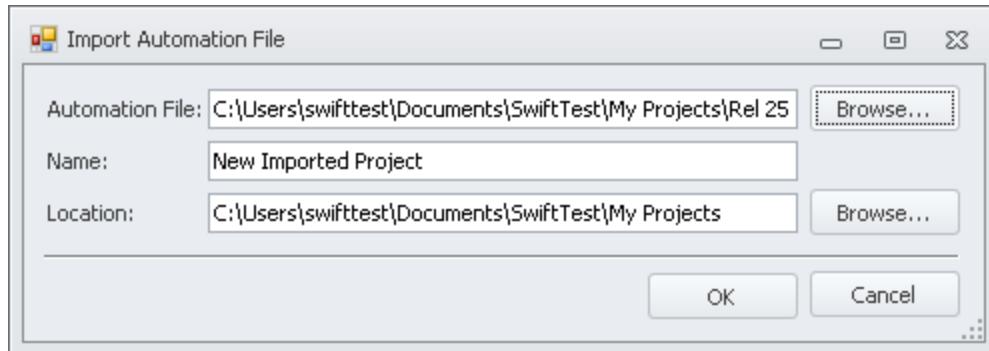
```

The example above is XML information extracted from the AutomationConfig.XML file.

- Automation files generated by the TDE, Resource XML files have a very uniform naming structure{Load DynamiX Resource Type}-{Internal-ID}.XML. This is NOT a requirement. The user may substitute other file names in the XML tags that associate a Resource XML file with its internal ID and these file names do not need to follow the file name structure used by the TDE.

Importing Automation Projects

AutomationConfig.XML files may be imported into the TDE which will create a version of the Project that can be viewed or manipulated within the TDE. This may be a convenient way to view or update Automation Projects that were created manually or initially created by the TDE and later updated manually. The process to Import an Automation Project is to click in the Import Automation File function in the Project drop down menu shown above. This will cause the following dialog box to appear on the screen:



Click the Browse... button to select the AutomationConfig.XML file, enter a Name for the Project if something other than the default is desired and pick a folder to hold the Project if the standard My

Projects folder is not appropriate and then click the OK button. A Project folder will be created in the specified location that can be viewed and/or manipulated by the TDE.

Basic Components

The components of Load DynamiX Automation that users interact with directly are the AutomationConfig.XML file that contains Load DynamiX Project parameters that users can change, the Client or Server Scenario files that contain the Actions executed by the Client or Server Scenario and the command LdxCmd.exe which is used to execute the Load DynamiX Projects from the command prompt or from a scripting language. **The Load DynamiX Project folder files are not required by Automation unless the /Generate or /Upgrade command line options are used.**

AutomationConfig.XML

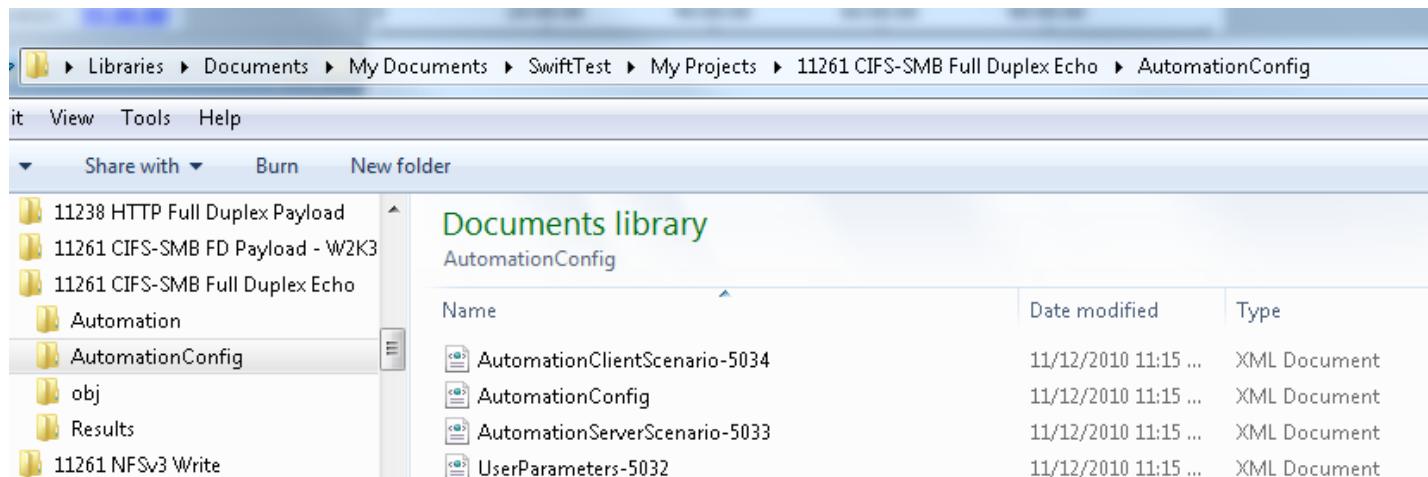
This configuration file contains basic information for both Clients and Servers.regarding:

- Load Profile
- Network Profile
- Port
- Scenario

It also contains information regarding:

- Appliances used,
- Logical Port mapping,
- Client network used,
- User parameters used,
- Tracing parameters used,
- Data File Systems used.

It is recommended that users should **NOT** modify the version of the Automation files that are created when Project menu > Generate Automation Files is executed. **Instead, users should copy the files in this folder to another folder and then modify and execute the copied files for automation.**



The table below shows the XML tags found in the AutomationConfig.XML file and their meaning. These XML tags are examples of the kinds of things that the Tester may need to change to get some change in behavior of the test.

XML meaning	AutomationConfig.XML Content
Comment	<!-- Scenario Configuration -->
Begin Client Scenario tag	<ClientScenarioConfig>

Port ID reference	<ClientPortID>1000</ClientPortID>
Network Profile reference	<NetID>5021</NetID>
Scenario ID #	<ClientScenarioID>5046</ClientScenarioID>
Scenario file name	<ClientScenarioFile>AutomationClientScenario-5046.XML</ClientScenarioFile>
Data file system tag	<DataFileSystemID />
User parameter file tag	<UserParametersID />
Tracing parameters tag	<TraceParametersID />
Scenario enabled tag	<Enabled>1</Enabled>
Begin Loads tag	<Loads>
Begin Load Profile tag	<Load>
Load profile reference	<LoadID>5023</LoadID>
Load Start time tag	<StartTime>0</StartTime>
Load Duration tag	<Duration>125000</Duration>
End Load Profile tag	</Load>
End Loads tag	</Loads>
End Client Scenario tag	</ClientScenarioConfig>

AutomationConfig.XML is the main focal point for automation. Scenario XML is stored in separate files, one file per Scenario (Client and/or Server). Scenario XML files are named

AutomationClientScenario-####.XML for Client Scenarios

AutomationServerScenario-####.XML for Server Scenarios

The "####" is a number generated by the TDE to uniquely identify Scenarios and must be preserved if the modified or extended Project is ever to be imported back into the TDE. These files contain the XML instructions that implement the Actions defined by the Scenario.

An example of the XML contents of an AutomationClientScenario-###.XML file that represents the Load DynamiX sample project CIFS-SMB Full Duplex Echo which contains only two Actions

Open SMB TCP Connection

Protocol	Name	Name	Value
SMB	Open SMB TCP Connection	Input	
SMB	Echo	Destination IP Address	String 172.16.244.1

Echo

Protocol	Name	Name	Value
SMB	Open SMB TCP Connection	Input	Connection Handle Default 1419 Repeat Count 24000
SMB	Echo	Response Handlers	Completion Status OnSuccess OnFailure

```

<scenario version="22.6">
  <actions>
    <action name="[Open CIFS/NFS TCP Connection]" protocol="SMB">
      <parameters>
        <parameter name="Destination IP Address" value="172.16.244.1" />
        <parameter name="Output Handle" value="1: SMBConnectionHandle" />
      </parameters>
    </action>
    <action name="[Echo/Ping]" protocol="SMB">
      <parameters>
        <parameter name="Bytes Total" value="1419" />
        <parameter name="Repeat Count" value="24000" />
      </parameters>
    </action>
  </actions>
</scenario>

```

Notice that the XML representation seems to be missing some inputs that are present in the Open SMB TCP Connection Action. This is on purpose. The missing values in the XML representation are the default values for these fields. The Load DynamiX TDE optimizes its XML output by only sending an XML value for a field that has been set to something other than the default value.

Extending Projects

You can see that to change the number of bytes per echo command or the repeat count would be a trivial replacement of either 1419 or 24000 in the XML content above. To add additional Actions to the scenario requires that the XML for those Actions be inserted into the appropriate location within this XML file. For example, if it were necessary to add 3 Actions to this Scenario: Negotiate, Session Setup and Session Logoff:

```

<action name="[Negotiate]" protocol="SMB" />
<action name="[Session Setup/Authenticate]" protocol="SMB">
  <parameters>
    <parameter name="Output Handle" value="4: SMBSessionHandle" />
  </parameters>
</action>
<action name="[Session Logoff]" protocol="SMB" />

```

into the appropriate locations in the AutomationClientScenario-###.XML file as shown below. It is important to notice in the XML above that the Negotiate and Session Logoff Actions are represented

by one line of XML because all of the default parameters are used in these Actions.

```
<scenario version="22.6">
  <actions>
    <action name="[Open CIFS/NFS TCP Connection]" protocol="SMB">
      <parameters>
        <parameter name="Destination IP Address" value="172.16.244.1" />
        <parameter name="Output Handle" value="1: SMBConnectionHandle" />
      </parameters>
    </action>
    <action name="[Echo/Ping]" protocol="SMB">
      <parameters>
        <parameter name="Bytes Total" value="1419" />
        <parameter name="Repeat Count" value="24000" />
      </parameters>
    </action>
    <action name="[Negotiate]" protocol="SMB" />
    <action name="[Session Setup/Authenticate]" protocol="SMB">
      <parameters>
        <parameter name="Output Handle" value="4: SMBSessionHandle" />
      </parameters>
    </action>
    <action name="[Session Logoff]" protocol="SMB" />
  </actions>
</scenario>
```

Once the desired changes have been made, the enhanced Project can be executed using the process described below.

Creating Scenarios from Scratch

Creating a Scenario from scratch is more complicated than extending an existing Project but it is possible. Creating Scenarios from scratch requires that the appropriate AutomationClientScenario-####.XML and AutomationServerScenario-####.XML files be created containing the syntactically correct XML content that represents the desired Scenario. Starting with an existing AutomationConfig.XML file created by the TDE, the new Scenarios can be added, integrated into that file.

LdxCmd.exe

This command is located in the Load DynamiX installation directory. This is not in the Project directory but the program installation directory which is typically

C:\Program Files (x86)\Load DynamiX\Load DynamiX TDE (see [Product Installation](#) for default locations).

Currently, the Project folder (the file location provided by /Project on the command line), is only required for a limited number of Automation operations (/generate and /upgrade). Otherwise, it is sufficient **and recommended** to specify only the /config: option to /start, /stop, /compile and /verify.

```
LdxCmd.exe /portstatus:<Appliance IP Address>
LdxCmd.exe /upgrade /Project:<ProjectFile> [/AppLog:<LogFileName>] (/update is also valid and has the same behavior as /upgrade)
LdxCmd.exe /generate /Project:<ProjectFile> [/out:<Configuration Folder>] [/upgrade] [/Force]
[/copydatacontent:all,datafolderonly] [/AppLog:<LogFileName>]
LdxCmd.exe /compile /Project:<ProjectFile> | /config:<ConfigurationFile> [/upgrade]
[/AppLog:<LogFileName>]
LdxCmd.exe /verify /Project:<ProjectFile> | /config:<ConfigurationFile> [/upgrade]
[/AppLog:<LogFileName>] [/statmode:stdout|csv|stdout, csv|off] [/statinterval:<seconds>]
[/out:<ResultsFolder>]
LdxCmd.exe /start /Project:<ProjectFile> | /config:<ConfigurationFile> [/upgrade]
[/AppLog:<LogFileName>] [/statmode:stdout|csv|stdout, csv|off] [/statinterval:<seconds>]
[/out:<ResultsFolder>]
LdxCmd.exe /stop /Project:<ProjectFile> | /config:<ConfigurationFile> [/AppLog:<LogFileName>]
LdxCmd.exe /stop /port:<Appliance IP Address>:<PortNumber>
LdxCmd.exe /help
```

Note:

Optional command line parameters are specified in **Teal** color.

The or bar "|" means use either the input to the left of the bar or the input to right but not both.

LdxCmd produces very limited output (to files or to the console/screen) unless options like /Statmode and /Applog are used. If no output is seen and no command line prompt appears, LdxCmd is likely executing.

Command line action to be performed:

- /start - start the test Project
- /verify - verify the test Project
- /stop - stop the running test Project or the Project that is running on the specified <Appliance IP Address>:<PortNumber>
- /compile - compile the test Project
- /portstatus - get port status for Appliance at the specified IP address
- /generate - generate the files necessary for Automation
- /upgrade - convert the specified Project to the current release level
- /help - print the command line help text above to the screen

Where the required and/or optional command line input are:

Appliance address parameter

<Appliance IP Address>. Specifies the IP address of the Appliance being accessed by the /Portstatus and /Stop command line actions.

Project file parameter (**only required for /upgrade and /generate, and not needed for /start,**

/stop, /compile and /verify when the /config: path to AutomationConfig.XML is provided)

/Project:<ProjectFile>. Specifies test Project file path (example /Project:c:\folder\file.swift_test)

Load DynamiX recommends that the /Project input only be used for /upgrade: and /generate: actions and that end users /generate: their Projects first and then use only the /config: input from that point forward to execute the Project using /verify: and /start:.

Configuration file parameter

/Config:<ConfigurationFile>. Specifies configuration file path (example /config:c:\folder\AutomationConfig.XML)

Upgrade (or Update)

/upgrade. upgrades the Project to the current version. If your Project was created in an older version of the TDE and you are trying to run it with a newer version, you will get an error message indicating that the test was created in a different version. Please use /upgrade if you want to run a Project created using a different software version.

Out

/Out:<Folder>. The /Out: parameter has different uses depending on the command line actions being used. If the action is /Generate then the /Out: parameter specifies the target folder for the output of the /Generate action (AutomationConfig.XML for example). If the action is /Verify or /Start then the /Out: parameter specifies the target folder for the Results of the /Verify or /Start - the statistics and log files generated (Client Port <X>(<IP Address> port >Y>.log for example). If /Out is not specified on the command line, the default folder name and location are used.

Force

The /Force option forces LdxCmd to create the <Automation Folder> if it is not present.

Applog

/AppLog:<LogFileName>. The /Applog option specifies the path to and the name of the log file for LdxCmd failures and Automation process steps. By default this logging feature is disabled and is only enabled when specified on the command line. Applog files are overwritten whenever used so the previous contents of any file used for Applog output will be replaced. /Log: may be used in place of /Applog: with the same results.

Statmode

/statmode:stdout|csv|stdout,csv|off. Controls real-time statistics output behavior during runtime for those statistics that were added to the AutomationConfig.XML file (see Automation_stats.XML discussion below). Default behavior (**/statmode:stdout**) is for added statistics to be output to Stdout at real-time during Project execution. If the user wishes to change the output target or disable output, the **/statmode:** command line option must be used. **/statmode:off** (disable real-time statistics output at runtime), **/statmode:csv** (output added statistics to a csv file at runtime) or **/statmode:stdout,csv** (output added statistics to both at runtime). csv mode is recommended for statistics output during runtime.

Statinterval

/statinterval:<seconds>. Controls statistics output behavior regarding how often added statistics (see Automation_stats.XML discussion below) are output in real-time when executing a Project via Automation. Default behavior is to output statistics every 1 second. Valid input to /statinterval are integers in the range 1..300.

Port

/Port:<Appliance IP Address>:<PortNumber>. Specify the Appliance IP address and Port Number when using the /stop command. IP Address is the management IP address of the Appliance. Valid Port values range from 0 to 7 on a Load DynamiX Models 3000/3108/5108S/5108T/6208/U1044, 0 to 3 on a Load DynamiX Model 6204/U1022 and 0 to 1 on a Load DynamiX Models 5000/5102/6202/6202E.

Copydatacontent

/Copydatacontent: All, Datafolderonly. Is used only with the /Generate command to (optionally) copy the contents of all real files in Data File System resource into the AutomationConfig folder. The options are All to copy all file into the \AutomationConfig\Data folder or Datafolderonly to only copy those files that reside in the Project Data folder.

Note: command line actions and optional entries may be specified either in Upper or lower case

Examples:

All command line options begin with a "/" or "--"

```
C:/Program Files/Load DynamiX/Load DynamiX TDE/LdxCmd.exe /Start /Config:C:/Load DynamiX/My Projects/CIFS-SMB_Full_Duplex Payload/AutomationConfig/AutomationConfig.XML /statmode:off /statinterval:3
```

```
C:/Program Files/Load DynamiX/Load DynamiX TDE/LdxCmd.exe /Compile /Project:C:/Load DynamiX/My Projects/CIFS-SMB Write/CIFS-SMB Write.swift_test /upgrade
```

```
C:/Program Files/Load DynamiX/Load DynamiX TDE/LdxCmd.exe /Generate /Force /Project:C:/Load DynamiX/My Projects/CIFS-SMB Write/CIFS-SMB Write.swift_test /Out:C:/Load DynamiX/My Projects/CIFS-SMB Write/AutomationConfig /AppLog:.\\Load DynamiX.log
```

```
C:/Program Files/Load DynamiX/Load DynamiX TDE/LdxCmd.exe /PortStatus:192.168.1.10
```

```
C:/Program Files/Load DynamiX/Load DynamiX TDE/LdxCmd.exe /stop /port:192.168.1.10:3
```

Note: On Unix or Linux systems, the command line syntax for LdxCmd.exe is the same as Windows command line:

Example: C:/Program Files/Load DynamiX/Load DynamiX TDE/LdxCmd.exe --PortStatus:192.168.1.10 --AppLog:/var/log/Load DynamiX.log

The PERL script used in the discussion below can be found in the **{Scripts}** folder on the system on which the TDE is installed. See the [Product Installation section](#) for a list of where the **{Scripts}** folder can be found.

Automation using the TCL scripting language

LdxCmd can be called from any scripting language. LdxCmd runs on Windows, but Mono (www.mono-project.com) can be used to run LdxCmd on Linux.

Scripting languages can be used to sequence the tests and change the required parameters. Here are a few examples using TCL.

NOTE: ActiveState TCL is available from www.activestate.com/activetcl/

In order to run any TCL script, make sure the path is configured correctly. By default TCL shell can be found at c:\Tcl\Bin\

1. Run a test.

This is the simplest example of automation. User would like to run suite of tests over the weekend or overnight. The simple tcl script for running a Load DynamiX Project is as follows

```
puts "Executing 4x4 SMB2 Connection Rate \n"
puts [exec "C:/Program Files/Load DynamiX/Load DynamiX TDE/LdxCmd.exe" /Start "/Config:C:/ Load
DynamiX/My Projects/SMB2 4x4 Connection Rate/AutomationConfig/AutomationConfig.xml"]
```

Copy the script to any notepad or Textpad and Save it as SMB24x4ConnRate.tcl in C:\Automation folder. Then run that .tcl file using TCL shell as follows

```
C:>tclsh85.exe c:\Automation\SMB24x4ConnRate.tcl
```

2. Run a test N times

```
set N 10

for { set i 1 } { $i <= N } { incr i } {
    puts "Executing CIFS-SMB Write #$i"
    puts [exec "C:/Program Files/Load DynamiX/Load DynamiX TDE/LdxCmd.exe" /Start "/Config:C:/Load
DynamiX/My Projects/CIFS-SMB Write/AutomationConfig/AutomationConfig.xml"]
}
```

Copy the script to any notepad or Textpad and Save it as Multiple Run.tcl or something.tcl in C:\Automation folder. Then run that .tcl file using TCL shell as follows

```
C:>tclsh85.exe c:\Automation\Multiple Run.tcl
```

Very important Note: In the examples above, the paths (highlighted in Blue) for the .exe, Project file and the Config file need to be changed according to where the Load DynamiX TDE is installed and where the test being run is located.

Automation using the PERL scripting language

The Perl statements below are an example of how to use the PERL programming language to automate executing Load DynamiX projects. Filling in {UserLoginName} , {InstallationFolder} (which is by default "Load DynamiX TDE") and the Perl variables: \$TestName, \$AutomationDir and \$StartCmd below is all that is necessary to execute the Project named by \$TestName.

NOTE: ActiveState Perl is available from <http://www.activestate.com/activeperl/downloads>

```
#!/Perl
#####
## Run a pre-configured Load Dynamix project for N number of times.

#####
## Variables to CHANGE
my $NumberofTimes = 3 ;

#####
## Required Project and Test information to configure
##
## FILL IN the 5 variables below (determined by where the product is installed and which Project is
## to be run)
##
## $SwiftCmd - full path to LdxCmd.exe
## $ProjectDir - where Project files are stored
## $TestName - the name of the Project to execute
## $AutomationDir - location where Automation files are Generated (default = AutomationConfig
```

```

folder in the Project folder)
##   $StartCmd - whether to execute fully (/Start) or just to Validate (/Verify)
##
##   {InstallationFolder} = the name of the folder that was selected at Install time (default =
Load DynamiX TDE)
##   {UserLoginName} = user id of the User who installed the TDE
##
## Examples:
##   WindowsXP: my $ProjectDir = '"C:\Documents and Settings\{UserLoginName}\My Documents\Load
Dynamix\My Projects"' ;
##   Windows7:   my $ProjectDir = '"C:\Users\{UserLoginName}\Documents\Load Dynamix\My Projects"' ;
#####

## The following paths may vary depending on the version of Windows that is used
##       $SwiftCmd - full path to LdxCmd.exe
##       $ProjectDir - where Project files are stored
## The TestName variable will change depending on which Project is to be executed

## On a Windows 7 system:

my $SwiftCmd      = '"C:\Program Files\Load Dynamix\{InstallationFolder}\LdxCmd.exe"' ;
my $ProjectDir = '"C:\Users\{UserLoginName}\Documents\Load Dynamix\My Projects"' ;
my $AutomationDir = $ProjectDir . "/" . $TestName . "/AutomationConfig" ;

## The TestName variable will change depending on which Project is to be executed

my $TestName      = '"CIFS-SMB Write"' ;

##### Set StartCmd to: "/Start" (for normal test runs) -or- "/Verify" (for validation only runs)

my $StartCmd      = "/Verify" ;
# my $StartCmd      = "/Start" ;

#####
## Fixed Variables

my $TestConfigFile = $ProjectDir . "/" . $TestName . "/" . $TestName . ".swift_test" ;
my $AutomationConfigFile = $AutomationDir . "/AutomationConfig.xml" ;

## Note: Running, Validating, or Compiling a test in the Load Dynamix TDE (GUI) or via LdxCmd
will NOT rebuild or overwrite the AutomationConfig.xml file.
## When needed, select the "Generate Automation Files" from the Project menu, or use the
"LdxCmd /Generate" option to rebuild the AutomationConfig.xml file.

#####
## Begin

print
"\n\n\n\n=====\\n\\n\\n" ;
## Confirm validity of the original project with conformance (/upgrade) & syntax (/compile) checks
##
print "Performing pre-test /upgrade & /compile operations for conformance & syntax checks on
the project: $TestName \\n\\n" ;
print "Load Dynamix Command: $SwiftCmd /upgrade /compile /project:$TestConfigFile \\n" ;
print           `"$SwiftCmd /upgrade /compile /project:$TestConfigFile` ;

## Generate fresh Automation files -- (this step is optional - ie. for demonstration purposes)
##
print "\\n\\n" ;
print "Generating fresh Automation files for the project: $TestName \\n\\n" ;
print "Load Dynamix Command: $SwiftCmd /generate /force /project:$TestConfigFile
/out:$AutomationDir \\n" ;
print           `"$SwiftCmd /generate /force /project:$TestConfigFile
/out:$AutomationDir` ;

```

```

print "\n\n" ;

## Main Loop

for (my $loop=1; $loop<=$NumberofTimes; $loop++) {
    print "\n-----\n" ;
    print "Preparing Test #\$loop (of \$NumberofTimes): $TestName \n\n" ;
    ##
    ## Here, you can alter different aspects of the test configuration.
    # For example, you can increment the load height, or cycle through different scenarios,
    IP's, User Parameter files, toggle action parameters, etc.
    ##
    print ". Configuration Changes for Test #\$loop: (none) \n\n" ;
    ##

    ## EXECUTE the Test now using the "AutomationConfig.xml" file
    print "Executing Test #\$loop (of \$NumberofTimes): $TestName \n\n" ;
    print "Load DynamiX Command: $SwiftCmd $StartCmd /project:$TestConfigFile
/config:$AutomationConfigFile \n (please wait)" ;
    print `\$SwiftCmd $StartCmd /project:$TestConfigFile
/config:$AutomationConfigFile` ;

    print "\nFinished Test #\$loop (of \$NumberofTimes): $TestName \n\n" ;
    ## Sleep 3 secs, then run another test.
    sleep(3) ;
}

## Finished Main Loop
##
print "\n\n" ;
print
-----
-\n" ;
print "Finished $NumberofTimes (of \$NumberofTimes) Test runs of project: $TestName \n" ;
print
-----
-\n\n\n" ;

```

Automation Scripts

The **{Scripts}** folder on the system on which the TDE is installed contains additional scripts that demonstrate how to Automate Load DynamiX Projects.

Scripts Documented in OnLine Help

- Perl Execute Test in a Loop.pl

Additional Automation Scripts

- Tcl Execute Test in a Loop.tcl
- Tcl Substitute User Parameters.tcl

Automation and Linux

Load DynamiX automation can be used in Linux environments using software available through the Mono Project. Mono software is available from <http://www.go-mono.com/mono-downloads/download.html>. Load DynamiX recommends Mono

version 2.6.x or higher and SQLite 3.6.x or higher. SQLite can be found at [SQLite web site](#). To verify the version of Mono installed, type `mono -V` at a command prompt. To verify the SQLite version, type `sqlite3 ?` at a command prompt.

Linux How To Steps

- From the Mono folder on your WorkStation (see [Product Installation section](#)) or from the Load DynamiX Beta website LinuxAutomation folder (see [Load DynamiX TDE and Appliance Introduction section](#)), extract the contents of the LdxCmd Linux.tgz onto your Linux system. Load DynamiX Automation_Stats.XML (described below), Help and Quick Start documents are provided in the Load DynamiX_Docs folder. The TCL and PERL scripts mentioned in Help are delivered in the Sample_Automation_Scripts folder. The files in this .tgz are:

Name	Size
..	
lib32_freebsd	
lib32_linux	
lib64_freebsd	
lib64_linux	
Ports	
Sample_Automation_Scripts	
SwiftTest_Docs	
Actions.zip	661,559
SwiftCmd.exe	64,512
Ionic.Zip.dll	462,336
SwiftTest.API.dll	2,959,360
SwiftTest.Compatibility.dll	140,800
SwiftTest.SharedAPI.dll	184,832
System.Data.SQLite.dll	159,232
SwiftCmd.exe.config	1,696
System.Data.SQLite.dll.config	97
SwiftCmd	2,734
System.Data.SQLite.XML	184,839

Total 7 folders and 4,821,997 bytes in 11 files

- Make sure that all *.dll files have a ".dll" file extension in lower case
- Make sure that the user that will be running the Load DynamiX commands on Linux has RW permissions to the /var/log directory
- Copy the Load DynamiX Project directory contents to a directory on your Linux machine
- Make sure that the user's environment variables include the path to the directory that contains the contents of the LdxCmd Linux.tgz file, the path to the Mono installation and the path to the directory that contains the Load DynamiX Project directory
- Execute LdxCmd as described above using Linux-style paths ("/" instead of "\")

Automation Statistics

Client Statistics

Users can also configure AutomationConfig.XML to receive realtime statistics on Stdout. Users need to create a Project in the GUI and successfully compile it so that they can use the Generate Automation File to create the AutomationConfig.XML file. Once the AutomationConfig.XML file is created, the following XML tags can be added anywhere in the AutomationConfig.XML file as long as it is within

the outermost XML start and end tags to receive statistics on Stdout.

```
<StatsOutput>
  <Port>
    <PortName>Client Port 0</PortName>
    <StatElement>
      <StatType>Load Status</StatType>
      <StatFieldList>
        <StatField>SC_ATTEMPTS</StatField>
      </StatFieldList>
    </StatElement>
  </Port>
</StatsOutput>
```

Currently, the following are supported in <StatType> statements:

- ARP
- CDMI
- DCBX
- Load Status
- Network Status
- FC Network Status
- FC Session
- Ethernet
- IPv4
- IPv6
- TCP
- TCPC (Connections)
- TCPS (State)
- TCPF (Flags)
- TCPR (Reconnects)
- SMB
- SMB2
- RPC
- KERBEROS
- HTTP
- HTTP Authentications
- iSCSI
- SCSI
- MSRPC2
- OpenStack_Swift
- Priority Flow Control
- SSL/TLS
- SSL/TLS Connections
- TCP6
- TCP6C (Connections)
- TCP6S (State)
- TCP6R (Reconnects)
- TCP6F (Flags)

Here is an example of how to configure statistics collection and output to Stdout:

```
<StatsOutput>
  <Port>
    <PortName>Client Port 1</PortName>
    <StatElement>
```

```

<StatType>Load Status</StatType>
<StatFieldList>
  <StatField>SC_ATTEMPTS</StatField>
  <StatField>SC_SUCCEEDS</StatField>
  <StatField>SC_FAILS</StatField>
  <StatField>SC_ABORTS</StatField>
  <StatField>AC_ATTEMPTS</StatField>
  <StatField>AC_SUCCEEDS</StatField>
  <StatField>AC_FAILS</StatField>
  <StatField>AC_ABORTS</StatField>
</StatFieldList>
</StatElement>
<StatElement>
<StatType>SMB</StatType>
<StatCommand>ALL</StatCommand> <!-- Optional ALL or specific Command Name -->
<StatFieldList>
  <StatField>ACT_ATTEMPTS</StatField>
  <StatField>ACT_FAILS</StatField>
  <StatField>ACT_SUCCEEDS</StatField>
  <StatField>ACT_ABORTS</StatField>
  <StatField>ERRORS_TOTAL</StatField>
</StatFieldList>
</StatElement>
</Port>
</StatsOutput>

```

Once AutomationConfig.XML is configured for statistics output, LdxCmd will write the desired statistics onto stdout at one second intervals.

Note: The complete list of statistics that can be collected is available in the file **Automation_Stats.XML**, located in the Load DynamiX **{Scripts}** folder (see [Product Installation](#) for default locations).

Using Automation on a Linux System

If you intend to automate the execution of a Project developed with Load DynamiX on Linux, Load DynamiX recommends that you install Mono version 2.6.x or higher and SQLite version 3.6.x or higher on the Linux system. The file LdxCmd.Linux.tgz contains the Load DynamiX files necessary to run on Linux (Mono and SQLite are not provided). The file LdxCmd.Linux.tgz can be found in the installation folder of the TDE (C:\Program Files (x86)\Load DynamiX\Load DynamiX TDE\Mono).

Load DynamiX Enterprise Integration (Common Project Library)

Common Project Library gives Testers the ability to

- Browse TDE Projects on an LDX-E Server from the TDE UI,
- Make modifications and easily save the TDE Project between TDE and LDX-E
- Easily customize the LDX-E Workloads.

The Load Dynamix TDE is one of several mechanisms that can be used to execute Load DynamiX Projects on Load DynamiX Appliances. Other mechanisms are Automation (described in this Appendix), the Load DynamiX API and the Load Dynamix Enterprise Server. This section describes the integration between the TDE and the Load DynamiX Enterprise Server.

The Load DynamiX Enterprise Server is and hardware and software solution n which Load DynamiX

Projects can be cataloged, executed and results logged. The Projects that are stored and executed by the Load DynamiX Enterprise Server are compatible at the lowest (XML) level with TDE Automation so it is possible to share Projects between the two environments.

The TDE provides four functions in the Project menu targeted at integration with Load DynamiX Enterprise installations. The four functions are:

LDX-E Server Login: Create a connection between the TDE and an LDX-E server by logging in. The server IP address and user credentials (email address and password) must be known. This is the first step required when integrating with an LDX-E Server. Once logged into an LDX-E Server, the TDE will remember that the connection has been established

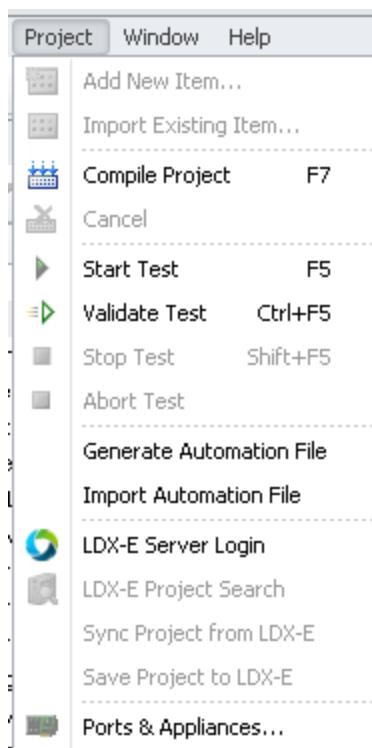
LDX-E Project Search: Once logged in, review the set of LDX-E Projects (and Workload Templates) that are available to copied (synced) into the TDE. LDX-E Project Search shows all of the Projects that are accessible to the user that is used to log in to the LDX-E Server. Note that the user may not have access to all of the Projects on the LDX-E server so the set of Projects displayed may only be a subset of the Projects that are cataloged on the LDX-E Server.

Sync Project from LDX-E Server: Once Logged in, Sync (import) the current TDE Project from the LDX-E server if it has been saved to the LDX-E server. This function applies to the currently open Project in the TDE. If the currently open Project has not been Saved to the LDX-E Server or an LDX-E Server Login has not been performed then this function will be non-operational (grayed out). LDX-E Workload Templates may also be imported into the TDE. Imported Workload Templates have characteristics that are slightly different from those of an imported Project. Imported Workload Template Resources are mostly Read-Only. The TDE Scenario Editor will highlight Workload Template Resources in Red that should not be changed in the TDE because they are controlled by an LDX-E Workload or Testbed. Imported LDX-E Projects and Workload Templates maintain a relationship to the Project or Workload Template on the LDX-E server such that changes made on the TDE can be Saved back to the LDX-E server. If a Save As of an imported Project is done on the TDE, that relationship vanishes and the "Saved As" Project would need to be saved as a different Project on the LDX-E server. If a Save As is done on an imported Workload Template on the TDE, it becomes a Project not a Workload Template.

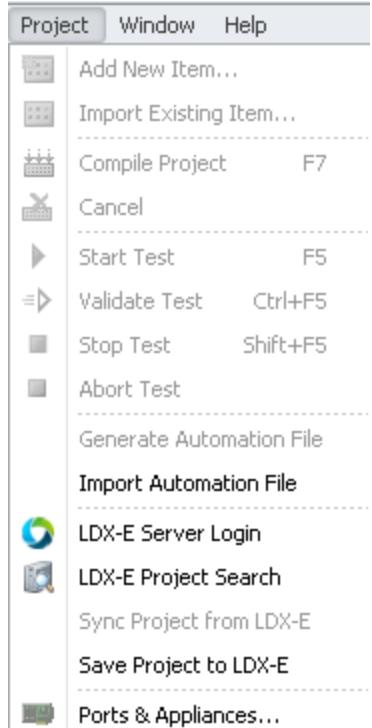
Save Project to LDX-E Server: Once Logged in, Save the current Project to the LDX-E server. Save Project to LDX-E Server is not operational until an LDX-E Server Login has been successfully completed.

User Interface

Project Menu:
(before successful login, all functions but Login disabled))

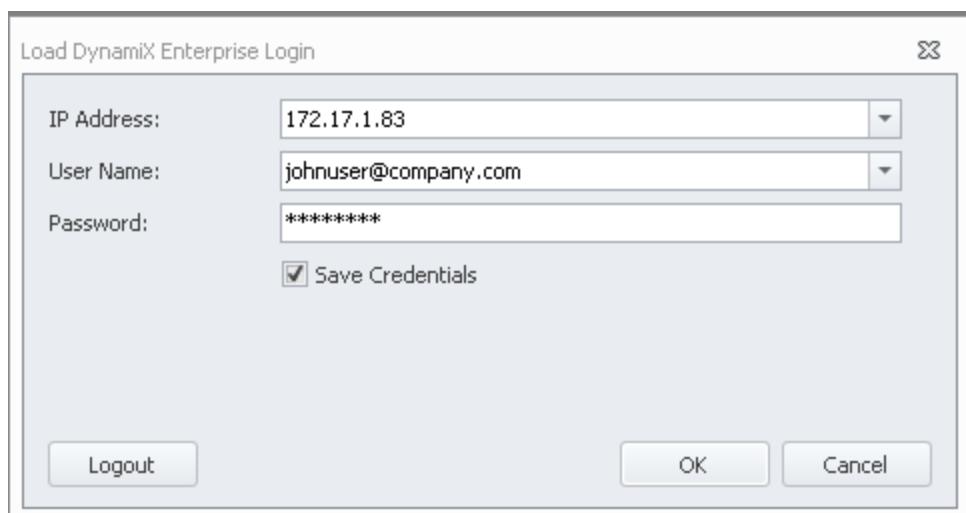


(after successful login, current Project has not been saved to LDX-E Server)



Function Interfaces

LDX-E Server Login: The TDE user must provide the IP address of the LDX-E Server, and the user credentials for a user that has been created on that Server.



LDX-E Project Search: All LDX-E Projects that are accessible to the user that is logged in are displayed through this interface. The Project name can be typed into the search bar or the user can scroll through the Projects that are displayed in a multi-page tabular form.

Load DynamiX Enterprise Project Search			
Name	Status	Details	
▶ Simple MPIO_Weighted Paths_R42		...	
Simple MPIO_RoundRobin_R42		...	
Simple MPIO_Fail Over_R42		...	
SMB2 Full Duplex Payload Compound Actions_R42		...	
SMB2 Formula Sample orig	Saved	...	
SMB2 Formula Sample	Saved	...	
SMB threads events async dv_R42		...	
SMB SMB2 NFSv3 HTTP data verify all datacontent type Int Server test_R42		...	
Performance Multi_Protocol Stress Test _IPv4_IPv6_R42		...	
LDX 5100 iSCSI Full Duplex Payload_R40	Saved	...	
LDX 5100 iSCSI Actions Per Sec (NOP)_R40		...	
LDX 5100 TCP Connections for Rate iSCSI (IPv4)_R40	Saved	...	
LDX 5100 TCP Connections for Rate SMB2 (IPv4)_R40		...	
LDX 5100 TCP Connections for Rate SMB (IPv4)_R40		...	
LDX 5100 TCP Connections for Rate NFSv4.1 (IPv4)_R40		...	
LDX 5100 TCP Connections for Rate NFSv4 (IPv4)_R40		...	
LDX 5100 TCP Connections for Rate NFSv3 (IPv4)_R40		...	

Buttons at the bottom include Prev, Next, Page: 1, Save, and Cancel.

Highlight a Project and click Details to get this summary

Load Dynamix Enterprise Project Search



SMB2 Formula Sample orig

Name	Value
Name	SMB2 Formula Sample orig
Project ID	563a5edd16ae12853f000053
Project Owner	john user
Protocols	
[0]	SMB2
[1]	SWT
Tags	
Created	11/4/2015 11:39:09 AM
Lat Run	
Number of Runs	0

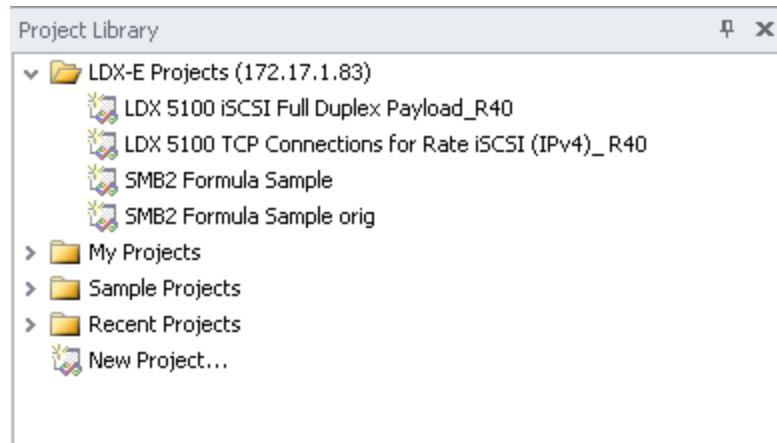
Save Cancel

Highlight a project and click **Save** to import a Project from the LDX-E Server into the TDE.

The TDE will automatically create a new folder to contain Projects that have been imported from this specific LDX-E server. Any login to a new/different LDX-E Server will likewise cause the TDE to create another new folder for Projects imported from that LDX-E Server. Click

Cancel

to close the Search window.



When a Project is Open in the TDE and an LDX-E Server has been logged into, the Sync Project from LDX-E and Save Project to LDX-E functions are operational.

Save Project to LDX-E generates the compatible (Automation) form of the Project and copies it into the Project repository for the user that is logged in. Once the Project has been Saved to the

LDX-E Server then it can be Imported (Sync'd) from the LDX-E Server to the TDE. The Project has been Sync'd will be stored in the LDX-E Projects folder created by the TDE.

Projects created on the TDE and Saved to the LDX-E Server can be executed after assignment to the appropriate Test Bed. Projects Sync'd from an LDX-E Server to the TDE can be executed on the appropriate Appliance once Port assignments have been made.

For more information on the LDX-E product family see <http://www.Load DynamiX.com/products>.

Appendix: Client Side DFS Support

Appendix: Client Side DFS Support

Client Side DFS Support

The Windows Distributed File System (DFS) defines a method used to organize distributed file shares into a single file system using a single namespace. DFS enables the location of a file to be transparent to the end user, and enables file redundancy to ensure data remains available if a server is removed from service.

The DFS name space may be domain-based, meaning that the name space is contained on one or more Windows Servers connected to a Windows Domain and known to Windows Active Directory, or standalone-based, where the DFS Name Space resides on a single Server. Domain-based DFS adds redundancy to DFS.

The Load DynamiX client supports Standalone DFS Root access.

DFS contains two main components; the DFS namespace and DFS replication. The Load DynamiX appliance currently supports client side DFS support for the Standalone DFS namespace. DFS replication is unsupported.

Standalone DFS

A Standalone DFS root identifies files that may be distributed throughout the domain. Unlike Domain-based DFS, Standalone DFS does not support redundant DFS Root servers. The Load DynamiX client uses the following format for a Standalone DFS Referral Request:

`\<Server_IP>\<path>`

Where:

`<Server_IP>` is the IP address of the domain controller,

`<path>` is the path pointing to the Standalone DFS Root directory.

Note: Samba servers can return either a share or a file in response to a DFS Referral Request. Additionally, a Standalone DFS Root returns only one share or file name in response to a DFS Referral Request.

Configuring DFS

DFS is configured after the client connects to a server operating as a Stand Alone DFS Root Server. The steps to use DFS are identical for CIFS-SMB and SMB2 .

Use the following steps to configure Standalone-based DFS client-side support. Note that DFS Referral Requests to a Standalone DFS Root can be for either a share (directory) or file. Connect to the Standalone DFS Root Server using the three following commands:

Issue an **Open SMB TCP Connection** Request:

Specify the Standalone DFS server IP address

Issue a **Negotiate** Request.

Issue a **Session Setup** Request:

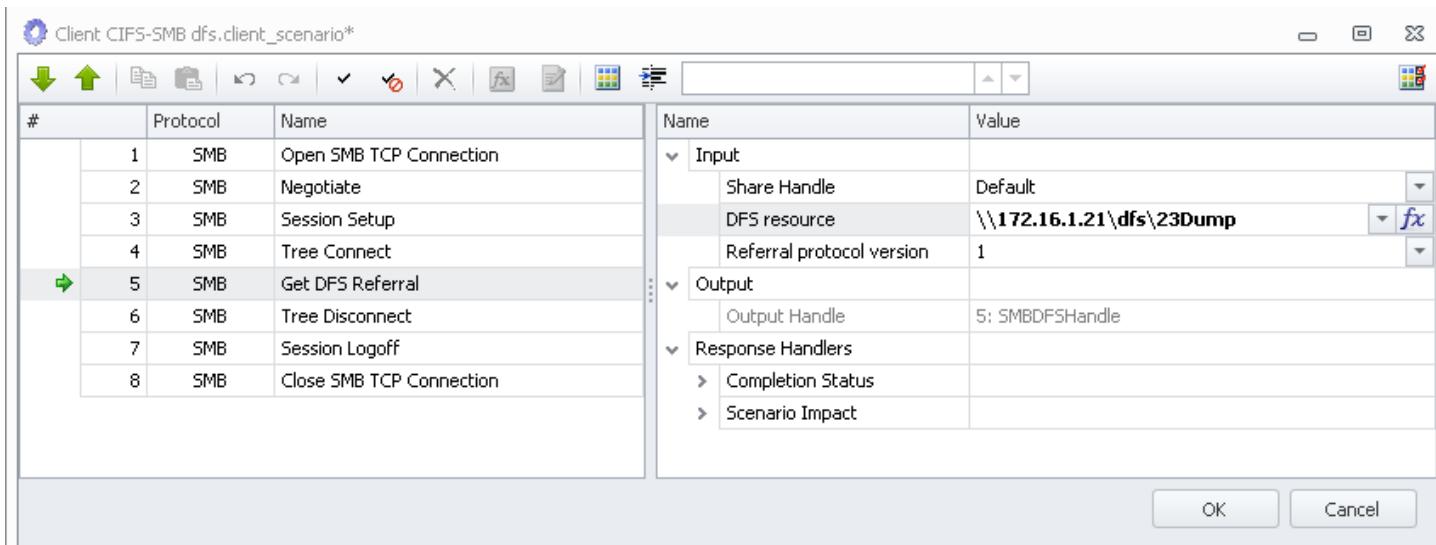
Specify NTLM authentication. Enter a valid User Name and Password.

Configure the **Tree Connect** command.

Specify the Tree connect in the form

“\\<Server_IP_Address>\\<DFS_Root_Directory>”.

Configure the **Get DFS Referral Request**.



Note: The Load DynamiX Appliance simulates the **Get DFS Referral** Request. It is necessary to use the same format as what is used in a Standalone DFS server response to the **Tree Connect** Request. You should obtain a packet trace of a DFS Referral Response and use the path specified in the response. If you want to refer to a second or subsequent server or file in the DFS Referral Response, use an offset (first share reference=0, second share reference=1, etc.) in the DFS Referral request.

Specify the DFS target file path in the form \\<Server_IP>\\<Path>

Optional: specify the Max Referral protocol version supported. Version numbers may be 1, 2, 3A, 3B, or 4.

The Domain Controller responds with a link to the absolute path name using the agreed version. The response is the DFS handle referring to the link.

If a **Get DFS Referral** Request was issued for a share and access a file in the share is needed, issue a Find First Request to search for a file in the specified DFS-referred share. Issue a Find Next if the file is not returned in the Find First Response.

Issue the appropriate command to continue operation. For example you may want to open the file. You may also want to rename, copy, move or delete the file.

See [Reference: CIFS-SMB Command List](#) for the complete set of supported CIFS-SMB commands

Appendix: Scenario Control Actions

Appendix: Load DynamiX Scenario Control Actions

In addition to networked storage protocols like CIFS-SMB, SMB2, NFSv3/v4/v4.1, Fibre Channel , iSCSI, HTTP/S, and HTTP Storage, the Load DynamiX TDE provides a powerful set of Actions to help Testers control execution behavior. These Actions allow the Tester to insert loops within Projects, start and end Threads, check for Events, influence test timing by inserting delays into Scenario execution paths, and advancing or resetting the next element pointer for User Parameter files, etc.

Load DynamiX Scenario Control Actions list

- ✓  **Scenario Control Actions**
 - ✓  **Basic Flow Control**
 -  Delay Execution
 -  Begin Loop
 -  End Loop
 -  Break
 -  Continue
 -  If
 -  Else If
 -  Else
 -  End If
 - ✓  **Advanced Flow Control**
 -  Raise Event
 -  Wait for Event
 -  Begin Thread
 -  End Thread
 -  Begin Async
 -  End Async
 -  Wait For All Threads
 -  Advance User Parameters
 -  Reset User Parameters
 -  Comment
 -  Distribution
 -  Formula
 -  IO Manager
 -  Log Message
 -  Set Auto Offset
 -  Create Variable
 -  Update Variable
 -  Create Handle Variable
 -  Update Handle Variable

Execution Control Actions

Events: See Events discussion below.

Threads and Async: See [Advanced Concepts: Threads and Async Operations](#) for a detailed discussion of Thread and Async Actions.

If [Else / Else If] End If

This collection of Actions allow the Tester to execute Actions under condition control. If and End If form an execution block. Else or Else If may appear within that block to specify a set of Actions to be executed if the result of the If condition is not True. Else If allows the Actions that follow to also be executed conditionally. Nested If [Else/Else If] End If blocks are supported.

Action Inputs:

If and Else If: Mode [Condition or P(true) = K / N]

Condition: If Condition is selected in Mode and the input to the Condition field evaluates to a non-zero value then Condition == TRUE

P(true) = K / N (Probability Mode): Generates a True result K/N % of the time. (N-K)/N % of the time, it generates a False result. If K >= N then the result is always True. If N == 0 then this is an error and the Scenario will stop executing.

The following scenario illustrates nested If - Else If - End If (the Blue highlighting is because the Action Dependency  feature is enabled) and Action Indentation is present because the Indentation  feature is enabled).

The Client Port log file that results from Validation of this Scenario is:

 10	Info	12/3/2015 2:34:49 PM	Client-side high-performance stack is entering active state...
 11	Info	12/3/2015 2:34:49 PM	Scenario 1 else K= 200 N = 101
 12	Warning	12/3/2015 2:34:49 PM	Scenario 1 K= 300 N = 100

Branch Control using Status Code return and If [Else / Else If] End If

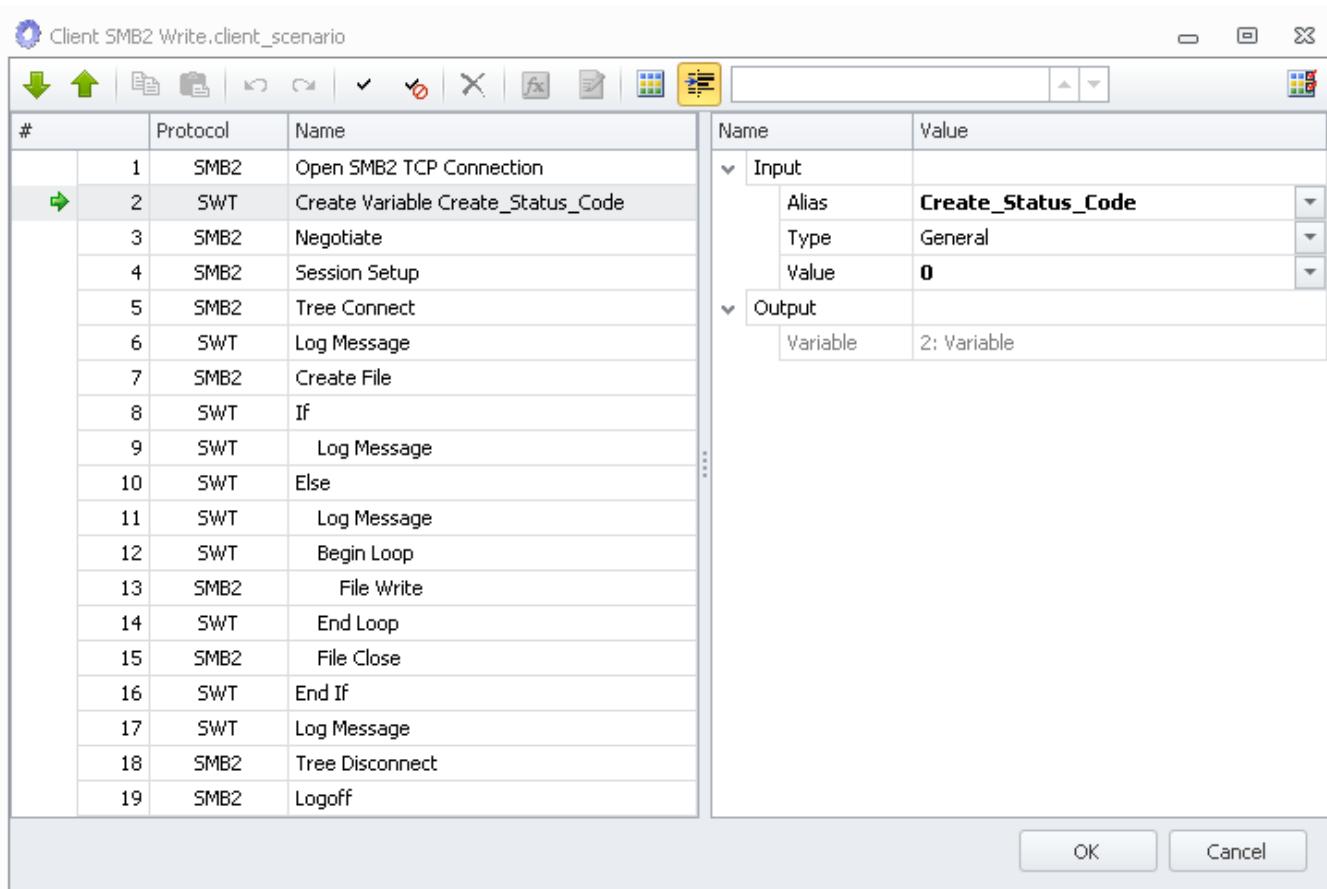
The If/Else/Else If/End If Scenario Control Actions used in conjunction with the Status Code Return mechanism can provide Branching Control within Scenarios. A Status Code Return mechanism sample SMB2 Scenario follows.

Status Code Return

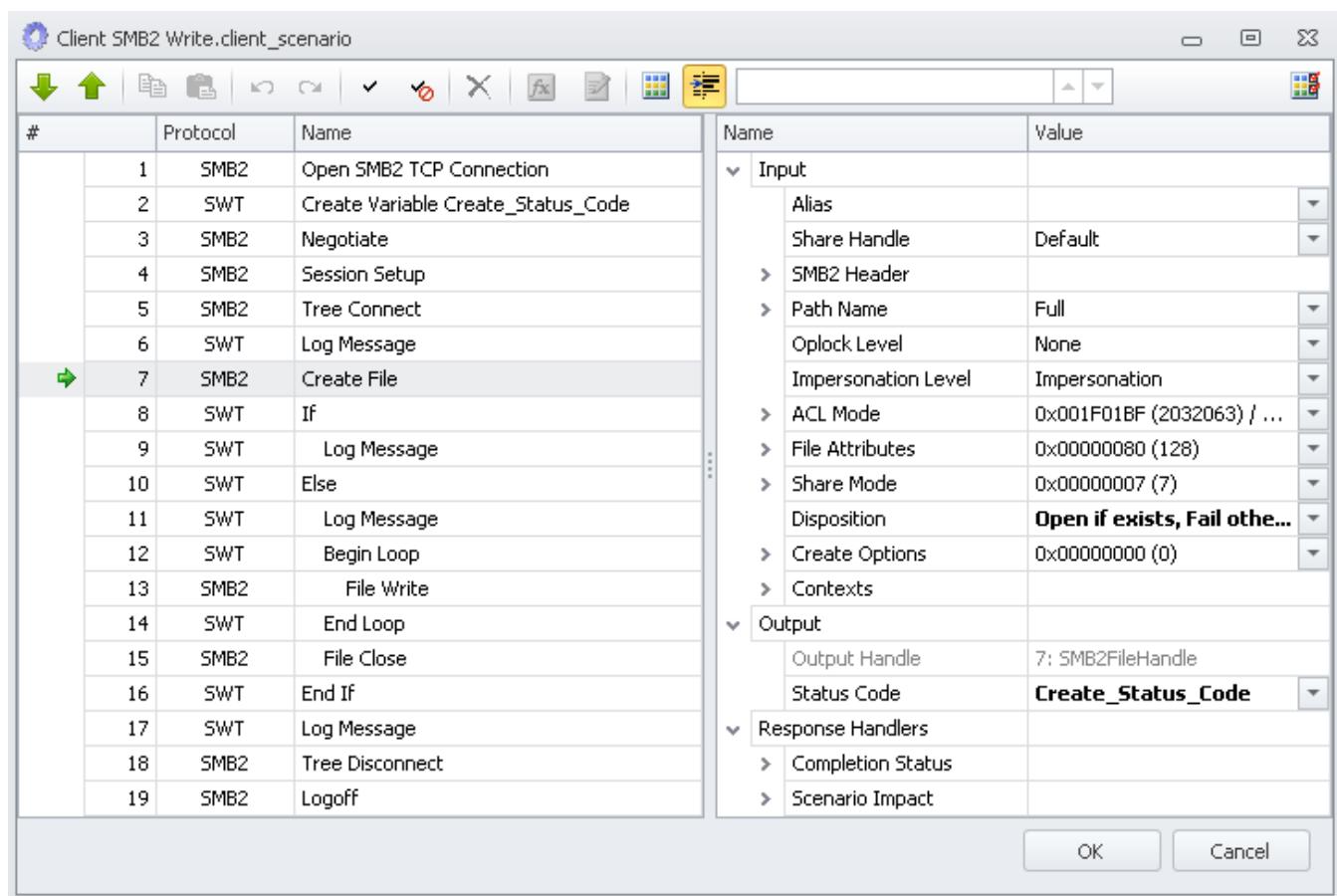
In supported Protocols, Actions that support Status Code Return will have visible in the Output

section of the Action, the input field Status Code. The input to that field **MUST** be the alias of a Variable created earlier in the Scenario. The next two screenshots show Variable creation with its Alias (in line 2) and the use of that Alias in the Status Code input. For more information on Variables and Aliases see [Advanced Concepts: Variables and Aliases](#).

Variable Creation

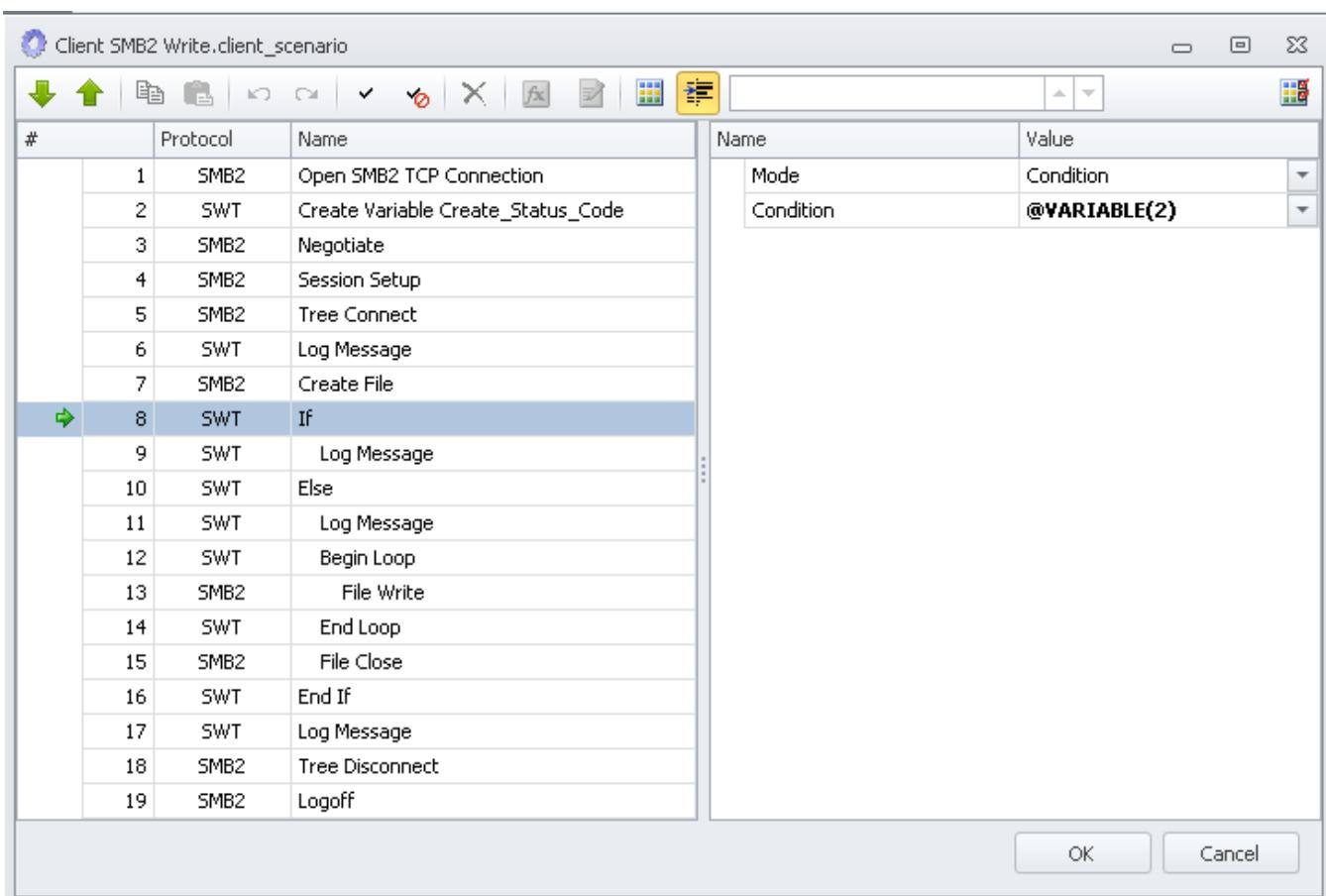


Variable Alias use in Status Code

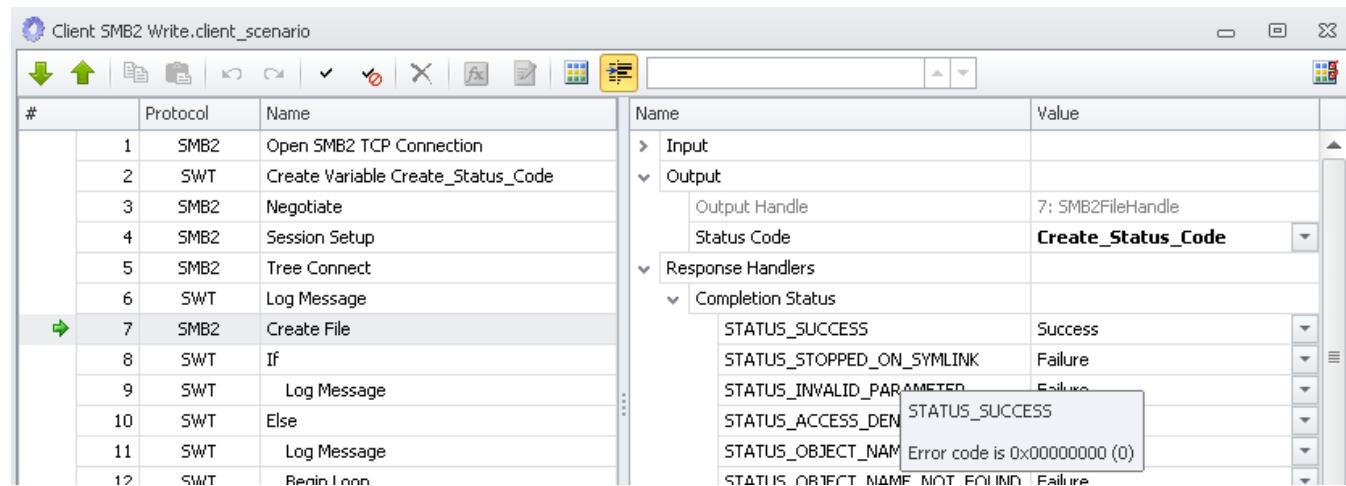


When the **Create File** Action is executed, the Status Code that the SMB2 server returns as a result of the execution of the Create File Action, is stored in the Variable associated with the Alias **Create_Status_Code**. The Scenario Control Action **If** that follows the **Create File** Action, the **If** in line 8, tests the contents of the Variable (**@variable(2)** using the line number reference format) and executes the Actions between the **If** and **Else** if the value of the Variable is anything but 0 (Failure for an SMB2 Action is some non-zero value).

If testing the value of the Variable



In the Scenario above, the Action between the **If** and **Else** is a Log Message Action which logs a text message to the Client Log file that says "File create not OK". If the value of the Variable is 0 (Success) then the Actions between the **Else** and the **End If** are executed which is a Log Message saying "File create OK" followed by a Loop writing content to the file that was created by the **Create File** Action. The potential Status Code values for a given Action can be seen by doing a mouse-over of the Action's Completion Status entries. See below for a subset of the Completion Status entries for the SMB2 **Create File** Action and a mouse-over of the STATUS_SUCCESS entry. For more details on Completion Status, see [Advanced Concepts: Response Handling](#).



Delay Execution Action

This Action's function is to delay the execution of the current instance of the a Scenario at the point of

inclusion of the Action in the Scenario

Action Inputs:

Delay Interval: Time interval (in milliseconds) to delay (idle) the execution of the current instance of current Scenario at the point of inclusion of the Action in the Scenario. Default is set to 1000 milliseconds.

Notes:

The execution of the current instance of the Scenario is delayed by the specified time interval unless:

- test expires while delay is in progress
- test is aborted while delay is in progress
- Scenario's execution was terminated due to an error in previous Action of the same Scenario

The expiration of current Load Profile does not affect delay processing;

Delay interval of 0 is allowed though effectively there will be no actual execution delay for the given Scenario instance. This is most usable in combination with User Parameters, since it allows the user to specify the desired proportion between active and idle Scenarios (i.e. if 1 row of User Parameters defines 0 delay, and 3 other rows define non-zero delays, then 75% of all running Scenarios will remain idle).

Begin Loop Action

The Action's function is to execute other Actions between this Action and matching "End Loop" Action in a loop until the specified condition is met.

Actions Inputs:

- Iteration Control: Type of data to be evaluated to determine whether to enter or continue this loop.

Two types of Iteration Control supported:

- Counter: execute the Begin-End loop this number of times (maximum value for a Loop Counter is $2^{63}-1 = 9,223,372,036,854,775,807$)
 - Load Profile: execute this Begin-End loop depending on the Load Profile condition
 - Time: execute this Begin-End loop for a specified period of time
 - IO Manager: execute this Begin-End loop for as long as the IO Manager has blocks to be Read or Written
- Condition Value: Value to be evaluated to determine whether to enter or continue this loop.

For "Iteration Control" = "Counter":

- the "Repeat Count" field indicates number of times to repeat the loop
- default value is 1
- used when a known number of iterations of a loop is known.

For "Iteration Control" = "Load Profile":

- the "Until start of ..." field indicates the duration based on the Load Profile state of the executing Project
- default value is "RampDown" (execute this loop until the Ramp Down Load Profile begins)
- non-default value is "any Load Profile" (execute this loop until the start of the next Load Profile)
- used when a loop is to be executed for as long as a Load Profile is executing

For "Iteration Control" = "Time":

- the "Repeat time(ms)" field specifies the length of Time in milliseconds that the Begin-End

loop is to execute

- the "Stop on RampDown" field is a True/False value that indicates if the Begin-End loop is to stop executing when the RampDown Load Specification begins

For "Iteration Control" = "IO Manager":

- the "IO Manager" field specifies the Handle to the IO Manager Action that controls this Begin-End loop

Notes:

Nested loops are allowed.

If there are no other Actions between "Begin Loop" and "End Loop", the loop is ignored and the warning message is logged.

If first "End Loop" is encountered before first "Begin Loop", this "End Loop" Action is ignored and the warning message is logged.

If "Condition Value" is 0 for "Repeat Count" condition, the Actions between "Begin Loop" and "End Loop" are not executed. Typically useful in combination with User Parameters.

Loop execution is aborted if:

- test expires
- test is aborted
- Scenario execution is aborted due to an error in the Action within loop
- Break Action is executed

The expiration of current Load Profile does not cancel the loop processing;

End Loop Action

The Action's function is to indicate the loop scope terminator for the matching "Begin Loop" Action, i.e. it indicates that the scope of the loop to be executed by "Begin Loop" Action is all the Actions that precede this "End Loop" Action up and including the closest "Begin Loop" Action. See "Begin Loop" Action for more details.

Action Inputs: None

Break

This Action's function is to force the execution of a Begin Loop - End Loop to exit to the next Action following the End Loop. Break can be used to exit from within a single Begin Loop - End Loop block or within nested Begin Loop - End Loop blocks. The Loop index remains at the current value when the Loop was exited.

Action Inputs:

Loop Handle: The output handle of the Begin Loop Action.

Continue

This Action's function is to force the execution of a Begin Loop - End Loop to the top (first Action) in a Begin Loop - End Loop block.. Continue can be used to start execution from the top of a single Begin Loop - End Loop block or from the top of nested Begin Loop - End Loop blocks. The Loop index is incremented.

Action Inputs:

Loop Handle: The output handle of the Begin Loop Action.

Logging Actions

Log Message

This Action's function is to write messages into the Client Port log file as the Project executes. Extensive use (many messages or lengthy messages) of Logging Messages can impact Project performance.

Action Inputs:

Logging Enabled: A means of controlling the writing of Log Messages

By Condition: Evaluate the input to the Condition field and if it evaluates to a non-zero value, then the Condition is True so write a message to the Client Port Log file (a maximum of up to Maximum Messages messages will be written). A zero value evaluates to False and no message is written.

Yes: Always write up to the Maximum Messages number of Log Messages to the Client Port log file.

No: Never write any Log Messages to the Client Port log file.

In Validation Run Only: Log Messages only if the Scenario is being executed in Validation Mode (a maximum of up to Maximum Messages messages will be written).

Maximum Messages: The maximum number of messages that will be written to the Client Port log file.

Severity: 3 classes of Severity Info, Status, Warning. Severity is indicated in the Status column of the Client Port log file.

Message: The content of the messages that are to be written to the Client Port log file.

Mathematical Actions

Formula: See [Appendix: Functions and Formula](#) for details

Distribution

This Actions function is to produce a distribution of values that can be used in Actions that require numeric input. This Action can be seen as similar to the @Random function that produces random integers given a set of inputs. The Distribution Action produces a Tester-defined distribution of integer values based on rules defined in the Distribution Action. The Weight values for the total set of Groups do not need to add up to any specific value. When the Distribution Action is executed, the Weight values will be added together and the Groups used the appropriate percentage of the time. For performance reasons, it is recommended that the DISTRIBUTION Action be outside of loops.

In the example below, the Distribution Action produces a distribution with 4 possible outputs. Group 1 which will occur 70% of the time = 4kb. Group 2 which will occur 10% of the time = 1kb. Group 3 which will occur 15% of the time = 16kb. Group 4 which will occur 5% of the time = 777.

The screenshot shows the LabVIEW TestStand Client interface with a scenario titled "Client SMB2 Write.client_scenario*". On the left, a list of actions is shown:

#	Protocol	Name
4	SWT	Reset User Parameters
5	SWT	Advance User Parameters
6	SWT	Reset User Parameters
7	SWT	Advance User Parameters
8	SWT	End If
9	#	
10	#	calculate write counters
11	SWT	Create Variable Total_Size
12	SWT	Create Variable Done_Size
13	SWT	Formula Remaining_Size
14	SWT	Distribution Block_Size_Distribution
15	#	-----
16	#	start of protocol flow
17	#	-----
18	SMB2	Open SMB2 TCP Connection
19	SMB2	Negotiate
20	SMB2	Session Setup

On the right, the configuration table for the Distribution action (row 14) is displayed:

Name	Value
Input	
Alias	Block_Size_Distribution
Count	4
Group 1	
Value 1	4kb
Weight 1	70
Group 2	
Value 2	1kb
Weight 2	10
Group 3	
Value 3	16kb
Weight 3	15
Group 4	
Value 4	777
Weight 4	5
Output	
Distribution	14: Distribution

In the formula below, the Distribution is used, referred to by its Alias "Block_Size_Distribution" to produce a value for the Variable named "block" which will be used in a File Write Action later in the Scenario. To get values generated by the Distribution Action, it is necessary to use the @Distribution function (DISTR below).

The screenshot shows the LabVIEW TestStand Client interface with a scenario titled "Client SMB2 Write.client_scenario*". On the left, a list of actions is shown:

#	Protocol	Name
17	#	-----
18	SMB2	Open SMB2 TCP Connection
19	SMB2	Negotiate
20	SMB2	Session Setup
21	SMB2	Tree Connect
22	SMB2	Create File
23	SWT	Begin Loop Main_Loop
24	#	block size
25	SWT	Create Variable block

On the right, the configuration table for the Create Variable block (row 25) is displayed:

Name	Value
Input	
Alias	block
Type	Unsigned Int
Expression	
CONDITION	dice < remaining
DICE	DISTR(Block_Size_Distribution)
REMAINING	FORMULA(Remaining_Size)
Output	
Variable	25: Variable

IO Control Actions

Set Auto Offset

This Action's function is to specify for various File-oriented and Block protocols (SMB, SMB2, SCSI, NFSv3, NFSv4, NFSv4.1), when Auto Offset is True for Read and Write, just Read or just Write, what the base offset value is.

Action Inputs:

Protocol: the Protocol that this Auto Offset setting applies to - SMB, SMB2, SCSI, NFSv3, NFSv4, NFSv4.1.

I/O Direction: The kind of I/O operations that this Auto Offset setting applies to - Read and Write, just Read or just Write.

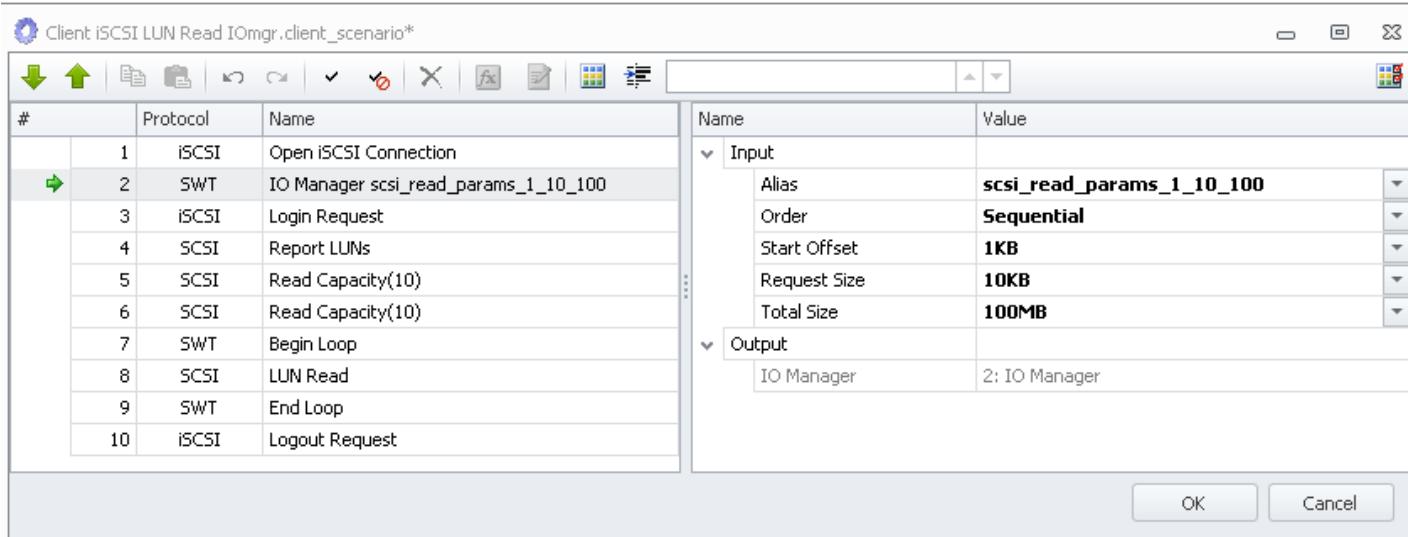
Read Offset: Initial Auto Offset for Read operations.

Write Offset: Initial Auto Offset for Write operations.

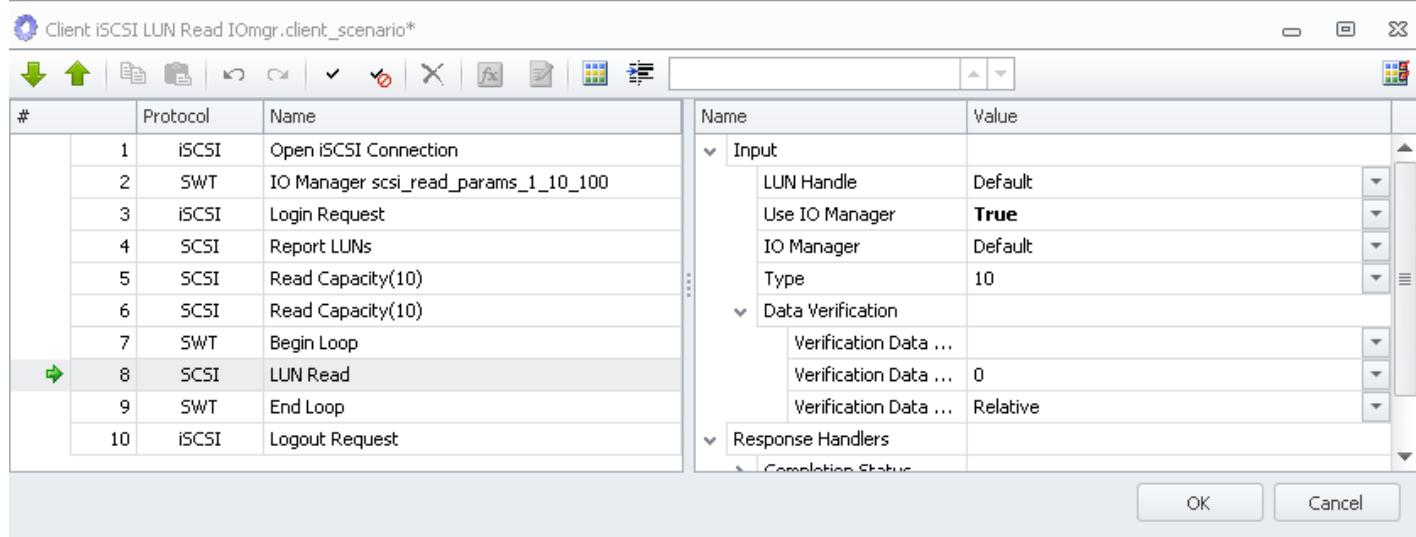
IO Manager Action

This Action's function is to specify to (currently) SCSI Read and Write Actions how the IO should behave (Order, Start Offset, Request Size, and Total Size). The inputs to the IO Manager Action are the same as what is input to SCSI Read and Write Actions: Initial Offset, Chunk Size, Total Transfer Size and Block Sequence. Order has two options: Sequential or Random and determines the order in which blocks are written. Start Offset is the lower bound of the Reads or Writes and Total Size is the total amount of data that is Read or Written. The data portion of each operation is Request Size bytes in length. A single instance of Read or Write Action (not in a Begin-End loop) referencing an IO Manager Action will be executed a single time with the parameters input to the IO Manager Action.

The IO Manager Action is currently has an experimental status and is not recommended for broader use.



The IO Manager use in this example would be.



User Parameter File Control Actions

Advance User Parameters Action

The Action's function is to advance the current User Parameters pointer to the next value or values. By default, when User Parameters are used in a Scenario, the runtime creation of the next instance of the Scenario advances the User Parameters next item pointer by 1. After the User Parameters data are advanced for given runtime Scenario instance, the User Parameters data remain constant in the context of this Scenario instance. If Scenarios share User Parameter files (either by using Global User Parameters or by sharing the same file instance as a Local User Parameter file), then each Advance

within a Scenario can impact the User Parameters in another Scenario.

Action Inputs:

- User Parameters to Advance: Default value of Local indicates to Advance the Local User Parameters file. Alternate values of UP0, UP1, ... indicate to Advance one of the Global User Parameters file as identified by the item selected (i.e. UP0 means Advance the User Parameter file with index = 0).
- Data Column to Advance: Default value of "All" indicates that current User Parameters are to be advanced to the next item. Alternatively, the single column header (specified as A, B, C,...) may be specified, in which case only the specified column is advanced to the next value.
- Scope: Default value of Global indicates that the impact of the operation (which UPF to Advance, which column or All to Advance and Step to indicate the number of rows to Advance) will be made to designated file on a Global basis which means that all Scenarios that use this file will be impacted.. Alternate value of Scenario indicates that the specified change is only felt local to the Scenario.
- Step: Default value of 1 indicates to Advance the next item pointer by one.

Reset User Parameters Action

The Action's function is conceptually the same as in "Advance User Parameters" Action, except that this Action advances current User Parameters to the first data value/values, instead of the next data value/values (i.e. Resets data value/values to the beginning of the list).

EXAMPLE

Consider the following table

A: User ID	B: Password
JohnJohnson	jj12345
FredFunstin	fred97
SamStrong	strong09876

Suppose that the contents of this table are in the Global User Parameter file at index 0 of the User parameters map.

In a two Scenario test, with no Advance or Reset Actions applied, the first Scenario reference to UP(0,A) returns "JohnJohnson". The second Scenario reference to UP(0,A) returns "FredFunstin". The third time a Scenario references UP(0,A), "SamStrong" is returned.

In the same two Scenario test, with a Reset User Parameters with

User Parameters to Reset = UP0
Data Column to Reset = ALL
Scope = GLOBAL

at the top of each Scenario will result in a reference to UP(0,A) returning "JohnJohnson" every time.

EVENTS

The Scenario synchronization/concurrency control feature.

Event Actions allow two Load DynamiX Scenarios to synchronize execution. One Scenario is programmed to **Wait For** an Event to arrive from another Scenario (pause executing) and the second Scenario is programmed to **Raise** (send) and Event when it is ready for the other Scenario to continue

executing. An example of where Events might be well utilized would be in a Scenario used to exercise file locking operations. One Scenario would Wait For a locked file to be created and the second Scenario would create that locked file and then Raise an Event to the first Scenario which could then begin its operations on that locked file.

Events are shared within a Logical Port so cooperating Scenarios can execute within the same Network Profile or they may execute on different Network Profiles but they **must** all be within the same Logical Port.

Cooperating Scenarios (Scenarios that exchange events) must also have Load Profiles in which the load generated by the Scenarios Raising Events is equal to the load generated by the Scenarios that are Waiting for Events. For example, if two Scenarios are cooperating, their Load Profiles must be equal. If three Scenarios are cooperating, the Load Profile of the Scenarios Raising the Events must equal the Load Profiles of the Scenarios Waiting for Events (e.g. one Scenario Raising Events Load Profile == 4 Concurrent Scenarios, two Scenarios Waiting for Events, Load Profile 1 == 3 Concurrent Scenarios and Load Profile 2 == 1 Concurrent Scenario).

Event Keys

Event Keys are strings. They must be identical between the two cooperating Scenarios and they must be unique for each instance of the pair of Scenarios. For this reason, User Parameter files provide the appropriate means of defining the Event Keys and there must be a pair of User Parameter files per cooperating pair of Scenarios. There are no limits on the number of or length of Event Keys. There must be more Event Key rows in the User Parameter File than there are concurrently executing Scenarios.

Raise Event Action

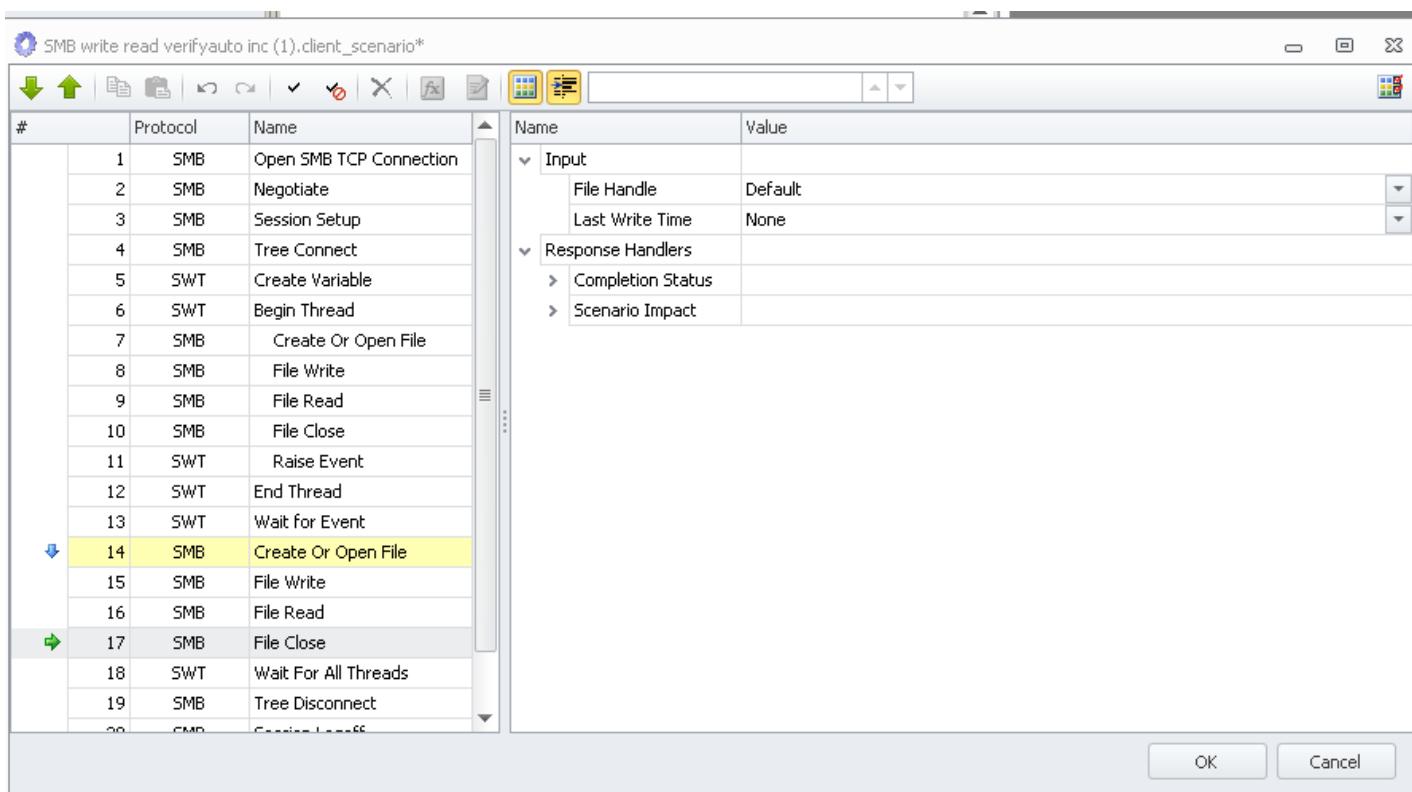
The purpose of the Raise Event Action is to signal the Event that is specified in the Key input field for the Action. The Key input field can be created by a Function or it may be a simple User Parameter file reference. The Action should be inserted in the Scenario at the point that the Event signal is required.

Wait For Event Action

The purpose of the Wait For Event Action is to wait for the Event specified in the Key input field to be signaled by some other Scenario. The Scenario that contains the Wait For Event Action will pause execution until that Event is signaled by some other Scenario. The Key input field can be created by a Function or it may be a simple User Parameter file reference but must be identical to the Event Key that is specified in the Scenario that is sending the signal.

Example

In the example Scenario below, a single Scenario uses Events to coordinate the execution between a Thread and the rest of the Scenario. The Thread (lines 6 - 12) is launched and when it is done an Event is Raised (line 11). Line 13, Wait for Event will pause the Scenario until the Thread signals it is done by Raising an Event. The Scenario then finishes executing starting with Line 14.



Event Troubleshooting

If a project that uses Events is not behaving as expected, make sure that the Project is following these 4 guidelines

Event Message Strings: Must be unique for each Wait For and Raise pair of Actions. The events can be made unique using Functions to create unique strings or they may be created by having unique strings in UP file entries. If Functions are used make sure that the Function calls are identical. If unique strings in UP files are used, make sure that all of the strings are unique.

Event Message String UP files: There must be a UP file for each of the cooperating Scenarios AND these files should only be used by one of the Scenarios. For example, if UP2 and UP3 contain the Event Message Strings then UP2 should be used by one Scenario and UP3 should be used by the other. It is imperative that these UP files remain in sync throughout the execution of the Scenarios.

Event Message String Count: The UP files that contain the Event Message Strings must have \geq rows than the number of concurrent cooperating Scenarios. For example, if there are 200 Cooperating Scenario pairs then the Event Message String UP files must have at least 200 rows of unique Event Message Strings.

THREADS and ASYNC

See [Advanced Concepts: Threads and Async Operations](#) for a detailed discussion of Thread and Async Actions.

COMMENTS

Testers may insert Comment actions in their Scenarios as a means to document what the Actions in the Scenario are intended to accomplish, what their inputs, expected outcomes, etc might be. When the Comment Action is inserted into a Scenario, the Comment shows as a "#" in the Protocol column and an empty name column entry. Comment text is created by typing into the empty window to the right of the Comment. The contents of the first line of the comment will be displayed in the Name field of the Comment. Comments have no impact on Scenario execution behavior.

VARIABLES

An updatable repository for Action inputs. See [Advanced Concepts: Variables and Aliases](#) for a detailed discussion of Create Variable and Update Variable Actions.

FORMULA

Mathematical calculations for Action inputs during Project execution. See [Appendix: Functions and Formula](#) for a detailed discussion of Formula.

Appendix: NFS v3, v4 and v4.1 Notes**Appendix: NFSv3, v4, v4.1 Notes:**

- Locking, Delegation and Reconnect
- pNFS, Open File/Open File Confirm, UID/GID and Owner ID
- Auto Lookup, Kerberos and Asynchronous I/O

Links to NFS protocol reference materials are provided in the [References and Terminology section](#).

Network Lock Manager

Network Lock Manager (NLM) is RPC-based protocol.

The NLM protocol provides advisory file locking semantics to NFS version 2 and 3.

This protocol is closely tied with the NFS protocol itself since it shares the file handle data structure with NFS.

NLM actions represented in Load DynamiX Client

Load DynamiX Client supports NLM version 4 protocol which is compatible with NFS version 3.

Open/Close NLM connection

#	Action	Comment
1	Open NLM TCP Connection	
2	Close NLM TCP Connection	

Synchronous procedures, monitored

#	Action	NLM Proc	Comment
3	Null	NLM_NULL: 0	Do Nothing This procedure does no work. By convention, procedure zero of any RPC program takes no parameters and returns no results. It is made available to allow server response testing and timing.
4	Test Lock	NLM_TEST: 1	Test Lock This procedure tests to see whether the monitored lock specified is available to this client.
5	Lock File	NLM_LOCK: 2	Establish a Lock This procedure attempts to establish a monitored lock.
6	Cancel lock	NLM_CANCEL: 3	Cancel Lock This procedure cancels an outstanding blocked lock request.
7	Unlock file	NLM_UNLOCK: 4	Unlock File This procedure will remove the lock specified.

Synchronous non-monitored lock and DOS file-sharing procedures

#	Action	NLM Proc	Comment
8	Share File	NLM_SHARE: 20	Share a File. This procedure indicates that a client wishes to open a file using the DOS file-sharing modes.
9	Unshare File	NLM_UNSHARE: 21	Unshare a File. The server will release the corresponding share reservation.
10	Non-monitored Lock File	NLM_NM_LOCK: 22	Non-monitored Lock This procedure has the same functionality as the NLM_LOCK procedure except that there is no monitoring performed via the NSM.
11	Free All Locks	NLM_FREE_ALL: 23	Free All. The server will discard all file-sharing reservations and file locks currently being held on behalf of the client.

Returns of NLM Procedures

#	Name:Num	Comment
1	NLM_GRANTED : 0	The call completed successfully.
2	NLM_DENIED : 1	The call failed. For attempts to set a lock, this status implies that if the client retries the call later, it may succeed.
3	NLM_DENIED_NOLOCKS : 2	The call failed because the server could not allocate the necessary resources.
4	NLM_BLOCKED : 3	Indicates that a blocking request cannot be granted immediately. The server will issue an NLM_GRANTED callback to the client when the lock is granted.
5	NLM_DENIED_GRACE_PERIOD : 4	The call failed because the server is reestablishing old locks after a reboot and is not yet ready to resume normal service.
6	NLM_DEADLCK : 5	The request could not be granted and blocking would cause a deadlock.
7	NLM_ROFS : 6	The call failed because the remote file system is read-only. For example, some server implementations might not support exclusive locks on read-only file systems.
8	NLM_STALE_FH : 7	The call failed because it uses an invalid file handle. This can happen if the file has been removed or if access to the file has been revoked on the server.
9	NLM_BIG : 8	The call failed because it specified a length or offset that exceeds the range supported by the server.
10	NLM_FAILED : 9	The call failed for some reason not already listed. The client should take this status as a strong hint not to retry the request.

TDE NLM action fields

#	Field	TDE Type	Comment
1	Connection Handle	Input of NLMConnectionHandle	NLM connection handle
2	Block	Boolean	Flag to indicate blocking behavior
3	Exclusive	Boolean	If exclusive access is desired
4	Caller Name	String	Uniquely identifies the host
5	File Handle	Input of NfsDirHandle	Identify a file
6	Owner	String	Identify owner of a lock
7	Offset	Integer	File offset (for record locking)
8	Length	Integer	Length (size of record)
9	Deny	List {None,Read,Write,Both}	Sharing mode. The mode determines the access that another client will be permitted when he attempts to share the file. A deny none value indicates that other clients can open the file for any kind of access.
10	Access	List {None,Read,Write,Both}	Indicates the kind of access to the file that the client requires. The SHARE request may be denied if the desired access is not compatible with the deny mode of an existing share.
11	Reclaim	Boolean	Used for recovering locks.

How to use NLM in TDE

TDE NLM action definitions were placed into *TDE/Toolbox/NFSv3/NLM*

1. Add program port for NLM:

- Add action *PMAPv2:Get Program Port*;
- Setup Program = "NLM";
- Setup Version = "4";
- Setup Protocol = TCP.

2. Add NLM connection:

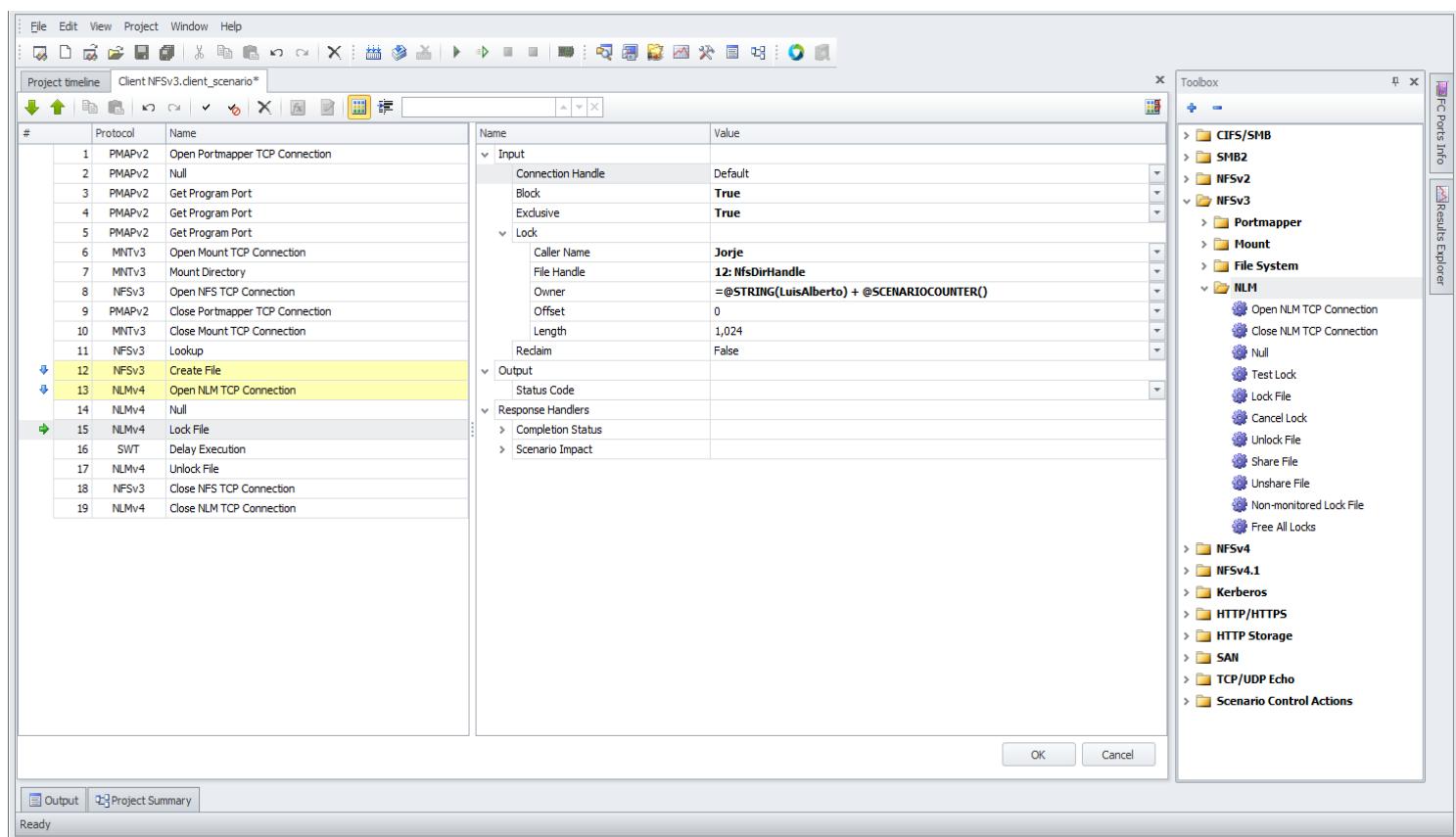
- Add action *Open NLM TCP Connection*;
- Setup Keep-Alive = "true".

3. Add NLM commands:

- Add some necessary NLM actions;
- Configure actions: *file handle, offset, length, ... (look at action fields above)*.

4. Close NLM connection:

- Add action Close NLM TCP Connection.



Caveats/Limits

1. NLM connection reconnect works if NLM Server does not change NLM service port after reboot.

NFSv4, NFSv4.1 Locking Commands

Lock File: This command is used to lock file blocks. The following fields are configurable:

Lock Type:

NFSv4	NFSv4.1
Read	Read
Write	Write
	Blocking Read (blocks other Read lock requests)
	Blocking Write (blocks other Write lock requests)

Reclaim

Offset

Length (for the whole file, set Length = NFS4_UINT64_MAX = 18446744073709551615 and set Offset = 0)

Lock Owner:

- Owner Type : There are two owner types:
 - New: This type is used to get lock for the first time.
 - Exist: This type is used to get the lock on a file already locked.
- Owner Name
- Client ID Handle

Unlock File : This command is used to Unlock file blocks. User specifies the offset and the length of the block.

Test for Lock : This command is used to check if a conflicting Lock exists or not. The fields required are similar to NFS Lock File command.

Note: If the File that is indicated in this command has a conflicting Lock then this command returns the Owner, Offset, Length and Type of Lock that is in place.

Release Lock Owner (NFSv4 only) : This command will release all the locks on a particular file specified by File Handle.

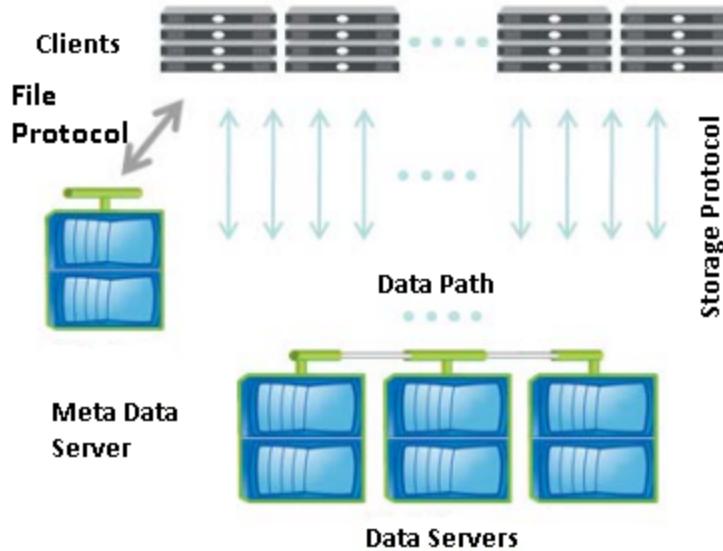
NFSv4.1 pNFS (parallel NFS)

NFSv4.1 provides a base on which the parallel NFS (pNFS) improvements have been introduced. pNFS allows users to realize improved storage performance and bandwidth in highly parallel or clustered computing environments. Load DynamiX hardware and software allow Testers to exercise pNFS file systems using the basic NFSv4.1 commands with a few new commands.

For NFSv4.1 pNFS, the Load DynamiX Client establishes at least two TCP connections:

- The File Protocol as primary connection for file meta data manipulation, session management and file layout info
- Storage Protocol connection to the Data Server for the purpose of file IO actions (Read, Write & Commit)
 - No manipulation of file/directory metadata (name, permissions, attributes) by the Storage Protocol, only data movement (read/write/commit).

A pNFS deployment might look like this:



There are four new Actions added to the Load DynamiX NFSv4.1 Toolbox to support pNFS:

Get Layout

- Only File type layouts are supported (no Block or Object)
- Read/Write, Read and Any IO Modes are supported
- The default layout request is set to full file layout access NFS4_UNIT64_MAX (Offset and Length are defaulted to 0 and 18446744073709551615 [0xFFFFFFFFFFFFFF]) respectively)

Get Device Info

- Provides the Device Address Handle for the Open pNFS TCP Connection
- Only a single Data Server address is supported

Open pNFS TCP Connection

- Establishes the Load DynamiX Client to Data Server connection.
- Only NFSv4.1 Read, Write and Commit commands are supported over this connection

Sequence

- A Keep-Alive command used to maintain an open connection between pNFS client and Meta Data Servers during long data transfer operations.

NOTE: During long pNFS data operations between the Client and a Data Server, the primary connection between the Client and the Meta Data Server will remain idle. This idle primary connection can timeout causing session errors for the data connection. The primary connection is therefore dependent on the servers TCP and NFS Session idle timeout configuration. A means of forcing the Meta Data Server to reset its Session Inactivity Timer for the primary connection is to spawn a thread that sends a periodic 'NFSv4.1 Sequence' request to the Meta Data Server for the duration of the scenario as shown below:

#	Protocol	Name		
7	NFSv4.1	Lookup		
8	NFSv4.1	Lookup		
9	NFSv4.1	Access		
10	NFSv4.1	Open File		
11	#	<i>begin pNFS Keep Alive</i>		
12	SWT	Begin Thread		
13	SWT	Begin Loop		
14	NFSv4.1	Sequence		
15	SWT	Delay Execution		
16	SWT	End Loop		
17	SWT	End Thread		
18	#	<i>end pNFS Keep Alive</i>		
19	NFSv4.1	Get Layout		
20	NFSv4.1	Get Device Info		
21	NFSv4.1	Open pNFS TCP Connection		

NFSv3 Auto Lookup Mode

NFSv3 servers may or may not require the execution of a Lookup command following the execution of a Create File, Create Directory or Create Symbolic Link command to get a File Handle that can be used downstream by commands such as Read File or Write File. To allow a single Project to operate correctly with NFSv3 servers that do and do not require Lookup, the Load Dynamix NFSv3 **Create File**, **Create Directory** and **Create Symbolic Link** Actions support Auto Lookup mode.

In Auto Lookup mode, the **Create File**, **Create Directory** and **Create Symbolic Link** Actions will determine if the NFSv3 server returns a viable File Handle in the response to the execution of a **Create File**, **Create Directory** and **Create Symbolic Link** Action. If the server does return a viable File Handle then the **Create File**, **Create Directory** and **Create Symbolic Link** Actions will use that Handle as its output. If the server does NOT return a viable File Handle then the **Create File**, **Create Directory** and **Create Symbolic Link** Actions will execute a Lookup command and use that Handle as its output. This allows Projects to be implemented such that they will operate correctly using both kinds of servers.

The default mode for Auto Lookup is Manual to maintain compatibility with Projects that have already been developed.

Auto Lookup Manual Mode

This screenshot shows the Client NFSv3 Write.client_scenario interface. On the left, a sequence of 16 actions is listed in a table:

#	Protocol	Name
1	PMAPv2	Open Portmapper TCP Connection
2	PMAPv2	Null
3	PMAPv2	Get Program Port
4	MNTv3	Open Mount TCP Connection
5	MNTv3	Null
6	MNTv3	Mount Directory
7	PMAPv2	Get Program Port
8	NFSv3	Open NFS TCP Connection
9	PMAPv2	Close Portmapper TCP Connection
10	MNTv3	Close Mount TCP Connection
11	NFSv3	Null
12	NFSv3	Access
13	NFSv3	Create File
14	NFSv3	Lookup
15	SWT	Begin Loop

On the right, the configuration pane displays the properties for action 13 (Create File). The "Input" section is expanded, showing the following settings:

Name	Value
Connection Handle	Default
Directory Handle	Default
File Name	=@UP(0,A)
Lookup	Manual
Create Mode	UNCHECKED

The "File Attributes" section is also expanded, showing the following settings:

Name	Value
Use Mode Attributes	True
Mode Attributes	0x000001B6 (438)
Use UID	False
UID	0
Use GID	False
GID	0
Use Size	False
Size	0

In Auto Lookup Manual mode, the **Create File**, **Create Directory** and **Create Symbolic Link** Actions do not produce an output Handle and require that Lookup be executed to provide that Handle. The **Write File** Action below will use the Handle produced by Lookup in line 14 as the File Handle input.

This screenshot shows the Client NFSv3 Write.client_scenario interface. The sequence table is identical to the one in the previous screenshot, ending at action 15 (Begin Loop).

The configuration pane for action 16 (Write File) is shown. The "Input" section is expanded, with the "File Handle" field highlighted and set to "Default". Other fields in this section include:

Name	Value
Connection Handle	Default
File Handle	Default
Automatic Offset	Default
Offset	6: NfsDirHandle 14: NfsDirHandle
Number of Bytes per R...	

The "Data Content" section is expanded, showing the following settings:

Name	Value
Number of Bytes	
How	
Block Sequence	Forward
Number of Outstanding...	1

The "Response Handlers" section is expanded, showing the following settings:

Name	Value
Completion Status	
Scenario Impact	

Auto Lookup Auto Mode

In Auto Lookup Auto mode, the **Create File** (and **Create Directory/Create Symbolic Link**) Action produces a Output Handle.

#	Protocol	Name
1	PMAPv2	Open Portmapper TCP Connection
2	PMAPv2	Null
3	PMAPv2	Get Program Port
4	MNTv3	Open Mount TCP Connection
5	MNTv3	Null
6	MNTv3	Mount Directory
7	PMAPv2	Get Program Port
8	NFSv3	Open NFS TCP Connection
9	PMAPv2	Close Portmapper TCP Connection
10	MNTv3	Close Mount TCP Connection
11	NFSv3	Null
12	NFSv3	Access
13	NFSv3	Create File

Name	Value
Connection Handle	Default
Directory Handle	Default
File Name	=@UP(0,A)
Lookup	Auto
Create Mode	UNCHECKED
File Attributes	
Use Mode Attributes	True
Mode Attributes	0x000001B6 (438)
Use UID	False
UID	0
Use GID	False
GID	0

The Handle output by Create File (line 13) is used by the **Write File** Action instead of the output of the Lookup Action which is no longer required to be in the Scenario and is executed by the **Create File** Action only if it is required.

#	Protocol	Name
1	PMAPv2	Open Portmapper TCP Connection
2	PMAPv2	Null
3	PMAPv2	Get Program Port
4	MNTv3	Open Mount TCP Connection
5	MNTv3	Null
6	MNTv3	Mount Directory
7	PMAPv2	Get Program Port
8	NFSv3	Open NFS TCP Connection
9	PMAPv2	Close Portmapper TCP Connection
10	MNTv3	Close Mount TCP Connection
11	NFSv3	Null
12	NFSv3	Access
13	NFSv3	Create File
14	SWT	Begin Loop
15	NFSv3	Write File

Name	Value
Connection Handle	Default
File Handle	Default
Automatic Offset	Default
Offset	6: NfsDirHandle 13: NfsDirHandle
Number of Bytes per R...	Number of Bytes How Block Sequence Number of Outstanding... 1
Data Content	Data Source
Response Handlers	Completion Status Scenario Impact

NFSv4/4.1 Open File and Open File Confirm

The NFSv4 and v4.1 **Open File** may or may not require an **Open File Confirm** Action to follow the Open File. Some NFSv4 and NFSv4.1 servers require it and some do not. To provide flexibility to the Tester, the Load DynamiX NFSv4/v4.1 Open File action can specify if the **Open File Confirm** Action is to be sent Automatically if required or sent Manually by including it in the Scenario Actions. The input that controls this behavior is Open Confirm (default == Manual). In Manual mode, the Scenario must provide the **Open File Confirm** Action following the **Open File** Action and it is always sent whether required or not. In Auto mode, if the NFSv4 or v4.1 server indicates that Open File Confirm is required then the Load DynamiX NFSv4/v4.1 Client will respond with the **Open File Confirm** automatically. This feature allows Clients to run against NFSv4/v4.1 servers some of which may require Open File Confirm and some of which may not.

Auto Mode specified

#	Protocol	Name
1	NFSv4	Open NFS TCP Connection
2	NFSv4	Null
3	NFSv4	Get Root File Handle
4	NFSv4	Access
5	NFSv4	Set Client ID
6	NFSv4	Set Client ID Confirm
7	NFSv4	Open File
8	NFSv4	Open File Confirm
9	NFSv4	Close File

Name	Value
Input	
Connection Handle	Default
Directory Handle	Default
Client ID Handle	Default
File Name	= @UP(0,A)
Open Confirm	Auto
Open Owner	Unique Per Open
Create On Open	True
Create Mode	UNCHECKED

Manual Mode specified

#	Protocol	Name
1	NFSv4	Open NFS TCP Connection
2	NFSv4	Null
3	NFSv4	Get Root File Handle
4	NFSv4	Access
5	NFSv4	Set Client ID
6	NFSv4	Set Client ID Confirm
7	NFSv4	Open File
8	NFSv4	Open File Confirm
9	NFSv4	Close File

Name	Value
Input	
Connection Handle	Default
Directory Handle	Default
Client ID Handle	Default
File Name	= @UP(0,A)
Open Confirm	Manual
Open Owner	Unique Per Open
Create On Open	True
Create Mode	UNCHECKED

NFSv4/4.1 Owner ID

The NFSv4 and v4.1 **Open File** specify an "owner ID" of the file being opened. Load DynamiX allows the Tester to generate a unique owner ID for every file open or every open connection to a server. The behavior is controlled by the **Open File** Action Open Owner input field (default == Unique Per Open). Allowed values for Open Owner are:

- **Unique Per Client** - create a unique owner ID once for this Scenario and use it in every Open File Action in this Scenario

#	Protocol	Name
1	NFSv4	Open NFS TCP Connection
2	NFSv4	Null
3	NFSv4	Get Root File Handle
4	NFSv4	Access
5	NFSv4	Set Client ID
6	NFSv4	Set Client ID Confirm
7	NFSv4	Open File
8	NFSv4	Open File Confirm
9	NFSv4	Close File

Name	Value
Input	
Connection Handle	Default
Directory Handle	Default
Client ID Handle	Default
File Name	= @UP(0,A)
Open Confirm	Manual
Open Owner	Unique Per Client
Create On Open	True
Create Mode	UNCHECKED

- **Unique Per Open** - create a unique owner ID with every execution of this Open File Action in this Scenario

#	Protocol	Name
1	NFSv4	Open NFS TCP Connection
2	NFSv4	Null
3	NFSv4	Get Root File Handle
4	NFSv4	Access
5	NFSv4	Set Client ID
6	NFSv4	Set Client ID Confirm
7	NFSv4	Open File
8	NFSv4	Open File Confirm
9	NFSv4	Close File

Name	Value
Connection Handle	Default
Directory Handle	Default
Client ID Handle	Default
File Name	= @UP(0,A)
Open Confirm	Manual
Open Owner	Unique Per Open
Create On Open	True
Create Mode	UNCHECKED

NFSv3/v4/v4.1 UID/GID

Testers may specify NFS Authentication mode in the **Open NFS TCP Connection** Action. Whatever mode is specified is reused throughout the NFS Scenario whenever Authentication information is required. Two choices for Authentication Mode ("Flavor") AUTH-UNIX and AUTH-NULL. AUTH-NULL means no Authentication information is provided (AUTH_NULL is a synonym for AUTH_NONE which is documented in SUNRPC). AUTH-UNIX means that the User ID # and Group ID # specified in the Open TCP Connection Action is used as credentials in all NFS Actions that require credentials (except the NULL Action which always uses AUTH-NULL). AUTH_UNIX is a synonym for AUTH_SYS which is documented in SUNRPC).

AUTH-NULL mode

#	Protocol	Name
1	NFSv4	Open NFS TCP Connection
2	NFSv4	Null
3	NFSv4	Get Root File Handle
4	NFSv4	Lookup
5	NFSv4	Lookup

Name	Value
Destination Address	172.16.1.142
Destination Port	2049
Flavor	AUTH_NULL

AUTH-UNIX mode

#	Protocol	Name
1	NFSv4	Open NFS TCP Connection
2	NFSv4	Null
3	NFSv4	Get Root File Handle
4	NFSv4	Lookup
5	NFSv4	Lookup
6	NFSv4	Access
7	NFSv4	Set Client ID

Name	Value
Destination Address	172.16.1.142
Destination Port	2049
Flavor	AUTH_UNIX
UID	0
GID	0

In AUTH-UNIX mode, the UID and GID fields (default == 0 for both) can be provided by any Function input (@RANDOM, @UP, @SCENARIO_COUNTER, etc)

NFS Kerberos Authentication

Kerberos may be used in NFS tests in much the same way it is used in SMB tests. The same four Kerberos commands are used to establish credentials for access to the NFS server and then the rest of the NFS proceeds as normal. The following NFS test Scenario illustrates the use of Kerberos Actions in an NFS context:

NFS (v3, v4, v4.1) **Create RPSEC GSS Context** Action is used when an NFS Project is required to use Kerberos Authentication. The **Create RPSEC GSS Context** Action takes a Connection Handle (NFS Server connection), a Kerberos TKT Handle (Kerberos TGS-REQ output) and RPSEC GSS version (1 or 2, default == 1), RPSEC GSS Service (none, privacy, integrity, default == none). Version specifies the RPSEC version (most servers support version 1) and Service indicates whether to encrypt the header information of the RPSEC packet or not (default == none/not). The **Destroy RPSEC GSS Context** Action destroys the RPSEC GSS Context.

#	Protocol	Name
2	Kerberos	AS-REQ
3	Kerberos	TGS-REQ
4	Kerberos	Close Kerberos Connection
5	NFSv4	Open NFS TCP Connection
6	NFSv4.1	Create RPCSEC GSS Context
7	NFSv4	Null
8	NFSv4	Get Root File Handle
9	NFSv4	Access
10	NFSv4	Set Client ID
11	NFSv4	Set Client ID Confirm
12	NFSv4	Open File
13	NFSv4	Open File Confirm
14	NFSv4	Close File
15	NFSv4.1	Destroy RPCSEC GSS Context

NFSv4/4.1 ACL Support

The NFSv4 and V4.1 Set Attributes Action can be used to set (write) the ACL for files or directories on NFSv4 and NFSv4.1 servers using NFSv4 and NFSv4.1 Scenarios.

NFSv4/v4.1 ACL

NFS Access Control List (ACL) is a list of Access Control Entries (ACE), each specifying an entity (such as a user) and some level of access for that entity. ACLs are used to specify access rights and permissions associated with a file or directory. NFSv4 protocol includes integrated support for ACLs which are similar to those used by Windows.

ACL Format

An NFSv4 ACL is written as an acl spec, which is a whitespace-delimited string consisting of one or more ace specs. An NFSv4 ACL would appear as:

<ACE_Spec_1> <ACE_Spec_2> ... <ACE_Spec_N>. An ace spec is a colon-delimited, 4-field string in the following format:

type:flags:principal:permissions

Where

Type is one of:

A	Allow	Allow Access
D	Deny	Deny Access
U	Audit	Log Access when access attempted based on methods specified
L	Alarm	Generate System Alarm when access attempted based on methods specified

Flags are zero or more of:

g	Group	Indicates that the Principal in this ACE is a Group
d	Directory-Inherit	Placed on a Directory, all new Directories should inherit this ACE
f	File-Inherit	Placed on a Directory, all new non-Directory files should inherit this ACE
n	No-Propagate-Inherit	Do not place ACEs on new Files or Directories that are Inheritable by subdirectories
i	Inherit-Only	Placed on a Directory, all new Files and Directories behave according to Directory-Inherit and File-Inherit settings
S	Successful-Access	If a file/directory that has an Audit or Alarm on it is Successfully Opened and has this Flag set, the Audit or Alarm will be executed
F	Failed-Access	If a file/directory that has an Audit or Alarm on it is UN-Successfully Opened and has this Flag set, the Audit or Alarm will be executed

Principal is one of:

A principal is either a named user (e.g., ‘myuser@nfsdomain.org’) or a group (provided the group flag is also set), or one of three special principals: ‘OWNER@’, ‘GROUP@’ and ‘EVERYONE@’, which are, respectively, analogous to the POSIX user, group and other

OWNER@	Current owner of the file or directory
GROUP@	Current group of the file or directory
EVERTONE@	Everyone

Permissions are zero or more of:

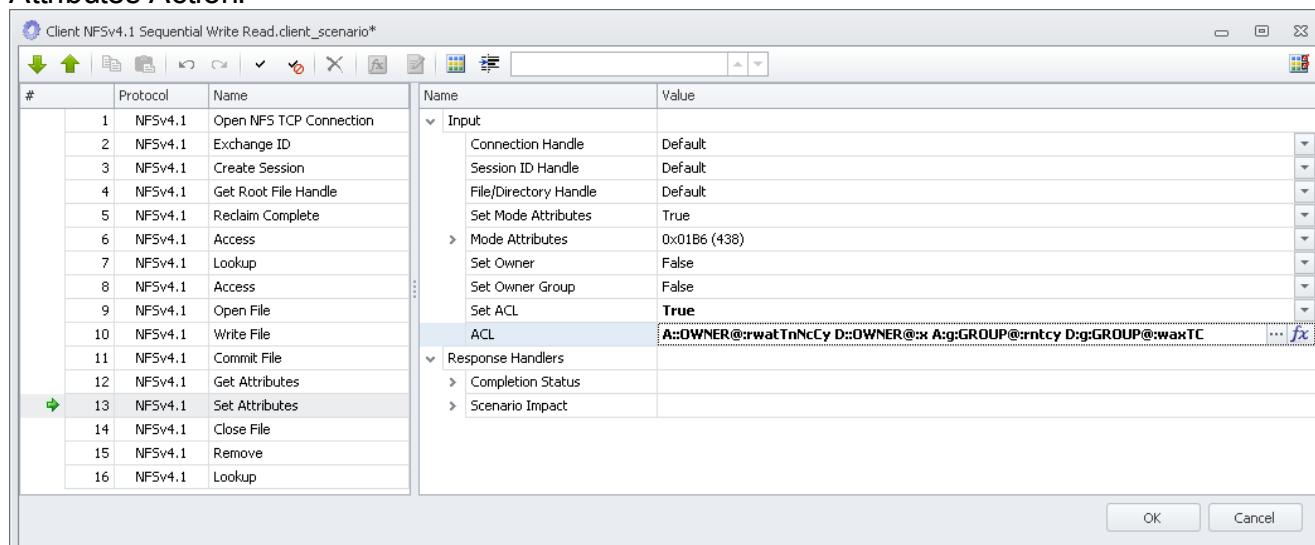
ACE representation	Permission	In 4.1?	In 4.0?
r	Read-Data (files) OR List-Directory (directories)	Y	Y
w	Write-Data (files) OR Create-File (directories)	Y	Y
a	Append-Data (files) OR Create-Subdirectory (directories)	Y	Y
x	Execute (files) OR Change-Directory (directories)	Y	Y
d	Delete (files) OR Delete (directories)	Y	Y
D	Delete-Child (directories only)	Y	Y
t	Read-Attributes (file) OR Read-Attributes (directory) (basic attributes only, not ACLs)	Y	Y
T	Write-Attributes (file) OR Write-Attributes (directory) (basic attributes only, not ACLs)	Y	Y
n	Read-Named-Attributes (file) OR Read-Named-Attributes (directory)	Y	Y
N	Write-Named-Attributes (file) OR Write-Named-Attributes (directory)	Y	Y
c	Read-ACL (file) OR Read-ACL (directory)	Y	Y
C	Write-ACL (file) OR Write-ACL (directory)	Y	Y
o	Write-Owner (file) OR Write-Owner (directory)	Y	Y
y	Synchronize (permission to access the file on the server with synchronous reads and writes)	Y	Y
e	Write Retention	Y	N
E	Write Retention Hold	Y	N

ACL Processing

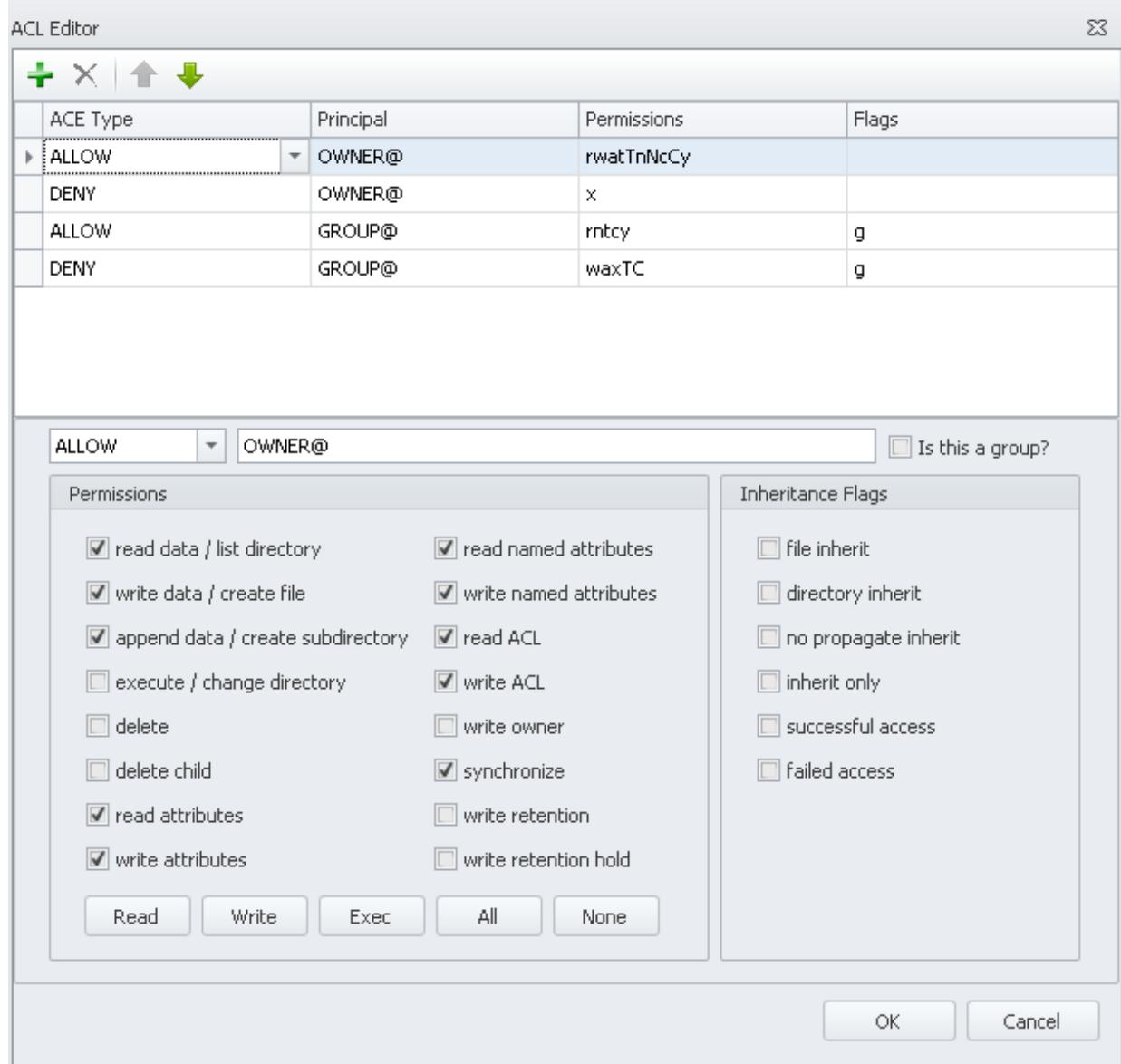
The NFSv4 server will process the ACE_Specs in the order received, thus the settings specified in ACE_Spec_N-1 could change the servers ability to process ACE_Spec_N.

TDE/Appliance Support

The TDE provides an ACL Editor which can be accessed through the NFSv4 and NFSv4.1 Set Attributes Action.



When Set ACL is set to True in the Set Attributes Action and by clicking the ellipsis (...) on the right hand side of the input field, the TDE will launch the ACL Editor (v4.1 ACL Editor is shown below)



The contents of the ACL input field of the **Set Attributes** Action can be typed in by hand or created by using the ACL Editor. The ACL Editor creates and orders ACE entries (<ACE_Spec_N>).

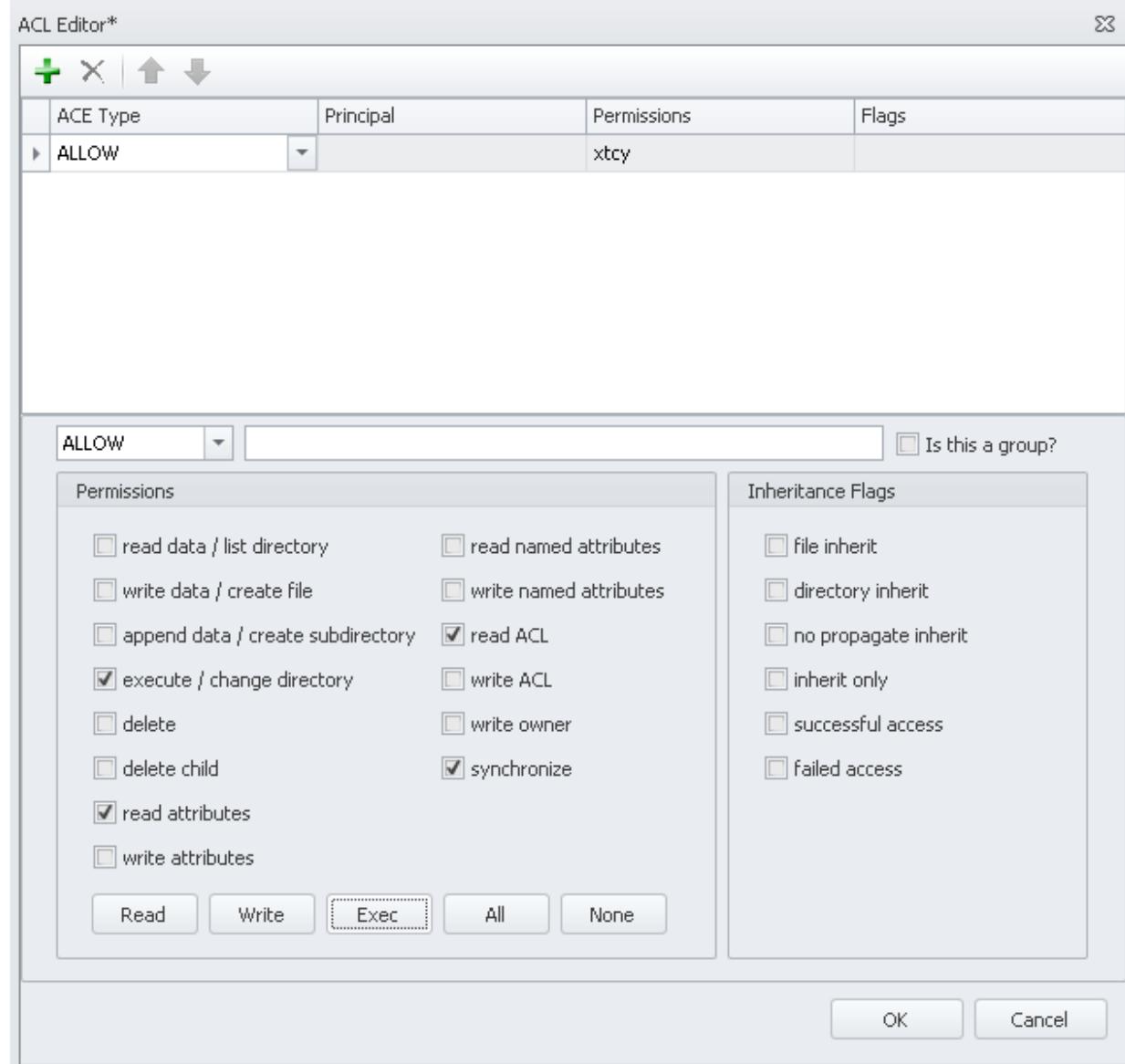
In the example above, the ACL Editor is used to create the contents ACL input field of the Set Attributes Action which will be applied to the file that was opened in line 9. ACE entries are added to the ACL by using the button. Moved Up and Down using the Up and Down Arrows and Deleted using the button.

In the first ACE entry, OWNER@ is being Allowed Read, Write, Append, Read Attributes, Write Attributes, Read Named Attributes, Write Named Attributes, Read ACL, Write ACL and Synchronize.

In the second ACE entry, OWNER@ is Denied Execute permission. In the third ACE entry, the current Group (GROUP@) is being given Read-Data, Read-Named-Attributes, Read-Attributes (basic), Read-ACL and Synchronize permissions. In the final ACE entry, the current Group is being Denied Write-Data, Append-Data, Write-Attributes and Write-ACL.

The NFSv4.1 ACL Editor supports two Permissions (e: Write Retention and E: Write Retention Hold) that the NFSv4.0 ACL Editor does not.

If the Project is a NFSv4, the NFSv4.0 ACL Editor will be launched when the (...) is clicked



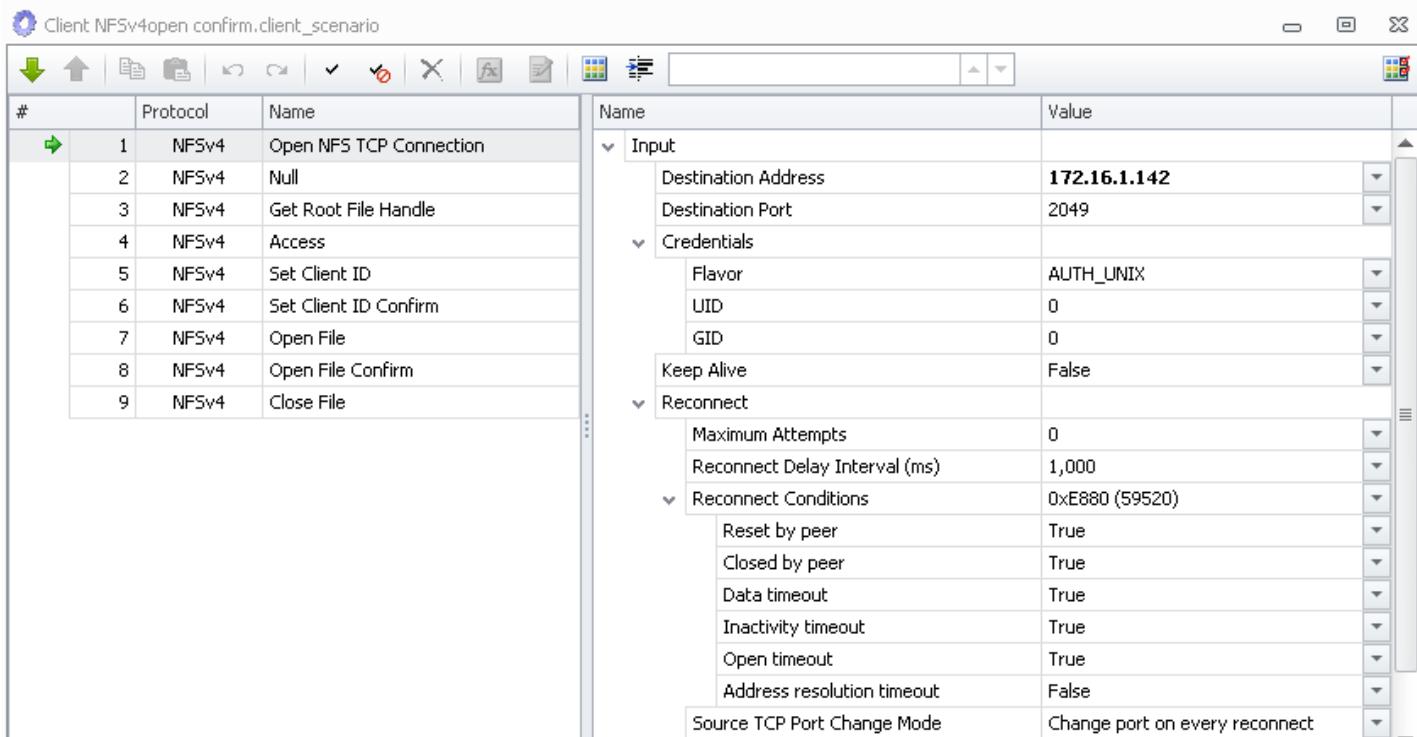
with similar operating behaviors as described above in the v4.1 ACL Editor.

NFSv4 Delegation

Delegation facilitates distributed file sharing in NFSv4. NFSv4 Delegation has been added to the supported NFSv4 behaviors via the NFSv4 Action: **SetClientID**. **SetClientID** now has a property called Delegation Callback that must be Enabled for NFSv4 operations to support Delegation.

NFSv4 Reconnect

NFSv3, v4 and v4.1 Projects support the ability to reconnect to an open connection that has failed for some reason during execution. The screenshot below shows an NFSv4 Open NFS TCP Connection with the reconnect parameters set to try to reconnect up to 4 times, with a 1 second (1000 milliseconds) delay between attempts, the set of the disconnect causes allowed (Reset by Peer, Close by Peer, Data Timeout, Inactivity Timeout and Open Timeout) and guidance to change TCP port on every reconnect attempt. If the reconnection succeeds, the Scenario will remember where it was executing and continue from there. If the reconnect does not succeed in 4 attempts or failed due to Address resolution timeout or Asynchronous Logout Request, then the Scenario will Abort.



NFSv3 & NFSv4.1 Asynchronous I/O (Read and Write Number of Outstanding Requests)

NFSv3 and NFSv4.1 Read and Write operations can be segmented into multiple operations that are issued asynchronously. To do this use the Number of Outstanding Requests feature of the NFSv3 and v4.1 Read and Write Actions. Below are two screen shots showing the Read and Write Actions with the Number of Outstanding Requests field. The allowed range of values for the Number of Outstanding Requests field is 1 to 128. Values < 1 or > 128 will result in an error that will prevent saving the Project.

The default value of Number of Outstanding Requests is 1 which results in same behavior as before Number of Outstanding Requests was introduced. The generic behavior of a Read or Write with Number of Outstanding Requests > 1 is that the operation (Read or Write) is broken into up to N

independent operations of size Number of Bytes per Request where

$N = (\text{Number of Bytes}/\text{Number of Bytes per Request}) \text{ or Number of Outstanding Requests}$,

whichever is smaller and these N operations are all issued at the same time. As these requests complete, new operations are issued until the total number represented by ($\text{Number of Bytes}/\text{Number of Bytes per Request}$) has been completed. Any errors encountered during the processing of these requests will cause the entire Action to fail.

In the **Read File** Action below, Number of Bytes per Request == 1kb, Number of Bytes == 10kb so the result of the division is 10. Number of Outstanding Requests is 5 so $N == 5$ and initially 5 requests will be sent in parallel to the target NFSv4.1 server and as any one of these 5 requests completes, the remaining requests will be issued.

#	Protocol	Name
22	NFSv4.1	Exchange ID
23	NFSv4.1	Create Session
24	NFSv4.1	Reclaim Complete
25	NFSv4.1	Write File
26	NFSv4.1	Commit File
27	NFSv4.1	Read File
28	NFSv4.1	Close File
29	NFSv4.1	Remove
30	NFSv4.1	Destroy Session
31	NFSv4.1	Close NFS TCP Connection
32	NFSv4.1	Destroy Session

Properties for Action 27 (Read File):

Name	Value
Input	
Connection Handle	Default
Session ID Handle	Default
File Handle	Default
State ID Handle	10: NfsStateIDHandle
Automatic Offset	False
Offset	0
Number of Bytes per Request	1kb
Number of Bytes	10kb
Block Sequence	Forward
Number of Outstanding Requests	5

Similarly, in the **Write File** Action below, the result of the division of Number of Bytes per Request / Number of Bytes == 10 and the Number of Outstanding Requests == 8, so 8 Write requests will be issued initially and then as the 8 complete, the remaining two requests will be issued.

#	Protocol	Name
22	NFSv4.1	Exchange ID
23	NFSv4.1	Create Session
24	NFSv4.1	Reclaim Complete
25	NFSv4.1	Write File
26	NFSv4.1	Commit File
27	NFSv4.1	Read File
28	NFSv4.1	Close File
29	NFSv4.1	Remove
30	NFSv4.1	Destroy Session
31	NFSv4.1	Close NFS TCP Connection
32	NFSv4.1	Destroy Session
33	NFSv4.1	Close NFS TCP Connection

Properties for Action 25 (Write File):

Name	Value
Input	
Connection Handle	Default
Session ID Handle	Default
File Handle	Default
State ID Handle	10: NfsStateIDHandle
Automatic Offset	False
Offset	0
Number of Bytes per Request	1kb
Number of Bytes	10kb
Block Sequence	Forward
Number of Outstanding Requests	8

Properties for Action 27 (Read File):

Name	Value
Data Verification	

Performance Impact

For tests that utilize a single connection between the Load DynamiX Appliance and the DUT (i.e. contain a single Scenario with a Load Specification of 1 Concurrent Scenario or 1 Concurrent Connection), the asynchronous I/O approach described above can provide a significant performance boost. Projects that utilize multiple connections can generally outperform a single connection that does not use asynchronous I/O.

See [Reference: NFSv3 Command List](#) and [Reference: Kerberos v5 Command List](#) for a complete list of supported NFS and Kerberos commands.

Handle Variable Actions

Two of the Load DynamiX Scenario Control Actions that apply ONLY to the NFS protocols are:

- **Create Handle Variable:** Create a Variable that contains NFS Handles of various kinds
- **Update Handle Variable:** Update a specific instance of a Handle Variable with a new or different NFS Handle

Variables (as documented in [Advanced Concepts: Variables and Aliases](#)) are storage for values created and potentially updated during Project execution that are used as inputs to Action parameters. Variables can contain the output of Actions that produce string or integer values but NOT the various Handles that Actions may also produce. Handle Variables are just the opposite, they can only contain Handles (NFS Handles). These Variables can be created and used/re-used throughout NFS scenarios as inputs to NFS Actions.

The Scenario shown below, shows how a Handle Variable could be used within a Begin Loop-End Loop sequence to write multiple NFSv4.1 files.

#	Protocol	Name
1	NFSv4.1	Open NFS TCP Connection
2	NFSv4.1	Exchange ID
3	NFSv4.1	Create Session
4	NFSv4.1	Get Root File Handle
5	NFSv4.1	Reclaim Complete
6	NFSv4.1	Access
7	NFSv4.1	Lookup
8	NFSv4.1	Lookup
9	NFSv4.1	Access
10	NFSv4.1	Open File
11	SWT	Create Handle Variable [10]
12	SWT	Begin Loop
13	NFSv4.1	Write File
14	NFSv4.1	Close File
15	NFSv4.1	Remove
16	SWT	Advance User Parameters
17	NFSv4.1	Open File
18	SWT	Update Handle Variable [17]
19	SWT	End Loop
20	NFSv4.1	Close NFS TCP Connection

The Handle Variable is created in line # 11 and contains the NfsFileHandle created by the Open File Action in line # 10. The Open File Action gets the File Name to Create from a User Parameter File column A and opens it.

Within the Begin Loop-End Loop sequence, the Handle Variable is used by the Write File Action in line # 13 to write 640,000 bytes to an open file. At the bottom of the Begin Loop-End Loop sequence, the next File Name to open is acquired by Advancing the User Parameter File (line # 16), Opening the new File Name (line #17) and, in line # 18, storing the NfsFileHandle output of the Open File Action in the Handle Variable created in line # 11. This process will be repeated for the duration of the Begin Loop-End Loop sequence.

Create/Update Handle Variable Notes:

1. Only valid in NFS Protocol (NFSv2, NFSv3, NFSv4, NFSv4.1) Scenarios.
2. The TDE will compile Variable Handle Actions successfully in other Protocol Scenarios (ex: SMB2, iSCSI, HTTP, etc.) but these Scenarios will fail at run time.

Appendix: Sample Projects

Appendix: Load DynamiX Sample Projects

Load DynamiX TDE sample Projects are installed in the Project Library's Sample Projects folder. The sample Projects are arranged in a folder hierarchy by protocol type HTTP, Block (iSCSI and Fibre Channel), File (SMB and NFS), HTTP Storage (CDMI and OpenStack Swift) and Miscellaneous (Load DynamiX 5000 and IPv6). The SMB folder contains CIFS-SMB, SMB2 and SMB3 sub-folders). The NFS folder has NFSv2, NFSv3, NFSv4 and NFSv4.1 sub-folders. The Load DynamiX Sample Projects are shipped read-only and must be saved elsewhere before they can be used. Links to CIFS-SMB, SMB2, NFSv3, NFSv4, NFSv4.1, HTTP, Amazon S3, CDMI, OpenStack protocol reference materials are provided in the [References and Terminology section](#).

Samples	Project Information
FOLDER: CIFS-SMB	
CIFS-SMB 4x4 Connection Rate	8 Ports, SMB TCP connect operations benchmark, internal Load DynamiX server
CIFS-SMB 4x4 Sequential Write Read	8 Ports, SMB write then read operations, internal Load DynamiX server
CIFS-SMB FD Payload - W2K3 Server	1 Port, SMB interlaced write and read operations, external Windows 2003 server
CIFS-SMB FD Payload - W2K3 Server Kerberos	1 Port, SMB interlaced write and read operations, external Windows 2003 server
CIFS-SMB Full Duplex Echo	2 Ports, SMB echo operations, internal Load DynamiX server
CIFS-SMB Full Duplex Payload	2 Ports, SMB interlaced write and read operations, internal Load DynamiX server
CIFS-SMB Full Duplex Payload SMB Signing	2 Ports, SMB interlaced write and read operations, SMB signing enabled, internal Load DynamiX server
CIFS-SMB Read	2 Ports, SMB read operations, internal Load DynamiX server
CIFS-SMB Read Compound Actions	2 Ports, SMB read operations using Compound Actions, internal Load DynamiX server
CIFS-SMB Read - W2K3 Server	1 Port, SMB read operations, external Windows 2003 server
CIFS-SMB Sequential Write Read	2 Ports, SMB write then read operations, internal Load DynamiX server
CIFS-SMB Threads and Async	2 Ports, SMB write and read operations using Thread and Async blocks, internal Load DynamiX server
CIFS-SMB TCP Connection Rate	2 Ports, SMB TCP connect operations benchmark, internal Load DynamiX server
CIFS-SMB Tree Connection Rate	2 Ports, SMB Tree Connect operations benchmark, internal Load DynamiX server
CIFS-SMB Write	2 Ports, SMB write operations, internal Load DynamiX server
CIFS-SMB Write - W2K3 Server	1 Port, SMB write operations, external Windows 2003 server
FOLDER: SMB2	
SMB2 4x4 Connection Rate	8 Ports, SMB2 TCP connect operations benchmark, internal Load DynamiX server
SMB2 4x4 Sequential Write Read	8 Ports, SMB2 Write then read operations, internal Load DynamiX server
SMB2 FD Payload - W2K8 Server	1 Port, SMB2 interlaced write and read operations, external Windows 2008 server
SMB2 FD Payload - W2K8 Server Kerberos	1 Port, SMB2 interlaced write and read operations, external Windows 2008 server, Kerberos authentication
SMB2 File Open Close Rate	2 Ports, SMB2 file open and close operations benchmark, internal Load DynamiX server
SMB2 Full Duplex Payload	2 Ports, SMB2 interlaced write and read operations, internal Load DynamiX server
SMB2 Full Duplex Payload Compound Actions	2 Ports, SMB2 interlaced write and read operations using Compound Actions, internal Load DynamiX server
SMB2 Full Duplex Payload SMB Signing	2 Ports, SMB2 interlaced write and read operations, internal Load DynamiX server SMB Signing enabled
SMB2 Read	2 Ports, SMB2 read operations, internal Load DynamiX server
SMB2 Sequential Write Read	2 Ports, SMB2 server write then read operations, internal Load DynamiX server
SMB2 TCP Connection Rate	2 Ports, SMB2 TCP connect operations benchmark, internal Load DynamiX server
SMB2 Tree Connection Rate	2 Ports, SMB2 Tree Connect operations benchmark, internal Load DynamiX server
SMB2 Write	2 Ports, SMB2 Write operations, internal Load DynamiX server
MS-RPC NetShare Operations	1 Port, SMB2/MSRPC NetShare operations used to Create, Enumerate and Delete a Share on a target server
MS-RPC NetFile Operations	1 Port, SMB2/MSRPC NetFile operations used to Create a Share and Create a File on that Share target
SMB2.1 Read Write Multi-Credit	1 Port, SMB2.1 Read and Write Actions using multiple credits to increase maximum write and read size
FOLDER: SMB3	
SMB3 Multi-Channel w/ Query Interface Handle	1 Port, SMB3 MultiChannel connection and operations, external SMB3 server
SMB3 Directory Leasing	1 Port, SMB3 Directory Leasing using v2 protocol, external SMB3 server
SMB3 Durable V2 Reconnect	1 Port, SMB3 Persistent Handles using v2 protocol, external SMB3 server

FOLDER: NFSv3	
NFSv3 Connection Rate	1 Port, NFSv3 Open TCP Connection benchmark, external NFS server
NFSv3 Full Duplex Payload	2 Port, NFSv3 interlaced write and read operations, internal Load DynamiX server
NFSv3 Null	1 Port, NFSv2 Null operation benchmark, external NFS server
NFSv3 Sequential Write Read	1 Port, NFSv3 Write then Read operations, external NFS server
NFSv3 Sequential and Random Data Content	1 Port, NFSv3 Write then Read with Data Verification, external NFS server
NFSv3 Write	1 Port, NFSv3 write operations, external NFS server
FOLDER: NFSv4	
NFSv4 Commands	1 Port, NFSv4 operations, external NFSv4 server
NFSv4 Commands with Delegation	1 Port, NFSv4 operations, Delegation enabled, external NFSv4 server
NFSv4 Full Duplex Payload	1 Port, NFSv4 interlaced write and read operations, external NFSv4 server
NFSv4 Sequential Write Read	1 Port, NFSv4 write then read operations, external NFSv4 server
FOLDER: NFSv4.1	
NFSv4.1 Full Duplex Payload	1 Port, NFSv4.1 operations, interlaced write and read operations, external NFSv4.1 server
NFSv4.1 Sequential Write Read	1 Port, NFSv4.1 write then read operations, external NFSv4.1 server
NFSv4.1 Sequential Write Read using pNFS	1 Port, NFSv4.1 in pNFS environment write then read operations, external NFSv4.1 server
NFSv4.1 Write Read using Threads Async NOR	1 Port, NFSv4.1 write read operations using Threads, Async blocks and NOR, external NFSv4.1 server
FOLDER: HTTP	
HTTP Full Duplex Payload	2 Port, HTTP file transfer performance to an internal HTTP server
HTTP Pre-Emptive Authentication Maximum Rate	2 Port, HTTP Get using preemptive Authentication to an internal HTTP server
HTTP PUTGET with Data Verification	2 Port, HTTP Put and Get with Data Verification to an internal HTTP server
HTTPS Full Duplex Payload	2 Port, HTTPS file transfer performance to an internal HTTPS server
HTTP Response Header and Body Parsing	2 Port, HTTP GET used to demonstrate use of Header and Body values received by one Action rule
HTTP Encoding	2 Port, HTTP Put and Get using Transfer Encoding Chunked and Content Encoding gzip, to internal and external servers
HTTP Pipelining	2 Port, HTTP actions executed in Async Blocks
HTTP Threads	2 Port, HTTP actions executed in Threads
HTTP Completion Status	2 Port, HTTP actions generate specific Completion Status codes: 409, 405, 404
FOLDER: Fibre Channel	
FC All Commands	1 Port, Fibre Channel connection, SCSI operations, external Fibre Channel device
FC Full Duplex Payload	1 Port, Fibre Channel connection, SCSI interlaced read and write operations, external Fibre Channel device
FC IOPS and Throughput	1 Port, Fibre Channel connection, SCSI operations, maximize IOPS or Throughput, external Fibre Channel device
FC Random Read Write	1 Port, Fibre Channel connection, SCSI operations, random reads 4K block size, external Fibre Channel device
FC Sequential Read Write	1 Port, Fibre Channel connection, SCSI operations, sequential reads 4K block size, external Fibre Channel device
FC UnMap Command	1 Port, Fibre Channel connection, SCSI UnMap command, external Fibre Channel device
FC Extended Copy	1 Port, Fibre Channel connection, SCSI Extended Copy command moves blocks from disc to disc, external Fibre Channel device
FC Compare and Write - Compare Fail	1 Port, Fibre Channel connection, SCSI Compare and Write command fails, external Fibre Channel device
FC Compare and Write - Compare Succeed	1 Port, Fibre Channel connection, SCSI Compare and Write command succeeds, external Fibre Channel device
FC Backup and Recovery	1 Port, Fibre Channel connection, SCSI SBC (block) and SSC (streaming) commands simulating Backup and Recovery
FC SSC All Commands	1 Port, Fibre Channel connection, SCSI SSC Commands, external Fibre Channel device
FC Write Same	1 Port, Fibre Channel connection, SCSI Write Same command used to zero out portions of a disc, external Fibre Channel device
FC MPIO and ALUA Commands - Single Scenario	2 Port, Fibre Channel connections, demonstrates MPIO configuration and ALUA target management
FOLDER: iSCSI	
iSCSI Actions Per Second	2 Ports, iSCSI NOP-Out operations, internal Load DynamiX target
iSCSI All Commands	1 Port, iSCSI connection, All SCSI Commands, external iSCSI server
iSCSI Discovery	1 Port, iSCSI LUN discovery using Login command and Text request, external iSCSI target
iSCSI Full Duplex Payload	2 Ports, iSCSI interlaced LUN write and read operations, internal Load DynamiX target
iSCSI FD Payload Async Bandwidth Generation	2 Ports, iSCSI interlaced LUN write and read operations in Async blocks, internal Load DynamiX target
iSCSI FD Payload Threads Bandwidth Generation	2 Ports, iSCSI interlaced LUN write and read operations in Thread blocks, internal Load DynamiX target
iSCSI VAAI Workload	1 Port, iSCSI connection, use of VAAI related iSCSI commands, external iSCSI target
iSCSI Full Duplex Payload DCB and HTTP	3 Ports (2 Client, 1 Server), iSCSI and HTTP Clients, DCB enabled, internal Load DynamiX iSCSI and external Load DynamiX HTTP server
iSCSI Unmap Command	1 Port iSCSI connection SCSI UnMap command external iSCSI device

iSCSI MPIO and ALUA Commands - Single Scenario	1 Port, multiple iSCSI connections, demonstrates MPIO configuration and ALUA target manager
FOLDER: Object Storage	
FOLDER: AMAZON S3	
Amazon S3 All Commands	1 Port, HTTP connection, All Amazon S3 Commands, Amazon S3 enabled external device
Amazon S3 Multipart Upload	1 Port, HTTP connection, Multipart Upload feature, Amazon S3 enabled external device
FOLDER: CDMI	
CDMI Client	1 Port, HTTP connection, Cloud Data Management Interface Container and Data Object Commands
FOLDER: OpenStack	
OpenStack Swift All Commands	1 Port, HTTP connection, All OpenStack Swift Commands, OpenStack Swift enabled external dev
OpenStack Swift All Commands (Keystone)	1 Port, HTTP connection, All OpenStack Swift Commands, Keystone authentication, OpenStack S
OpenStack Swift Client	1 Port, HTTP connection, OpenStack Swift Container and Data Object Commands, OpenStack Sw
OpenStack Cinder All Commands	1 Port, HTTP connection, All OpenStack Cinder Commands, OpenStack Cinder enabled external d
FOLDER: Miscellaneous	
FOLDER: High Performance	
10Gbe Samples	
10Gbe CIFS-SMB Full Duplex Echo	2 Ports, SMB echo operations, internal Load DynamiX server, configured for the Load DynamiX 1
10Gbe CIFS-SMB Full Duplex Payload	2 Ports, SMB interlaced write and read operations, internal Load DynamiX server, configured for
10Gbe CIFS-SMB TCP Connection Rate	2 Ports, SMB TCP connect operations benchmark, internal Load DynamiX server, configured for t
10Gbe SMB2 Full Duplex Payload	2 Ports, SMB2 interlaced write and read operations, internal Load DynamiX server, configured fo
10Gbe NFSv3 Full Duplex Payload	2 Ports, NFSv3 interlaced write and read operations, internal Load DynamiX server, configured fo
FOLDER: Specific Functionality Samples	
Advanced Load Profiles using SMB2	8 Ports, SMB2 TCP connect operations benchmark, internal Load DynamiX server, uses the Advan
Arithmetic Functions using SMB2	2 Port, SMB2 project using Formulas to calculate input to Read and Write Actions
User Parameter Aliases using SMB2	2 Ports, SMB2 project using User Parameter File Aliases from Local and Global User Parameter fil

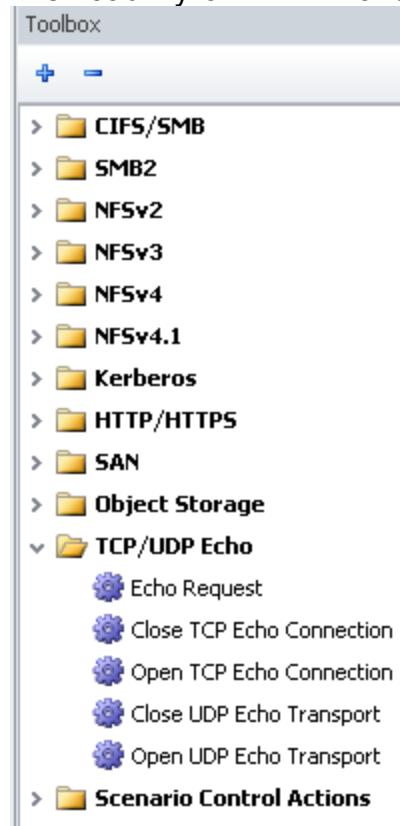
Appendix: TCP/UDP Echo and Discard Protocols

Appendix: TCP/UDP Echo and Discard Protocols

TCP ECHO Protocol

The TCP ECHO Protocol is defined as one of the services in the Internet Protocol Suite ([RFC 862](#)). The TCP ECHO Protocol is intended to be used as a debugging and measurement tool. It is a service that listens on TCP port 7. Data received by the TCP ECHO server is echoed back to the sender.

The Load DynamiX TDE and Appliance support five TCP/UDP ECHO Protocol Client Actions

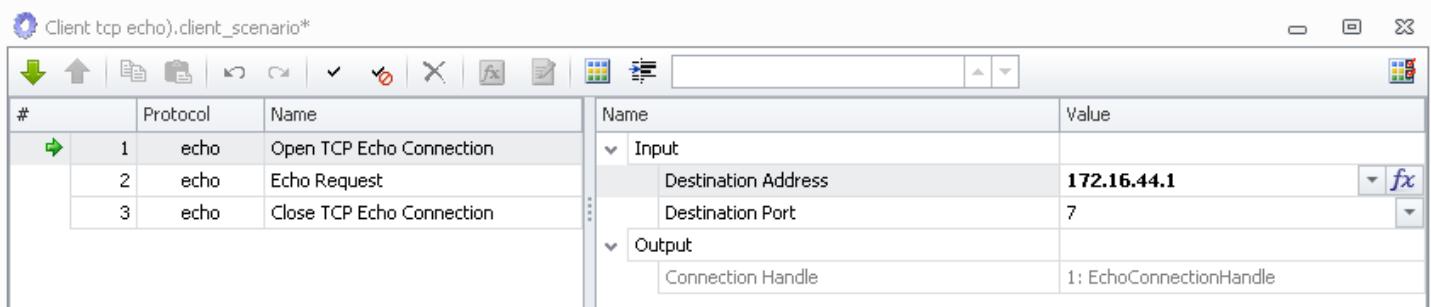


and three TCP and UDP ECHO and DISCARD Protocol Server Actions



These actions can be found in the Toolbox.

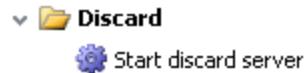
A simple TCP ECHO Protocol Project that opens a connection to the TCP Echo service, and then executes the TCP Echo Transmit Action and then closes the connection would look like



This Project would be useful in debugging connectivity to a target device that supports the ECHO Protocol and measuring round-trip times.

DISCARD Protocol

The DISCARD Protocol is defined as one of the services in the Internet Protocol Suite ([RFC 863](#)). The DISCARD Protocol is intended to be used as a testing, debugging and measurement tool. It is a service that listens on TCP port 9 and discards anything it receives. The Load DynamiX TDE and Appliance support a single Server Action that listens on port 9 and discards anything that is sent to it.



See [Reference: TCP/UDP Echo Protocols Commands List](#) for a detailed list of commands for these protocols

Appendix: Functions and Formula

Appendix: Functions and Formula

FUNCTIONS

Functions provide the ability to create values for Action input fields during the execution of a Project.

Action Input fields that support Functions will have the Function Editor button  enabled when focus is put on that field (by clicking into the field or doing a "mouse over" on the field). The output of any Function is either a String or a Number. Functions that generate a number in a input field that expects a String will generate a Number in String format.

Functions do not behave arithmetically unless used in a Formula (see below) or used individually to provide input to a field that is expecting an integer. For example. @ScenarioCounter() can be used as input to the Begin Loop Action counter input and its output is interpreted as an integer by the Begin Loop Action.

@RANDOM(0,100) + @RANDOM(1000,10000) does not produce an integer with a value somewhere between 1100 and 10100, it will produce a string like 19959 (where "1" comes from the first @RANDOM and 9959 comes from the second @RANDOM).

Function	Description
@STRING(<STRING>)	Produces a constant that can be used in any text field. As the name string implies, you enter a string of characters that is then used in a field.
@RANDOM(<MIN_VALUE>, <MAX_VALUE>, <STEP>, <PAD_LENGTH>) ; [shortcut = @RAND(...)]	Produces a value that can be used in numeric fields and in some cases text fields. Requires a minimum value and a maximum value. The result of an @RANDOM variable can be anything up to a 64-bit number; however, the number of digits in the output value is specified by the number of digits in the Max Value field. For example, setting a Min Value of 1 and a Max Value of 100 results in values from 001 to 100. Note that @RANDOM does not support negative numbers and the resulting number has as many characters in it as the Max Value does (e.g. Max Value = 100, output strings are 3 characters long). <STEP> and <PAD_LENGTH> are optional and default to 1 and 0, respectively (a 0 value for PAD_LENGTH sets the length of output strings = the length of the MAX_VALUE). See below for <STEP>.
@RANDOM64() ; [shortcut = @RAND64()]	Generates a random positive 64bit integer between 0 and 18,446,744,073,709,551,614.
@UP(<INDEX>,<COL_HDR>); @UP(<ALIAS>)	References an element of a Global User Parameter file.
@UPL(<COL_HDR>); @UPL(<ALIAS>)	References an element of the Local User Parameter file and can be used in Functions.
@LOOPINDEX(<BEGIN_ACTION>, <PAD_LENGTH>, <MAX_VALUE>); [shortcut = @LI(...)]	The current value of the index for the loop defined by the Begin Loop action specified in <BEGIN_ACTION>, a line # or <ALIAS>.

@LOOPTOTAL(<BEGIN_ACTION>, <PAD_LENGTH>, <MAX_VALUE>); [shortcut = @LT(...)]	The maximum value of the index for the loop defined by the Begin Loop action specified in <BEGIN_ACTION>, a line # or <ALIAS>.
@VARIABLE(<VARIABLE_DEFINITION_ACTION>); [shortcut = @VAR(...)]	A reference to the contents of a CREATE VARIABLE Action, by line # or <ALIAS>.
@OUTPUT(<HEADER_REFERENCE>); [shortcut = @OUT(...)]	A reference to the contents of an HTTP header received from to to set sent to an HTTP server.
@THREADINDEX(<THREAD_ACTION>, <PAD_LENGTH>, <MAX_VALUE>); [shortcut = @TI(...)]	The count of this particular instance of the thread executing in a loop defined by the Begin Thread action specified in <THREAD_ACTION> (line # or <ALIAS>).
@FORMULA(<REFERENCE> , <PAD_LENGTH>, <MAX_VALUE>);	Evaluate the Formula Action <REFERENCE> (line # or <ALIAS>), <MAX_VALUE> is the maximum value that is allowed from the evaluation of the Formula, <PAD_LENGTH> is ignored.
@TIME()	The number of microseconds that has elapsed since the beginning of the active stage of Project execution.
@DISTRIBUTION (<REFERENCE>) [shortcut = @DISTR(...)]	Evaluate the DISTRIBUTION Action at <REFERENCE> (line # or <ALIAS>).
@SCENARIOCOUNTER(<PAD_LENGTH>, <MAX_VALUE>) [shortcut = @SC(.....)]	A global Scenario ID assigned to every scenario instance at creation, a counter that grows sequentially from 1. This ID is unique within each Logical Port.
@CUSTOM(<C_FUNCTION>)	Evaluate the sub-function <C_FUNCTION> specified in the @CUSTOM definition. One of four sub-functions are possible: Local Scenario Counter, Restarted Scenario, Get Write Offset or Get Read Offset.
C_FUNCTION: Local Scenario Counter [shortcut = LSC]	A local Scenario ID assigned to this Scenario instance at creation, a counter that grows sequentially from 1. This ID is unique within each Scenario on the timeline.
C_FUNCTION: Restarted Scenario [shortcut = RST]	Returns a 1 if the current Scenario has been restarted or a 0 if it has not. See Load Profile for a discussion of Restart Scenario feature.
C_FUNCTION: Get Read Offset [shortcut = GRO]	Get the current value of the Read Offset. Get Read Offset must be accompanied by a line # or Alias reference to an Action controlled by the Set Auto Offset Action.
C_FUNCTION: Get Write Offset [shortcut = GWO]	Get the current value of the Write Offset. Get Write Offset must be accompanied by a line # or Alias reference to an Action controlled by the Set Auto Offset Action.

Function Inputs

Input	Description
-------	-------------

<PAD_LENGTH>	For Functions that produce strings that are numbers, <PAD_LENGTH> allows the user to specify the number of characters that the string will contain in all cases where the value of the output is character-wise smaller than the <PAD_LENGTH>. If the MAX_VALUE = 1000 (random numbers up to 1000) and the PAD_LENGTH were 2, the @RANDOM Function would produce strings of 01, 02, 03, 04, 05, 06, 07, 08, 09 whenever it generated a random number < 10 but for numbers >= 10, the strings would be just the number (10, 234, 789, etc) because the strings generated already have >= 2 characters.
<MAX_VALUE>	The maximum value produced for Functions that produce strings that are numbers.
<INDEX>	The zero-based index in the User Parameter Map for the UP file desired.
<COL_HDR>	Column header name (A through ZZ).
<MIN_VALUE>	The minimum value for output of the @RANDOM Function.
<STEP>	Guarantees that the delta between two consecutive numbers generated by an @RANDOM Function will never be less than the value of <STEP>.
<BEGIN_ACTION>	The line # or <ALIAS> of a Begin Loop Action.
<VARIABLE_DEFINITION_ACTION>	The line # or <ALIAS> of a CREATE VARIABLE Action.
<STRING>	Any set of characters including separators like "", "." and space. Not allowed are Carriage Return, Line Feed and Right Parenthesis.
<HEADER_REFERERNCE>	A reference to specific HTTP Header items in a specified line in a Scenario. Ex, @OUTPUT(2.1) would refer to the first Header defined in Line 2 of the Scenario. Only used with the @OUTPUT Function.
<THREAD_ACTION>	The line # or <ALIAS> of a Begin Thread Action.
<ALIAS>	ALIAS defined in a User Parameter File (see Advanced Concepts: User Parameters for details) or in a Scenario Control Action (Begin loop or Formula or Begin Thread or Create Variable or ... (see Advanced Concepts: Variables and Aliases for details)).
<REFERENCE>	Line number of an Action in a Scenario.

MAX_VALUE, PAD_LENGTH examples

MAX VALUE	PAD LENGTH	OUTPUT
1500	3	001, 099, 100, 1237
1500	0	0001, 0099, 0100, 1237
1500	1	1, 99, 100, 1237
1500	5	00001, 00099, 00100, 01237

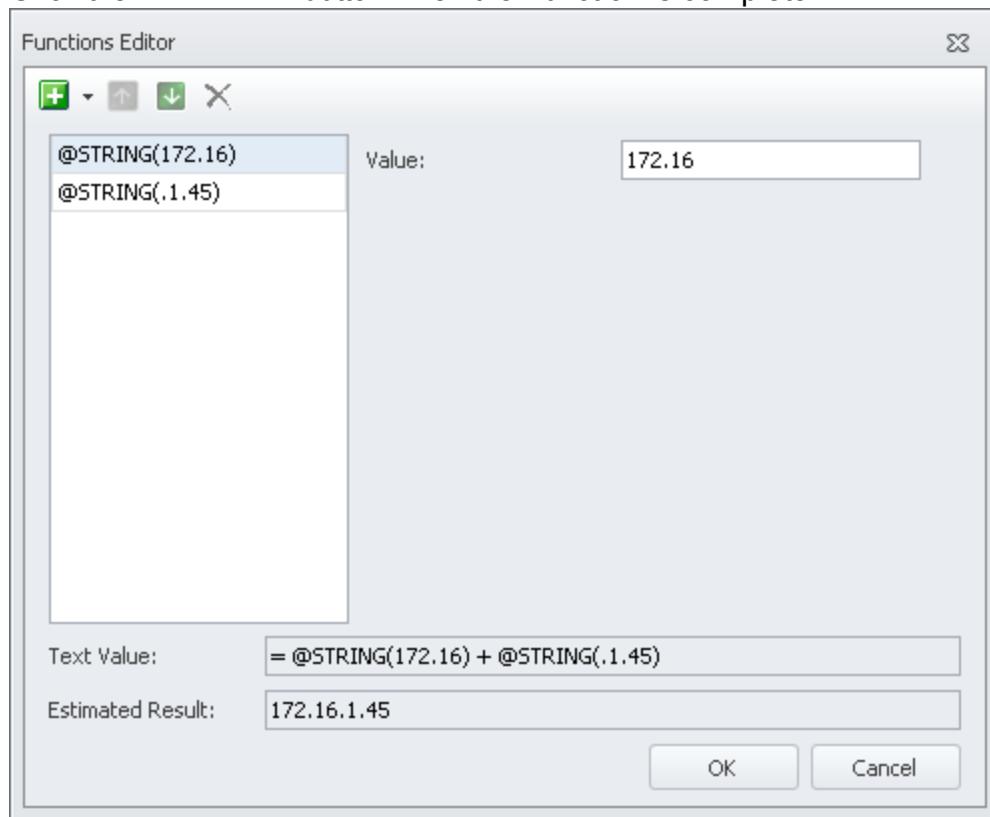
To place a Function in an input field, select the Function Editor by clicking on the input field in a Scenario, then clicking the Function Editor button  that appears on the right edge of the input field. Note that not all input fields support Functions. Input fields that have fixed values such as True/False

or some other predefined values, do not support Functions.

Input		
Connection Handle		Default
LUN	= @UP(4,C)	
LBA	0	

Click the down arrow selector to the right of the Add button and select Add STRING, Add RANDOM, Add UP, Add SCENARIOCOUNTER, Add LOOPTOTAL, Add LOOPINDEX or Add VARIABLE,. Provide content for the function by filling in the fields appropriate to that type of function (e.g. fill in the Value field for a String function with text). Edit an existing function by clicking the function to be edited and modifying the associated values in the right-hand portion of the Function Editor pane.

Click the button when the Function is complete.



Function Examples:

- =@RANDOM(1,150, 1) could be used to specify IP Port numbers 001 through 150.
- =@STRING(File) + @RANDOM(1,150,1) produces file names File001 through File150 in random order.
- =@STRING(USER) + @RANDOM(1,150,1) produces user names USER001 through USER150 in random order.
- =@STRING(Pass) + @RANDOM(1,150,1) produces passwords Pass001 through Pass150 in random order.
- =@UP(0,A)+@RANDOM(1,150,1) produces output = (next element of column A)001- 150 in random order.
- =@LOOPINDEX(3) produces the current value of the index for the loop defined by the Begin Loop Action in line #3.
- =@SCENARIOCOUNTER() produces the number of the current Scenario instance (counting starts at 1).
- =@LOOPTOTAL(3) produces the value of the Repeat Count field of the Begin Loop Action in line #3.

- `=@VARIABLE(5)` produces the contents of the VARIABLE created in line #5.
- `=@UPL(A)` references column A of the Local User Parameter file and can be used in Functions.
- `=@THREAD_INDEX()` produces the counter value of the instance of the referenced Thread (0 if referenced outside of a Thread block).
- `=@TIME()` produces the number of microseconds that have elapsed since the Project began execution.
- `=@DISTR(distri)` produces the value output by the evaluation of a DISTRIBUTION Action with the Alias of "distri".
- `=@CUSTOM(LSC)` produces the current value of the Local Scenario Counter for this Scenario.

The `@STRING`, `@SCENARIOCOUNTER` and `@RANDOM` functions may be embedded in User Parameter files.

See [Advanced Concepts: Variables and Aliases](#) for more details on `@VARIABLE()`, [Advanced Concepts: User Parameters](#) for more details on Column Aliases, `@UP()` and `@UPL()`.

Function Notes:

- Functions do not produce Integer values
- `@LOOPINDEX` and `@LOOPTOTAL` may not be embedded in User Parameter files.
- An `@STRING()` function containing text, used in input fields that expects a number, returns a "0".
- Local User Parameter references (e.g. `$(A)`) may not be used in Functions (use `@UPL()`).
- `@VARIABLE()` may not be used in the Open <Prot> TCP Connection Actions for FC, SMB, SMB2, NFS, HTTP, iSCSI and KERBEROS protocols.
- Function references that produce String type values in an input field that expects an Integer will yield 0 as input
- **@RANDOM Notes:**
 - `@RANDOM()`, `@RANDOM(,)`, `@RANDOM(, ,)` and `@RANDOM("xyz", "pqr", "abc")` all produce the same kind of output as `@RANDOM64()` - a random 64 bit integer.
 - `@RANDOM(0,0)` always produces a value of 0.
 - `@RANDOM(0,000x)` is the same as `@RANDOM(0,x,4)`.

FORMULA

A Formula Action produces a mathematical result based on inputs to the algorithm contained in the Formula Action. The output of Formula Actions can be used as input to other Load DynamiX Actions that require integers as input. Formulas operate on and produce 64bit integers. For example, a Read or Write Action that requires an Offset or Block Number to be calculated based on other inputs or results.

#	Protocol	Name
1	SMB2	Open SMB2 TCP Connection
2	SMB2	Negotiate
3	SMB2	Session Setup
4	SMB2	Tree Connect
5	SWT	Create Variable oneKB
6	SMB2	Create File
7	SWT	Create Variable TwoHundred
8	SWT	Begin Loop MainLoop
9	SWT	Formula WriteOffset
10	SMB2	File Write
11	SWT	End Loop
12	SMB2	File Close
13	SWT	Formula TotalBytesWritten
14	SMB2	File Read
15	SMB2	File Close
16	SMB2	Tree Disconnect
17	SMB2	Logoff

Name	Value
Input	
File Handle	Default
Credits Charged	0
Credits Requested	1
Automatic Offset	True
Bytes Per Block	=@FORMULA(WriteOffset)
Remaining Bytes to Write	0
Bytes Total	=@VARIABLE(oneKB)
Block Sequence	Forward
Data Content	
Data Source	::SeededRandom(0)
Data Source Offset	0
Output	
Status Code	
Response Handlers	
Completion Status	
Scenario Impact	

An empty Formula Action is of the form:

Name	Value
Input	
Alias	F1
Description	
Formula	✖
Output	
Output	22: Formula

Required Inputs

Formula : the algorithm elements being calculated such as
width * length or
blockoffset / blocklength or
writepermissions & readpermissions

Optional Inputs

Alias : a name that the Formula may be referred to by (in this case F1 - @FORMULA(F1)).

Aliases must be Alpha-numeric, start with a letter and may contain "_"

Description : which can be used to document the formula

For more details on Aliases see [Advanced Concepts: Variables and Aliases](#).

To create a Formula Action, give the Formula an (optional) Alias, a (optional) Description and enter the formula on the formula line. To delete an algorithm element that is no longer required, first delete it from the algorithm then delete any value that it has been given.

The following simple Formula demonstrates the use of a Formula Action to calculate the area of a rectangle:

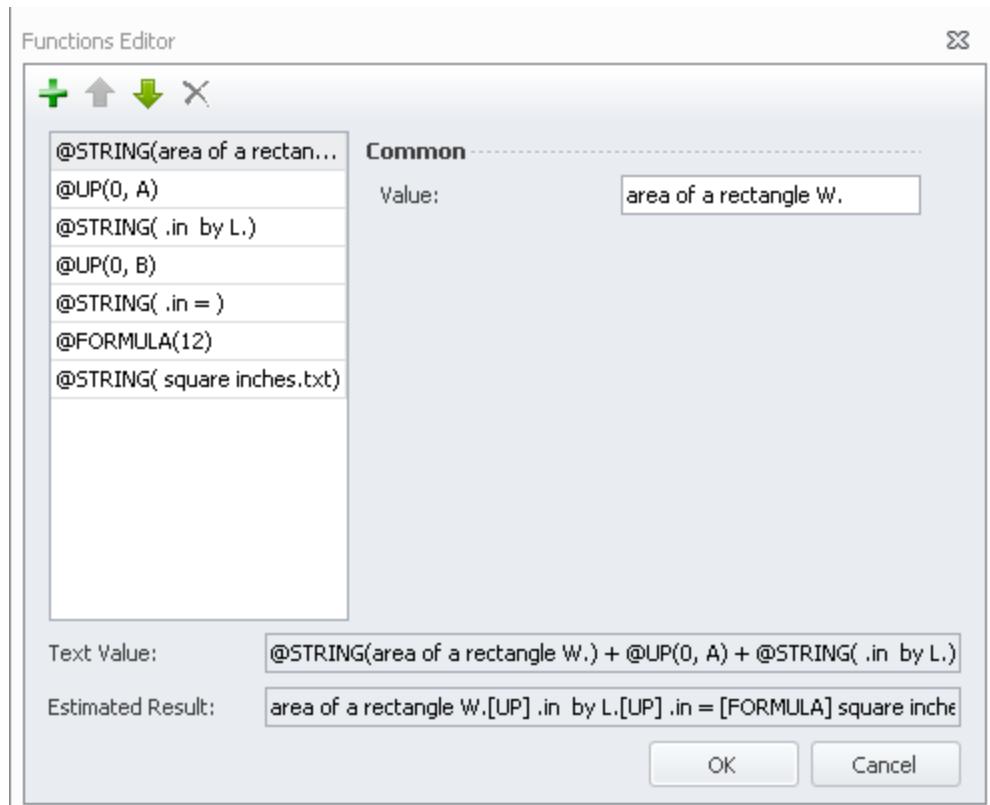
When the formula for the area of a rectangle was typed in as width * length, the Formula Action automatically created the two inputs: Length and Width. In this case, the length and width values are coming from Variables in lines 8 and 10.

When this Formula Action is executed, it returns the 64 bit integer value that is the result of multiplying width times length. The SMB2 client below shows how this Formula Action can be used to calculate the area of a rectangle and use that value in the name of a file to be created.

There are 8 width and length pairs used as inputs to the Variables in lines 8 and 10 from a User Parameter file

A:width	B:length	
1	10	
4	4	
193	2	
16	16	
1kb	1kb	
67	99	
1gb	1kb	
12	12	
*		

When the Client is executed it creates 8 files with the name of the form "area of a rectangle W.<UP value>.in by L.<UP value>.in = <FORMULA result> square inches.txt" which is created using the Function Editor to create the input to the Variable in line 14.



In the PCAP file that captures the execution of this client, we can see the file names as the files are created (highlighted in red below).

```

304 Create Request File: area of a rectangle w.1.in by L.10.in = 10 square inches.txt
211 Create Response File: area of a rectangle w.1.in by L.10.in = 10 square inches.txt
146 Close Request File: area of a rectangle w.1.in by L.10.in = 10 square inches.txt
182 Close Response
302 Create Request File: area of a rectangle w.4.in by L.4.in = 16 square inches.txt
211 Create Response File: area of a rectangle w.4.in by L.4.in = 16 square inches.txt
146 Close Request File: area of a rectangle w.4.in by L.4.in = 16 square inches.txt
182 Close Response
308 Create Request File: area of a rectangle w.193.in by L.2.in = 386 square inches.txt
211 Create Response File: area of a rectangle w.193.in by L.2.in = 386 square inches.txt
146 Close Request File: area of a rectangle w.193.in by L.2.in = 386 square inches.txt
182 Close Response
308 Create Request File: area of a rectangle w.16.in by L.16.in = 256 square inches.txt
211 Create Response File: area of a rectangle w.16.in by L.16.in = 256 square inches.txt
146 Close Request File: area of a rectangle w.16.in by L.16.in = 256 square inches.txt
182 Close Response
320 Create Request File: area of a rectangle w.1kb.in by L.1kb.in = 1048576 square inches.txt
211 Create Response File: area of a rectangle w.1kb.in by L.1kb.in = 1048576 square inches.txt
146 Close Request File: area of a rectangle w.1kb.in by L.1kb.in = 1048576 square inches.txt
182 Close Response
310 Create Request File: area of a rectangle w.67.in by L.99.in = 6633 square inches.txt
211 Create Response File: area of a rectangle w.67.in by L.99.in = 6633 square inches.txt
146 Close Request File: area of a rectangle w.67.in by L.99.in = 6633 square inches.txt
182 Close Response
332 Create Request File: area of a rectangle w.1qb.in by L.1kb.in = 1099511627776 square inches.txt
211 Create Response File: area of a rectangle w.1qb.in by L.1kb.in = 1099511627776 square inches.txt
146 Close Request File: area of a rectangle w.1qb.in by L.1kb.in = 1099511627776 square inches.txt
182 Close Response
308 Create Request File: area of a rectangle w.12.in by L.12.in = 144 square inches.txt
211 Create Response File: area of a rectangle w.12.in by L.12.in = 144 square inches.txt
146 Close Request File: area of a rectangle w.12.in by L.12.in = 144 square inches.txt

```

Notice that integer shorthand like 1kb and 1gb are supported as input to the Formula Action.

Formula Action Operators:

The following are the supported Formula Operators:

Operator Group	Operator Name	Operator Symbol	Example	Notes
Arithmetic	Plus	+	$2 + 5 = 7$	
	Minus	-	$5 - 3 = 2$	
	Divide	/	$9 / 4 = 2$	no fractions supported
	Modulo	%	$9 \% 2 = 1$	division remainder
	Multiply	*	$3 * 2 = 6$	
Bitwise	And	&	$12 \& 6 = 4$	and of 0x1100 and 0x0110 = 0x0100
	Or		$12 6 = 14$	or of 0x1100 and 0x0110 = 0x1110
	Xor	^	$12 ^ 6 = 10$	xor of 0x1100 and 0x0110 = 0x1010
	Not	~	~ 12	= 0x111111...1111110011 (inverts bits of a 64bit integer)
	Shift Right	>>	$127 >> 5 = 3$	0x01111111 shifted right 5 = 0x0011
Comparison	Shift Left	<<	$3 << 5 = 96$	0x0011 shifted left 5 = 0x01100000
	Less Than	<	$3 < 5 = 1$	True
	More Than	>	$3 > 5 = 0$	False
	Equal	==	$3 == 5 = 0$	
	Not Equal	!=	$3 != 5 = 1$	
Logical	Less Than or Equal	<=	$3 <= 5 = 1$	
	More Than or Equal	>=	$3 >= 5 = 0$	
	Logical And	&&	$3 \&& 0 = 0$	False
	Logical Or		$3 0 = 3$	True
All Others = True	Logical Not	!	$!0 = 1$ $!3 = 0$	

Formula Action Precedence:

Precedence	Operator	Description	Associativity
3	+ -	Unary Plus and Minus	Right-to-Left
	! ~	Unary Logical NOT and Bitwise NOT	
5	* / %	Multiplication, Division, Modulo	Left-to-Right
6	+ -	Addition and Subtraction	
7	<< >>	Bitwise Shift Left and Shift Right	
8	< <=	Comparison Less Than and Less Than or Equal	
	> >=	Comparison More Than and More Than or Equal	
9	== !=	Comparison Equal and Not Equal	
10	&	Bitwise AND	
11	^	Bitwise XOR	
12		Bitwise OR	
13	&&	Logical AND	
14		Logical OR	

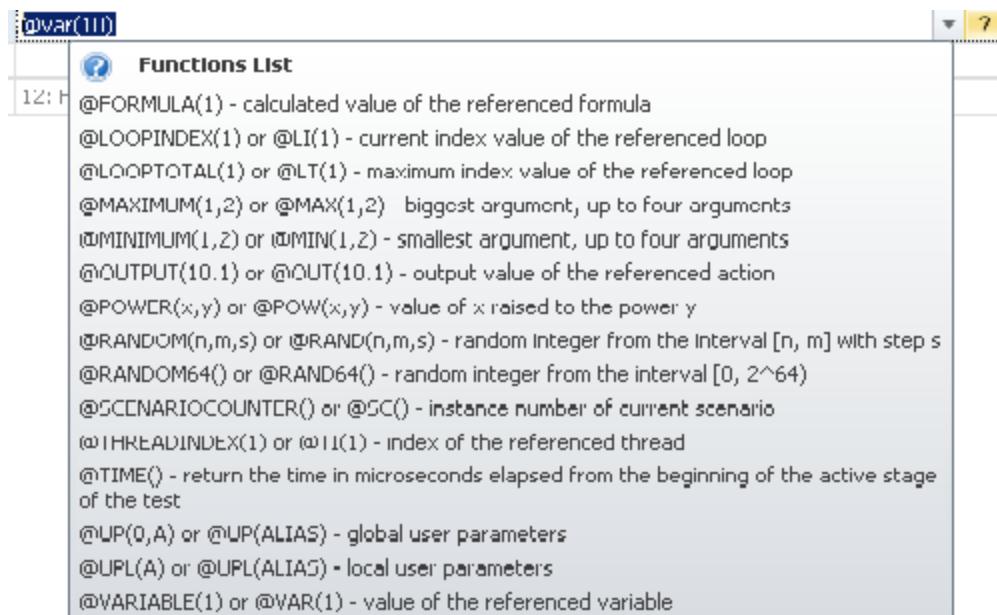
Mathematical Functions

Function	Description
POWER(<X>,<Y>) [shorthand = pow(<X>,<Y>)]	Raise <X> to the power <Y>

MAXIMUM(<V1>, <V2>, <V3>, <V4>) [shorthand = max(<V1>, <V2>, <V3>, <V4>)]	Find the highest value among <V1>, <V2>, <V3>, <V4> (2 to 4 values supported)
MINIMUM(<V1>, <V2>, <V3>, <V4>) [shorthand = min(<V1>, <V2>, <V3>, <V4>)]	Find the lowest value among <V1>, <V2>, <V3>, <V4> (2 to 4 values supported)

Formula Function Help

The Formula input fields that display the  contain information as to the set of supported functions. Click the  and you will see the table below.



Formula Action Example:

Increasing Block Size write: the SMB2 client below writes 1KB of data to a file at the Block Offset defined by the value of the Variable in line 6. The Formula Action in line 20 multiplies the current block offset (the value of the Variable in line 6) * 1kb (the value of the Variable in line 8). The Variable in line 6 is updated at the bottom of the loop by assigning it the value of the Formula Action in line 20.

Write with increasing block # based on last block #.client_scenario

#	Protocol	Name
1	SMB2	Open SMB2 TCP Connection
2	SMB2	Negotiate
3	SMB2	Session Setup
4	SMB2	Tree Connect
5	#	baseblock offset
6	SWT	Create Variable
7	#	block offset increment
8	SWT	Create Variable
9	#	loop counter
10	SWT	Create Variable
11	SWT	Begin Loop
12	#	create file name string including new ...
13	SWT	Create Variable
14	SMB2	Create File
15	#	write 1kb at var(6) offset
16	SMB2	File Write
17	#	close file and delete
18	SMB2	File Close
19	#	new blk offset = old blk offset * incr...
20	SWT	Formula increasingblkoffset
21	#	change base offset to new offset
22	SWT	Update Variable [6]
23	SWT	End Loop
24	SMB2	Tree Disconnect

Name	Value
Input	
Alias	increasingblkoffset
Description	increase block offset = last block offset * var(8)
Formula	<pre>baseblockoffset * blockoffsetincrement @VAR(6)</pre>
Output	
Output	20: Formula

The PCAP file from the execution of this project shows how the block offset increases in each successive Write Action. First Write has a Block Offset value of 1024 (1KB). The second has a Block Offset value of 1048576 (1KB * 1KB). The third Write has a Block Offset value of 1073741824 (1048576 * 1KB). Etc.

```
211 Create Response File: file. write offset = 1kb .txt
1194 Write Request Len:1024 off:1024 File: file. write offset = 1kb .txt
139 Write Response
146 Close Request File: file. write offset = 1kb .txt
182 Close Response
244 Create Request File: file. write offset = 1048576 .txt
211 Create Response File: file. write offset = 1048576 .txt
1194 Write Request Len:1024 off:1048576 File: file. write offset = 1048576 .txt
139 Write Response
146 Close Request File: file. write offset = 1048576 .txt
182 Close Response
250 Create Request File: file. write offset = 1073741824 .txt
211 Create Response File: file. write offset = 1073741824 .txt
1194 Write Request Len:1024 off:1073741824 File: file. write offset = 1073741824 .txt
139 Write Response
146 Close Request File: file. write offset = 1073741824 .txt
182 Close Response
256 Create Request File: file. write offset = 1099511627776 .txt
211 Create Response File: file. write offset = 1099511627776 .txt
1194 Write Request Len:1024 off:1099511627776 File: file. write offset = 1099511627776 .txt
139 Write Response
146 Close Request File: file. write offset = 1099511627776 .txt
182 Close Response
262 Create Request File: file. write offset = 1125899906842624 .txt
211 Create Response File: file. write offset = 1125899906842624 .txt
1194 Write Request Len:1024 off:1125899906842624 File: file. write offset = 1125899906842624 .txt
139 Write Response
146 Close Request File: file. write offset = 1125899906842624 .txt
182 Close Response
268 Create Request File: file. write offset = 1152921504606846976 .txt
211 Create Response File: file. write offset = 1152921504606846976 .txt
1194 Write Request Len:1024 off:1152921504606846976 File: file. write offset = 1152921504606846976 .txt
139 Write Response
146 Close Request File: file. write offset = 1152921504606846976 .txt
```

Formula Action Notes:

- All operations are on 64 bit Unsigned Integers
- Integer shorthand like 1KB for 1024 is supported. See [Reference: Action Input Shorthand](#) for more details on integer shorthands.
- To delete an algorithm element that is no longer required, first delete it from the algorithm then delete any value that it has been given.

Appendix: Max TCP Open Connections and Open Rate

Finding the Maximum Number and Maximum Rate of Open TCP Connections

Abstract

There is always a limit on the number of open TCP connections that a Device Under Test (DUT) can maintain simultaneously and a limit to how fast the DUT can open new connections. These are two very important pieces of information because they will be useful when designing performance tests to be executed by the Load DynamiX Appliance. They may also be useful in marketing or sales literature for the DUT.

This section shows the Tester how to determine these limitations for their DUT using the Load DynamiX Appliance and software.

Introduction

NMAX – the maximum number of open TCP connections.

RMAX – the maximum rate at which TCP connections can be opened.

Both of these limitations are indicative of the behavior that clients of the DUT will observe when a high volume of connections is attempted.

These parameters can vary widely depending on the role of the DUT. For example, a backup storage device might not support high values for NMAX and RMAX, whereas a high end file server should have very high values for both RMAX and NMAX.

Maximum Number of TCP Connections

Every TCP based network storage protocol session begins with opening a TCP connection. This demonstration will use the NFSv3 protocol but it could easily be done with the other protocols that Load DynamiX supports (CIFS-SMB, SMB2, NFSv4, HTTP or iSCSI).

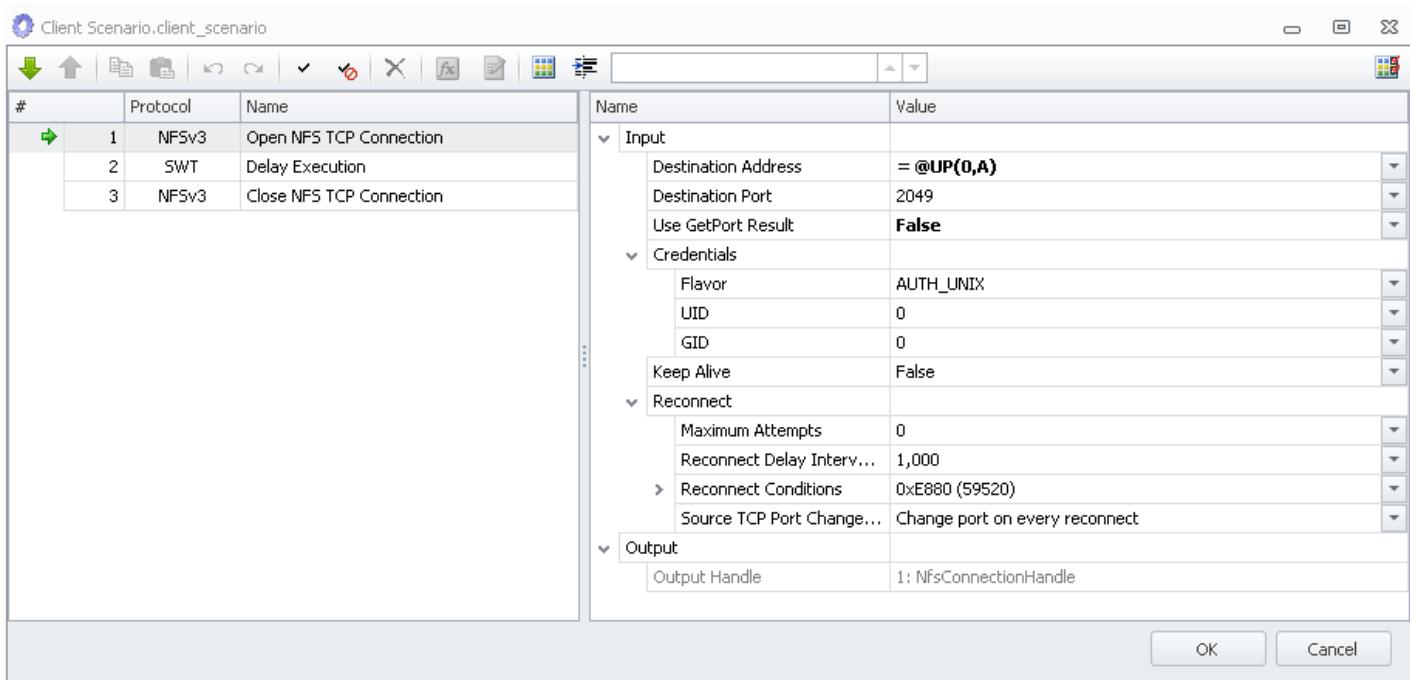
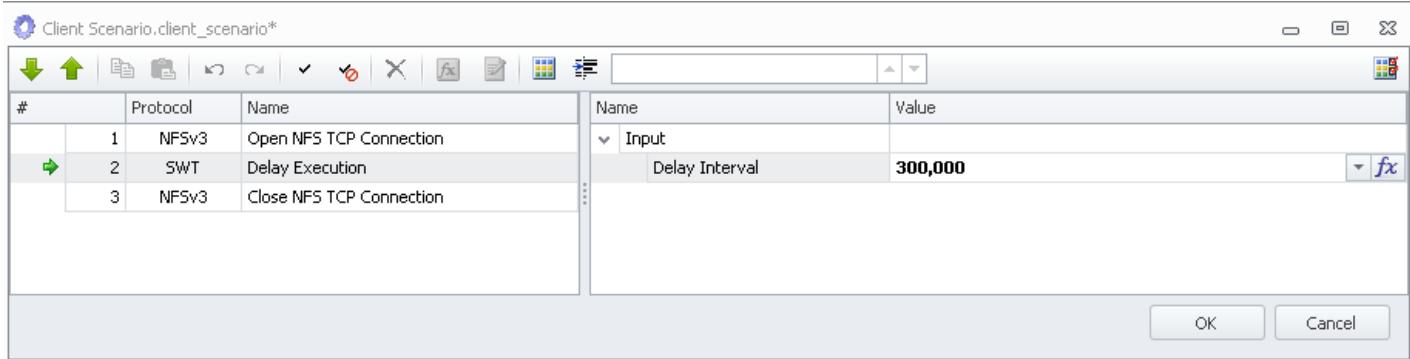
The demonstration will open TCP connections with the DUT at a relatively high rate and keep those connections open. After some time, the number of open connections will reach NMAX and the DUT will start rejecting new connections or closing existing ones and opening new ones while maintaining the number of open connection at the NMAX level.

To implement this demonstration using the Load DynamiX Appliance and TDE, first, create a Project Scenario which does the following:

- Establish a TCP connection to a DUT
- Idle for 300 sec
- Close the connection

Set the Client Load to 10,000 new scenarios per second in the Load Profile so that in 300 seconds, 300,000 connections will have been opened provided the DUT is capable of doing so. The demonstration will use the range of 254 different IP addresses configured in the Client Network Profile so that there will be sufficient IP and TCP port connections for the demonstration. The target for the demonstration is the NFSv3 server at IP address 172.16.0.7. A link to NFSv3 protocol reference material is provided in the [References and Terminology section](#).

Figures 1A and 1B below show the Scenario TCP Open and Delay Execution Action contents.

**Fig. 1A: Client Scenario, Open NFS TCP Connection Action****Fig. 1B: Delay Execution Action**

Client Network Profile are shown in Fig. 3 below:

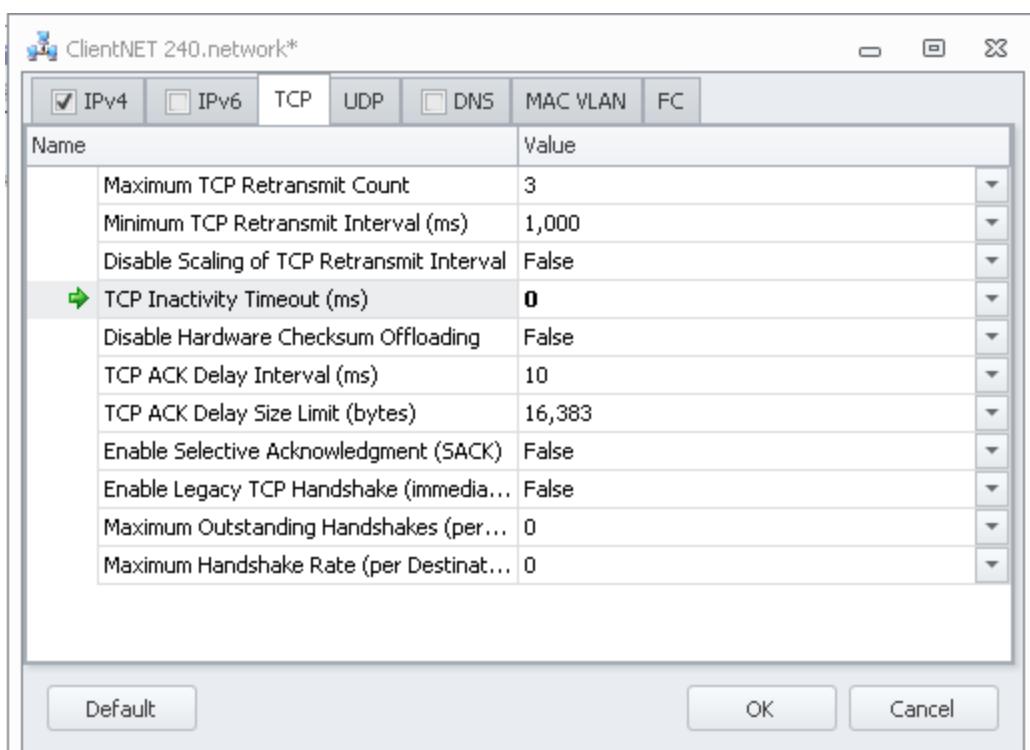


Fig. 3: Client Network Profile TCP tab

Important: In order for Load DynamiX to maintain the connection without timing out, set the Network Profile TCP tab TCP Inactivity Timeout to "0"..

Results of the test for a commercial NFSv3 file server are shown in the Load status graph, Fig. 4, from the Result Explorer. This graph shows that NMAX is around 56,000. After 56,000 are open, newly created connections are rejected at an escalating rate which is less than the creation rate but 56000 is the NMAX for this server.

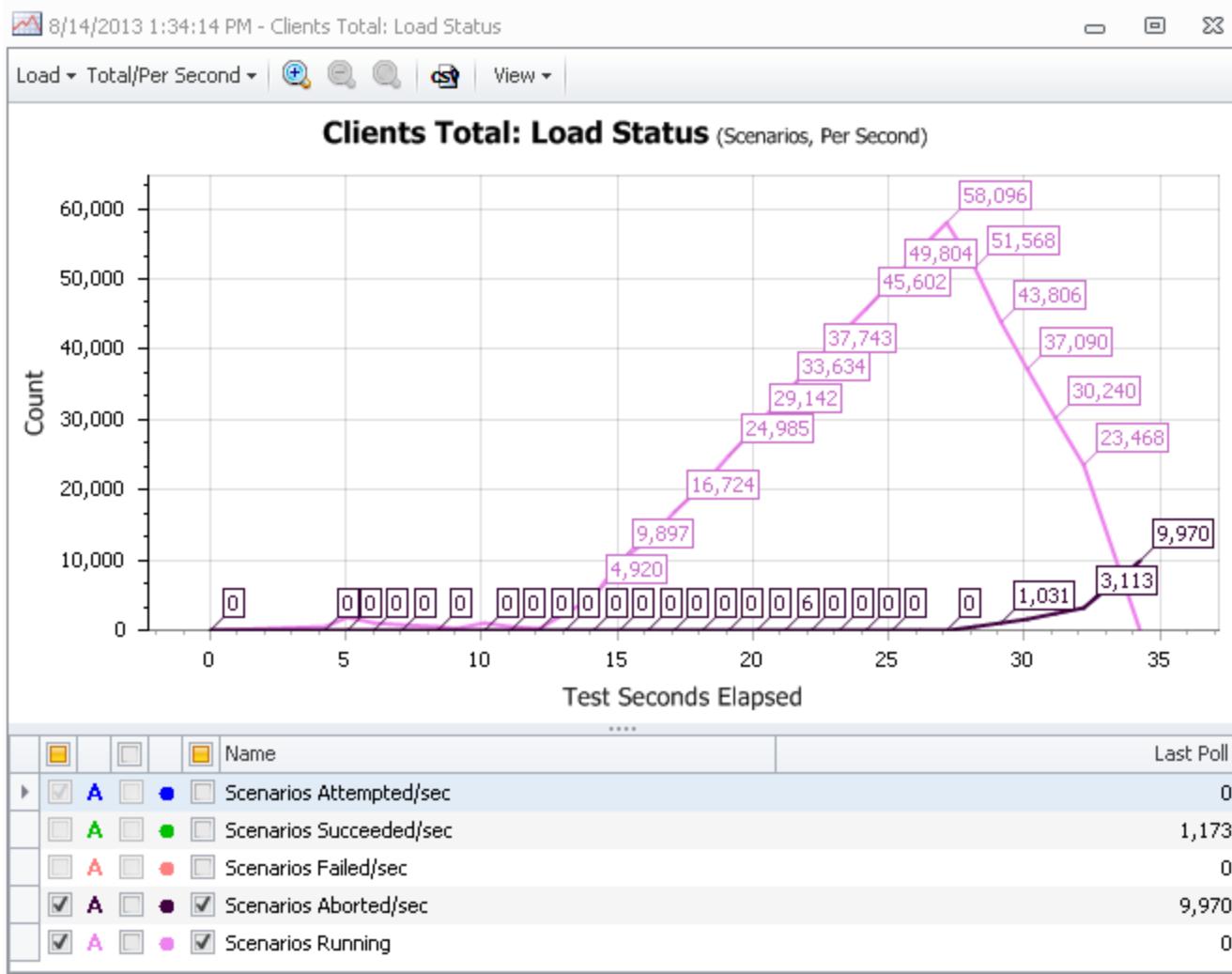


Fig. 4: NMAX for a File Server

Maximum Rate of Opening TCP Connections

In order to determine the maximum rate of opening TCP connections, RMAX, the Delay Timeout setting in Delay Execution Action is changed from 300,000 ms (Fig. 1B) to 1 ms so that TCP Connections will be opened as fast as is possible..

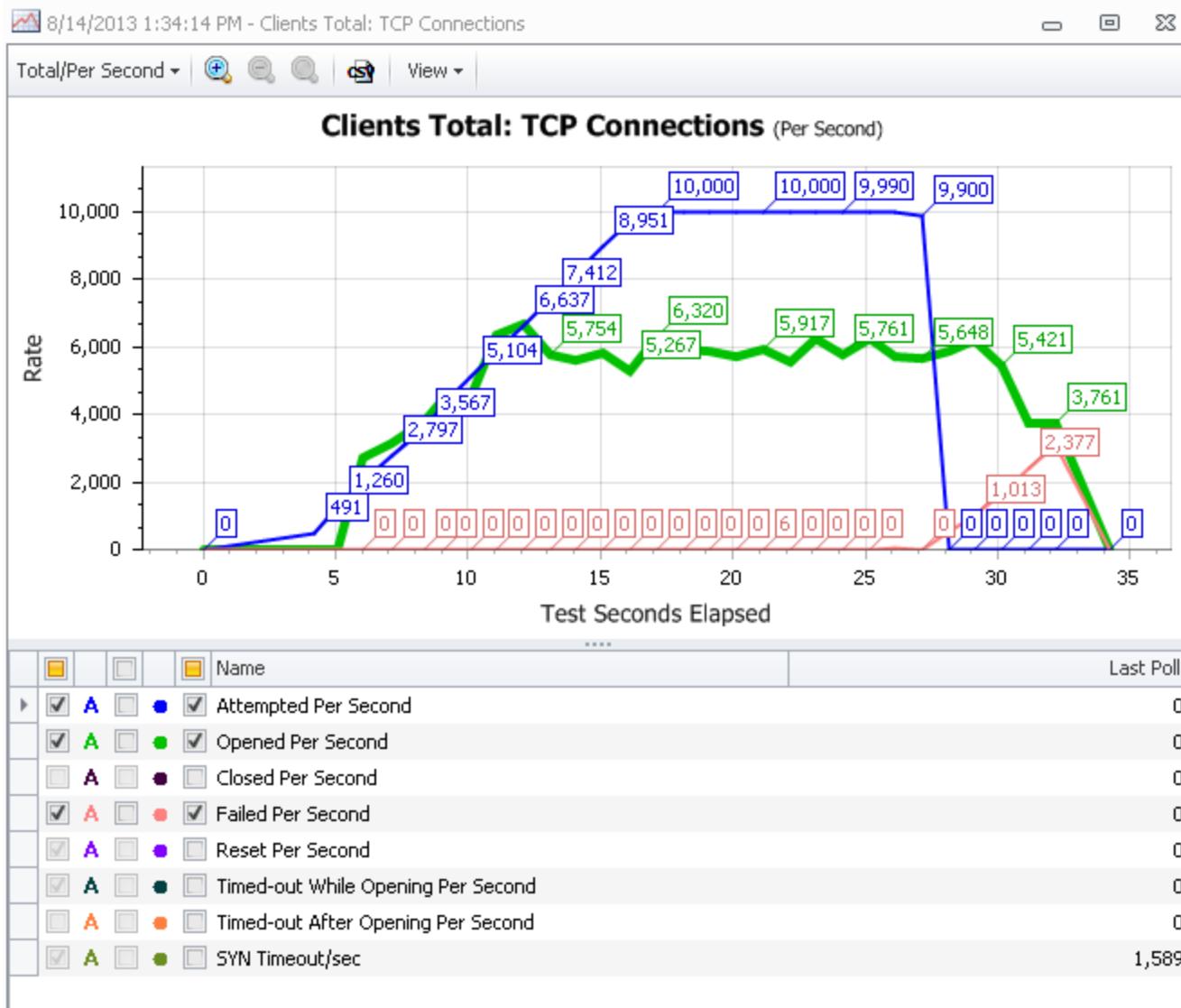


Fig. 6A: TCP Connections per Second

Setting the Delay Timeout to 1 ms results at a rate of 10,000 new connections per second shows that the DUT can handle about 5500 connections per second before it begins to have trouble handling the load.

It is clear from Fig. 6A that at the rate of approximately 5500 new connections per second, the DUT starts experiencing difficulties in handling incoming connections. Figs. 6B and 6C provide more insight into what is happening.

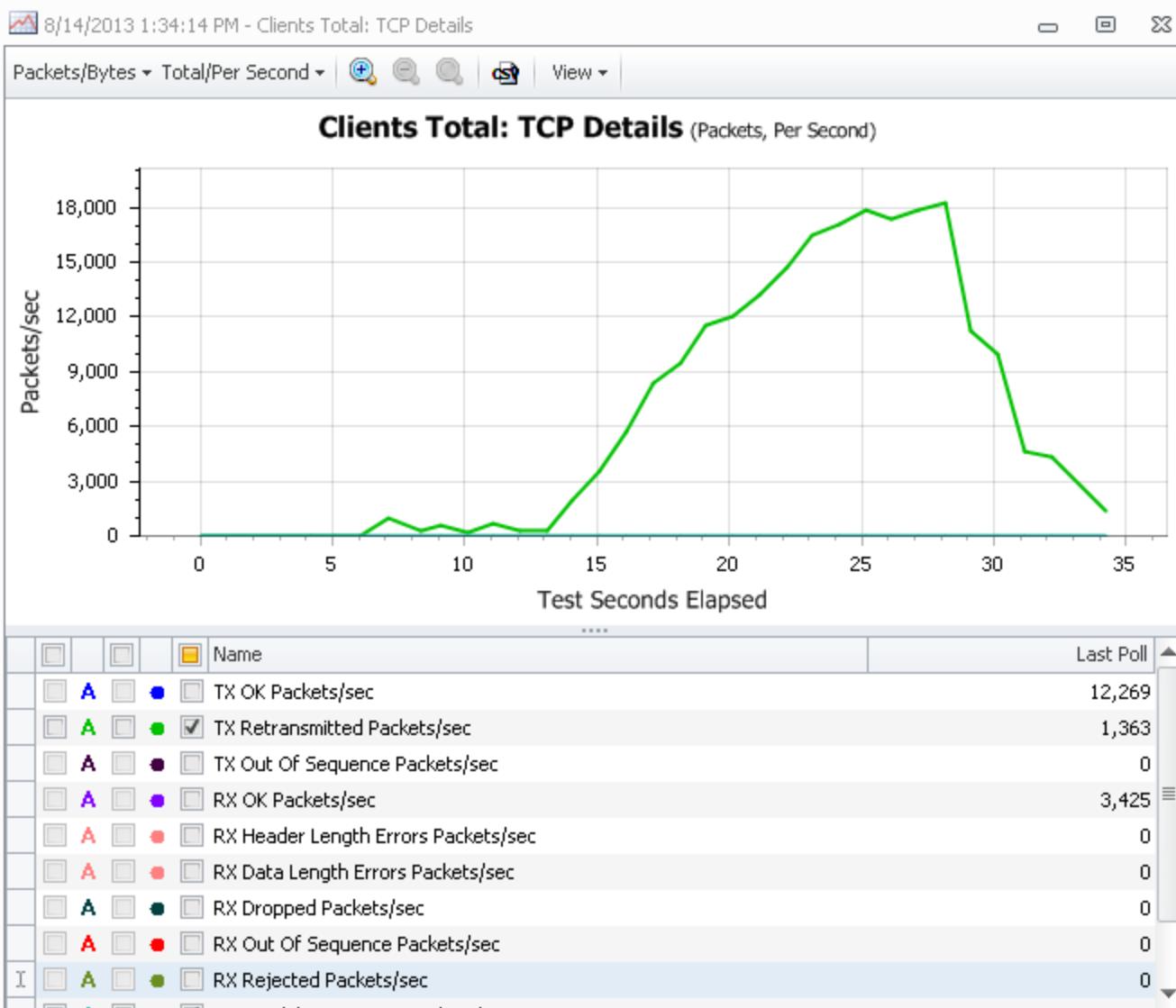


Fig. 6B: TCP Tx Retransmits and Rx Duplicates

At approximately 13 seconds into Project execution, TCP packet retransmission starts growing rapidly, indicating an overload of the DUT's ability to receive and reply to packets from the client. Also at ~13 seconds, the time it takes to send the open request and receive the first byte of the response triples (110ms to 321ms).

Both of these results indicate a decrease in DUT responsiveness.

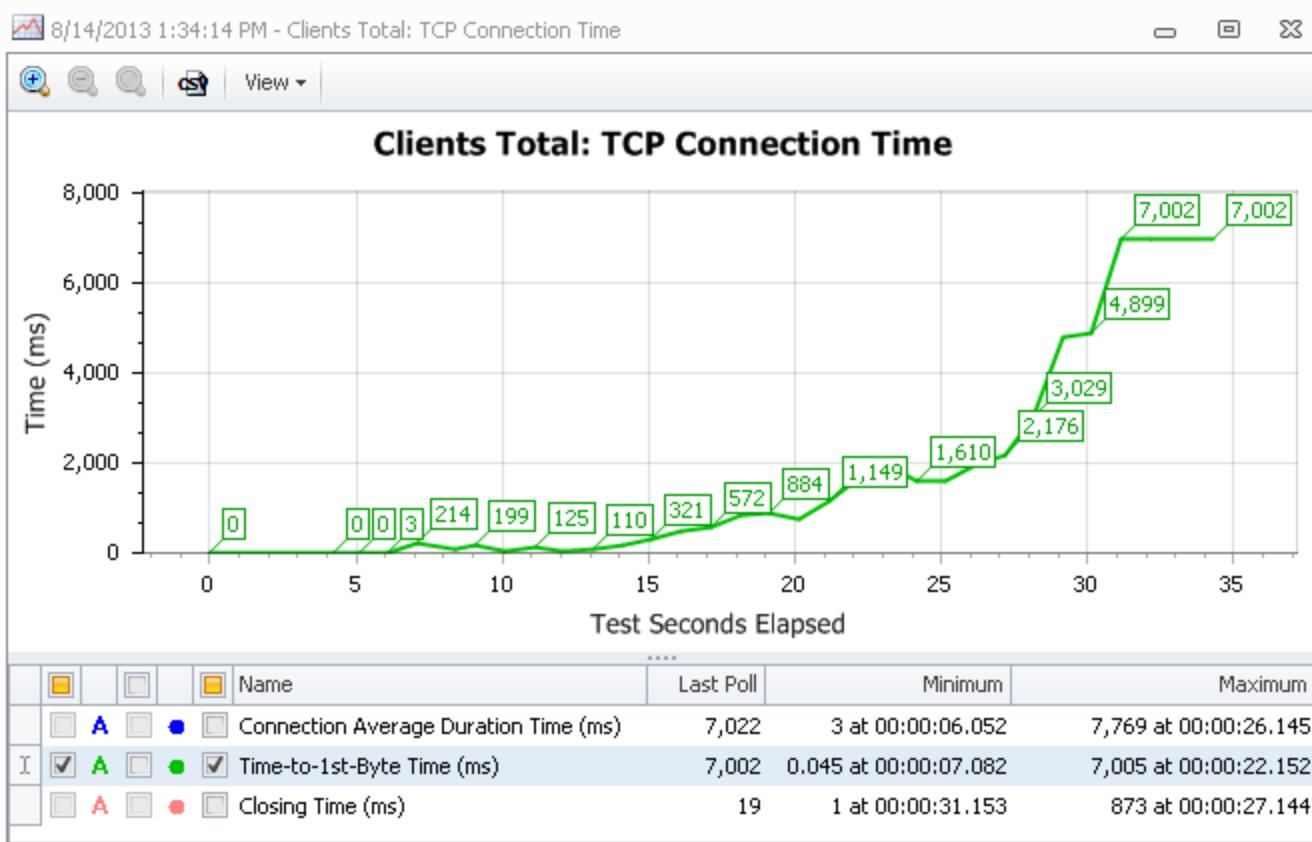


Fig. 6C: TCP Connection Time to First Byte

Charts in Figs. 6A, 6B and 6C are all standard TCP reports from the Results folder in the Load DynamiX TDE GUI.

From these three reports it can be seen that a new TCP connection per second rate greater than 5000 per second produces negative behavior from the DUT. A load rate of 5000 new connections per second produces a stable response from the DUT, easily handling the 5000 new connections per second (Fig. 7A below) with an average Time to First Byte of 50 micro-seconds and zero packet retransmissions.

5000 can then be considered as the RMAX value for this DUT.

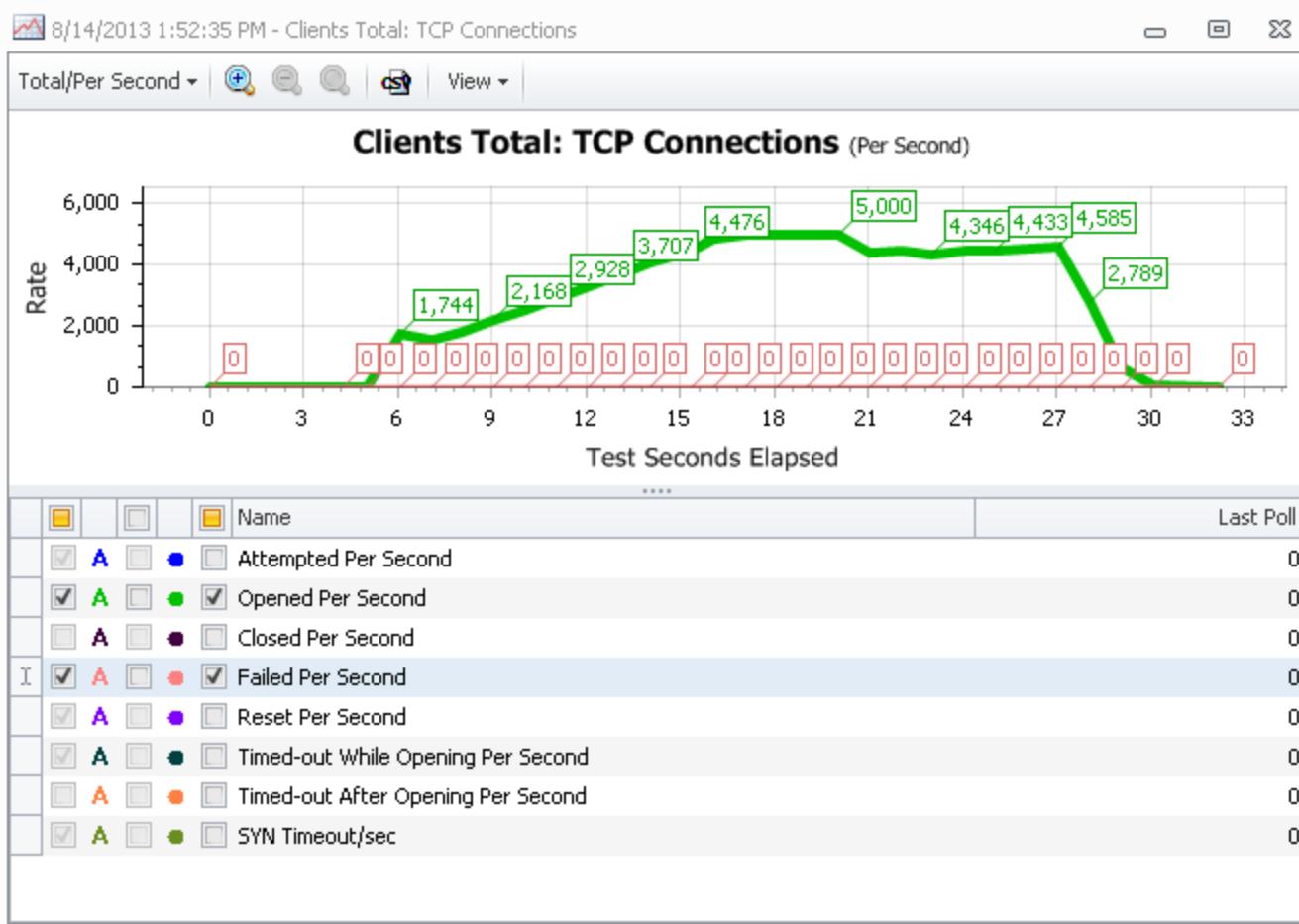


Fig. 7A: New TCP Connections per Second

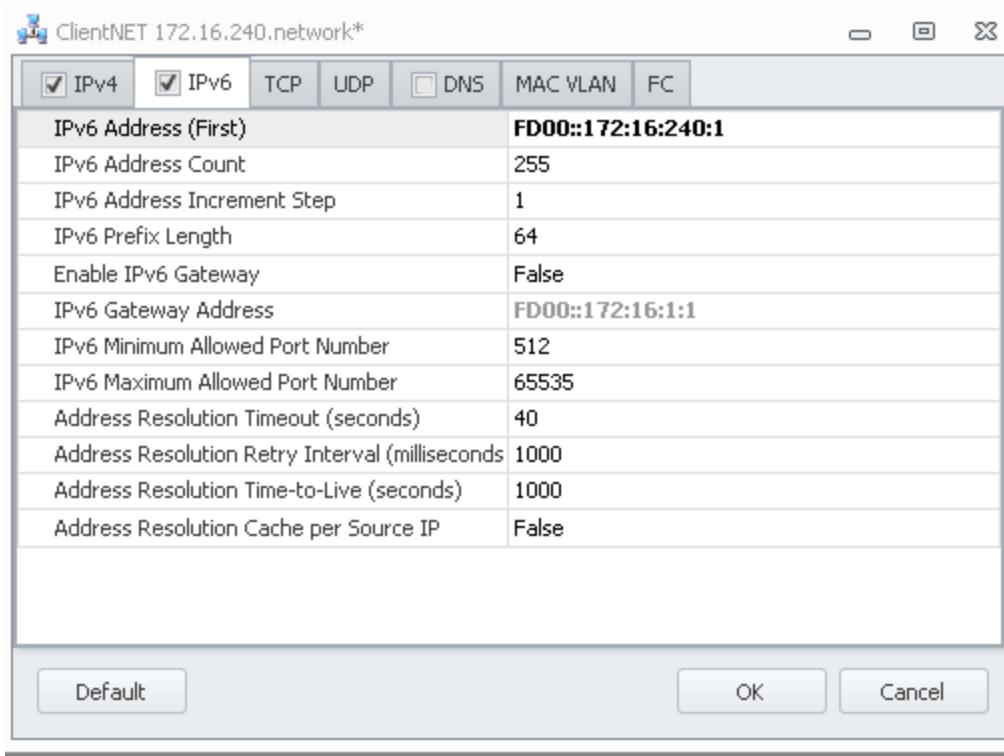
Conclusions

Both the maximum number of concurrently open connections (NMAX) and the maximum rate at which a server can open TCP connections (RMAX) can be easily benchmarked using the Load DynamiX appliance. The Project demonstrates how easy it is to determine important information regarding the connection breaking point for a DUT. These parameters are important characteristics of the DUT. They should be retested periodically to catch any degradation of performance due to the new changes and modifications to the system.

Appendix: IPv6

Appendix: IPv6

As described in the [Test Creation chapter](#), the Client and Server IP address configurations are defined in the Network Profile Resource;



If the Network Profile associated with a Client Scenario specifies only IPV4 addressing then that Client will operate in an IPv4 Address space (i.e. all Clients started by that Scenario will have IPv4 addresses).

If the Network Profile associated with a Server Scenario specifies only IPV4 addressing then that Server Scenario must operate in an IPv4 Address space (i.e any Server started by that Scenario must have an IPv4 address).

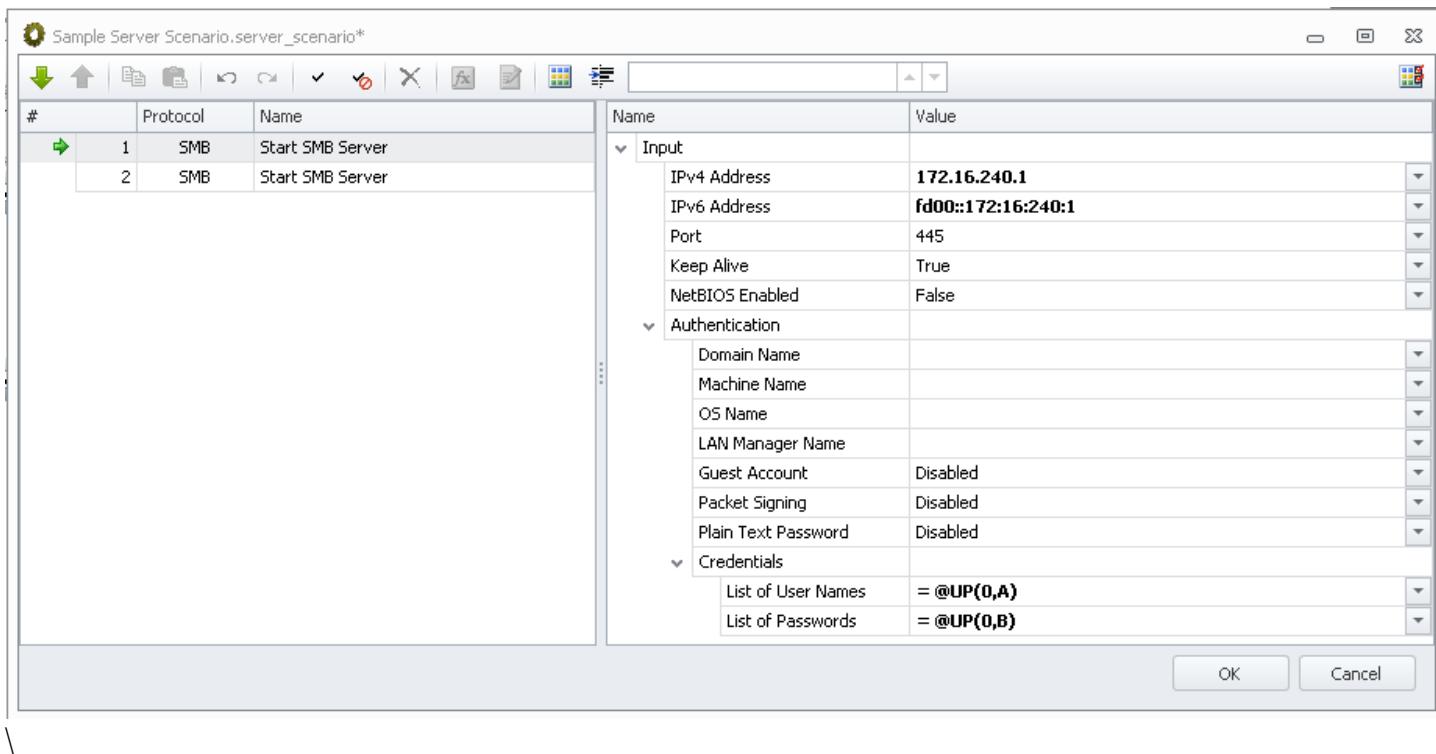
Likewise, if only IPv6 is specified for a Client Scenario, all Clients will have IPv6 addresses. If only IPv6 addresses are specified in the Network Profile for a Server Scenario, all servers started by that Scenario must be started with IPv6 addresses.

If the Network Profile for a Server Scenario specifies both IPv4 and IPv6 addresses then the Servers started in that Scenario may have either IPv4 addresses or IPv6 or both (it is possible for Load DynamiX internal servers to have both an IPv4 address and an IPv6 address). If the Network Profile for a Client Scenario specifies both IPv4 and IPv6 addresses then Client Scenarios associated with that Network Profile may access DUT's with either IPv4 or IPv6 addresses.

The initial Open Action in a Scenario will determine what kind of addressing (IPv4 or IPv6) that the client will use (i.e. if the first Open in an SMB2 Scenario is to an IPv6 address, the Client Scenarios in that test will have IPv6 addresses). It is possible to mix IPv4 addresses and IPv6 addresses in the same Scenario.

In the Server Scenario screen shot below, the SMB Server is being started with both IPv4 and IPv6 addresses which means that the Network Profile associated with this Server Scenario must have had both IPv4 and IPv6 enabled. The Network Profile above would be an appropriate network definition for

the **Start SMB Server** Action below.



IPv6 Statistics

Any Scenario that uses IPv6 addresses will generate an separate set of IPv6 statistics. the IPv6 statistics convey the same information as their IPv4 counterparts:

- TCPv6 Connection Time: IPv6 TCP Connection time in micro-seconds
- TCPv6 Connections: IPv6 TCP connection count or connections per second
- TCPv6 Details: IPv6 TCP packets/bytes transmitted or received, count or per second
- TCPv6 Throughput: IPv6 TCP throughput in packets/sec or kilobits/sec, receive and transmit

See the [Advanced Concepts: Test Execution Rule](#) for a complete list of the statistics generated when IPv6 addressing is used in a Scenario.

IPv6 CAVEATS

Like network types must always be present for Client and Server communications. An IPv4 only Client Scenario cannot open a connection to an IPv6 network, and an IPv6-only Client Scenario cannot open a connection to an IPv4 network.

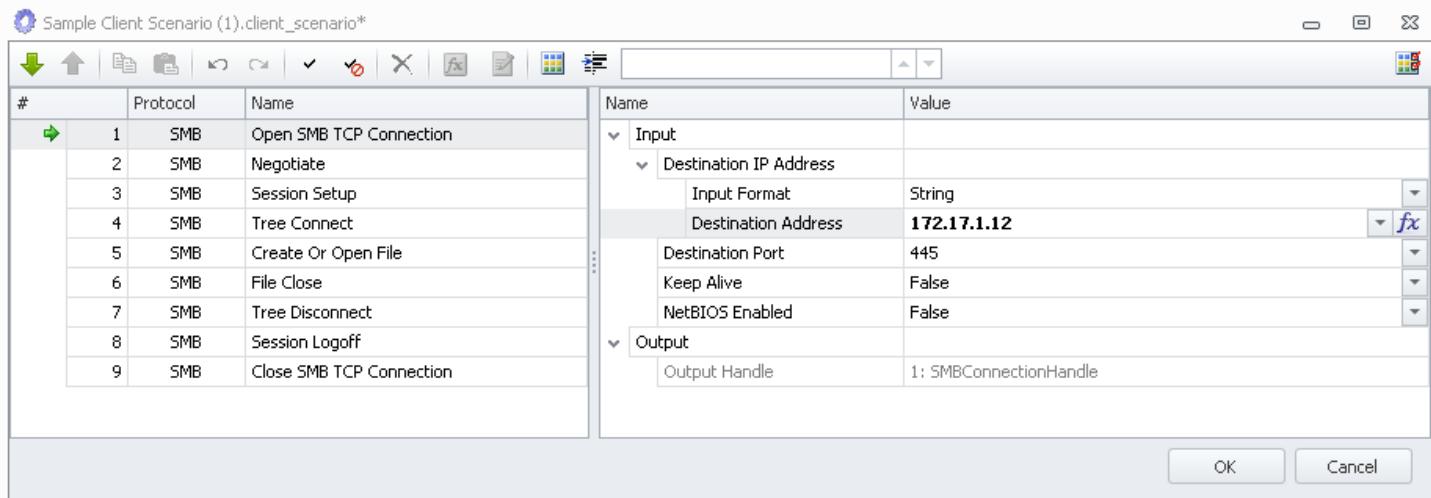
Appendix: DNS and UDP Protocols

Appendix: DNS and UDP Protocols

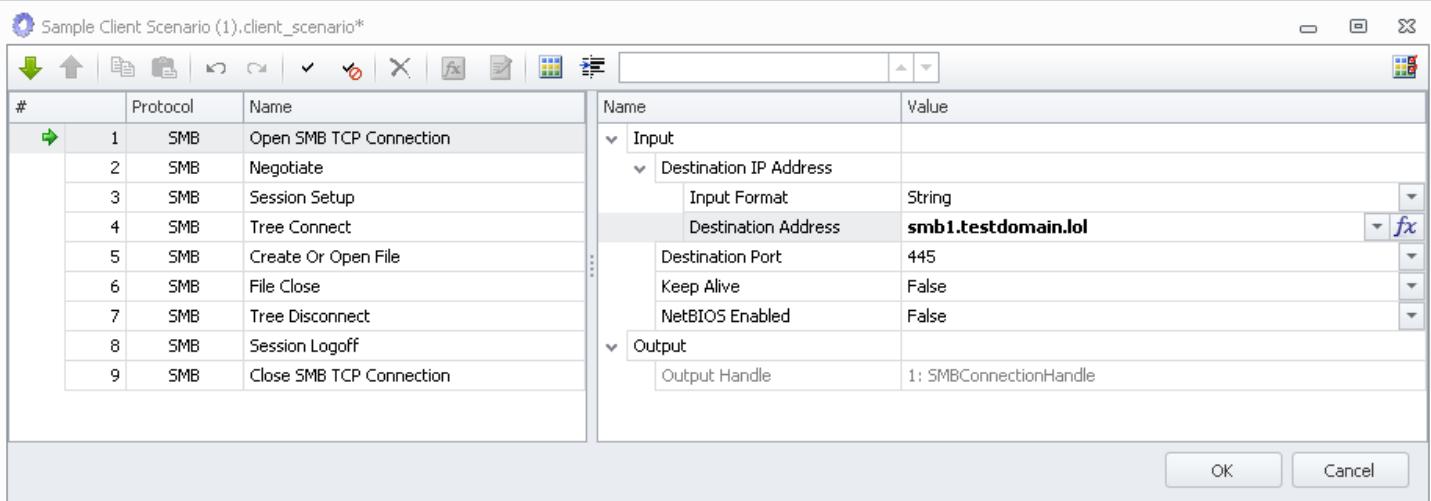
DNS

In the [Test Creation chapter](#), the DNS tab of the Network Profile is documented. DNS support allows the Tester to open connections to a Device Under Test or other device using its DNS name or using its IP Address. The Open Connection Action will take one of two forms

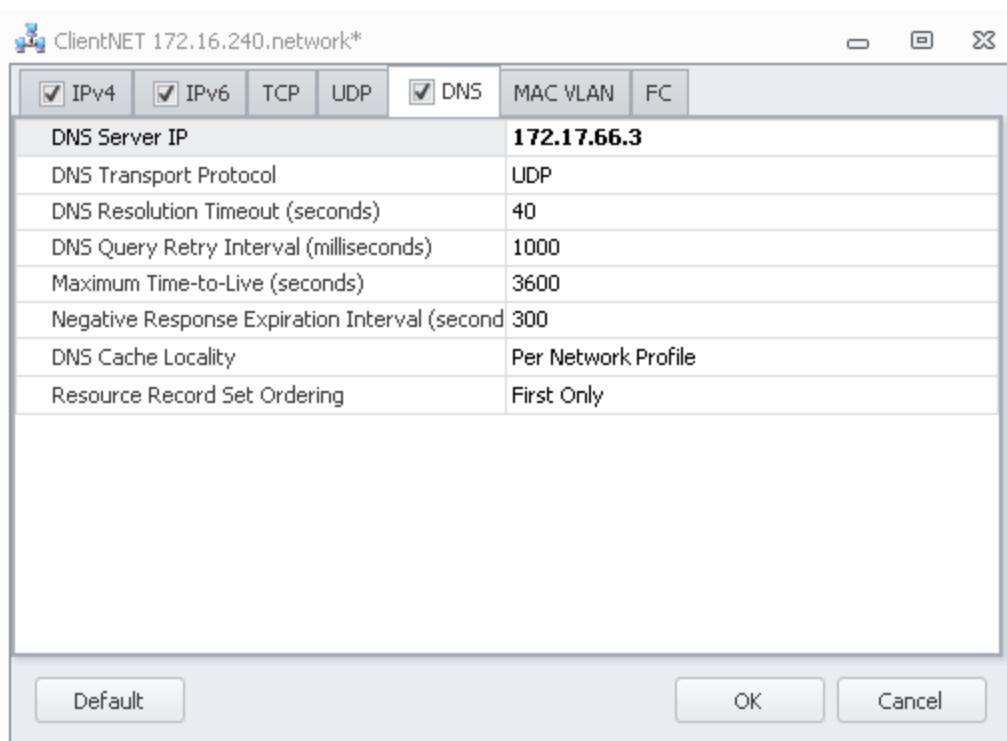
IP Address



DNS Name



To use the DNS Name in an **Open ... Connection** Action, DNS must be enabled in the Network Profile by checking the box in the DNS tab and providing the IP Address of a DNS Server that can resolve the Device Under Test's name.



DNS Parameter	Meaning	Options
DNS Server IP	IP address of the DNS server	IPv4 or IPv6 format
DNS Transport Protocol	Protocol used to send/receive DNS packets	UDP or TCP
DNS Resolution Timeout (seconds)	Timeout period in seconds for DNS requests	0 to 100 seconds
DNS Query Retry Interval (milliseconds)	The retry interval in milliseconds for DNS packets	0 to 10000 milliseconds
Maximum Time to Live (seconds)	The maximum time DNS responses are kept in the cache	0 to 360000 seconds
Negative Response Expiration Interval (seconds)	The length of time in seconds that a Negative Response is held in the Client's cache	0 to 360000 seconds
DNS Cache Locality	The locality of the DNS cache entries	Network Profile, Port, Source
Resource Record Set Ordering	How multiple responses are handled	First Only, Round Robin, Random

DNS Statistics

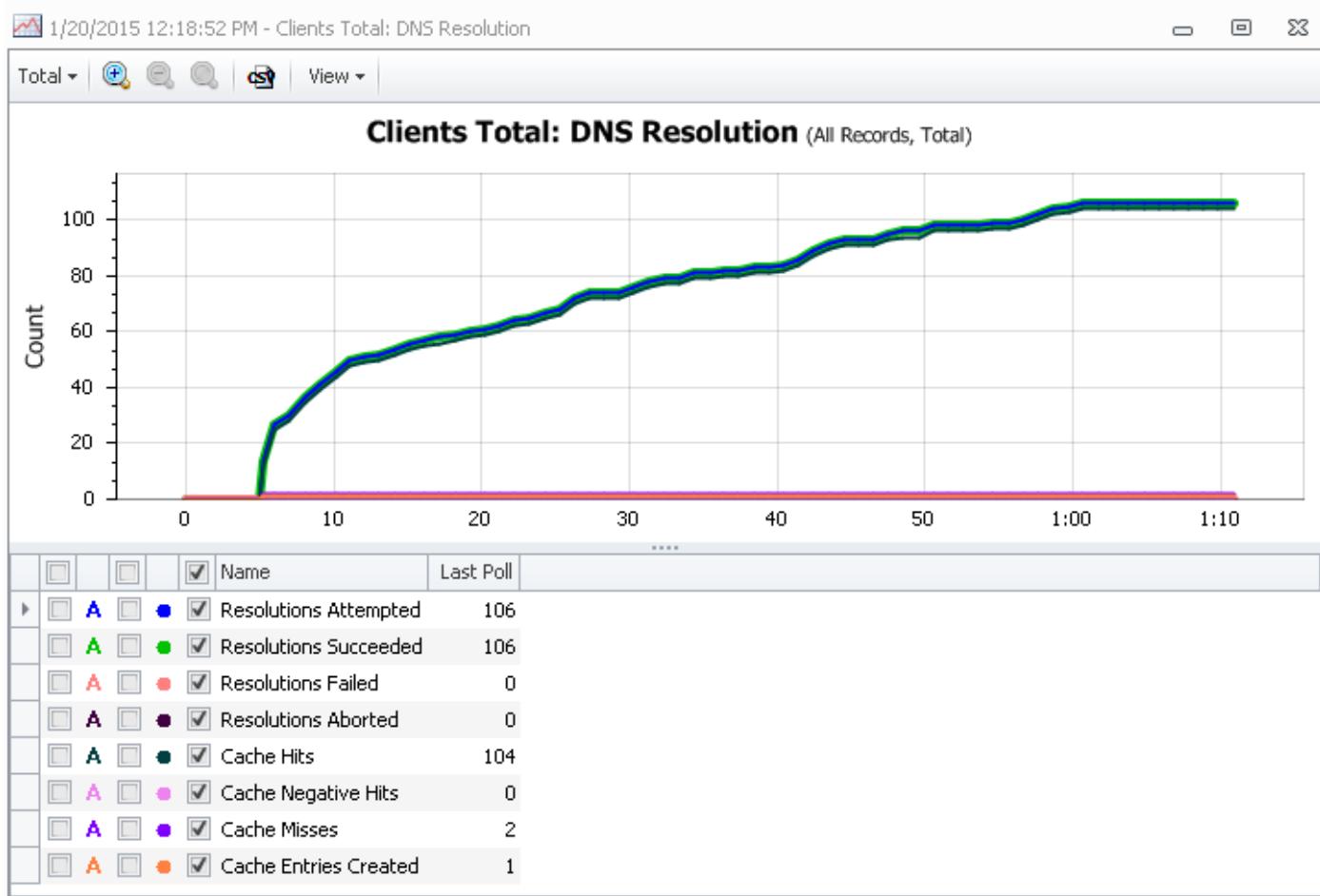
When DNS is used in the **Open ... Connection** Action of a Scenario, additional Statistics are generated in the

Report folder:

- DNS Query
- DNS Query Response Time
- DNS Resolution

DNS Resolution Time

An example DNS Resolution graph



See the [Advanced Concepts: Test Execution Rule](#) for a complete list of the statistics generated when DNS name resolution is used in a Scenario.

Client Log file:

```
=====
=====
DNS resolutions
=====
=====
Resolutions attempted:          106
Resolutions succeeded:         106
Resolutions failed:            0
Resolutions aborted:           0
Cache hits:                   104
Cache negative hits:           0
Cache misses:                 2
Cache entries created:          1
Cache updates:
  -- Record A:                  1
  -- Record AAAA:                0
Cache negative updates:
  -- Record A:                  0
  -- Record AAAA:                0
-----
```

```

Resolution Time Average: 382 (microsec)
Resolution Time Minimum: 381 (microsec)
Resolution Time Maximum: 383 (microsec)
=====
=====
=====
DNS Query Summary
Failed Aborted
=====
=====
Overall: 1 1
0 0
-----
Query A: 1 1
0 0
Query AAAA: 0 0
0 0
=====
=====
=====
DNS Query A
=====
=====
Attempted: 1
Succeeded: 1
Failed: 0
-- NXDOMAIN: 0
-- REFUSED: 0
-- Empty answer: 0
-- Truncated: 0
Aborted: 0
-- Transport Reset: 0
-- Transport Timeout: 0
Retries: 0
-----
Response Time Average: 246 (microsec)
Response Time Minimum: 246 (microsec)
Response Time Maximum: 246 (microsec)
=====
=====
=====
DNS Packets
TCP/IPv4 UDP/IPv4
TCP/IPv6 UDP/IPv6
=====
=====
Transmitted packets: 1 0
0 0
Transmitted bytes: 39 0
0 0
Received packets ok: 1 0
0 0
Received bytes ok: 89 0
0 0
Received packets dropped: 0 0
0 0
Received bytes dropped: 0 0

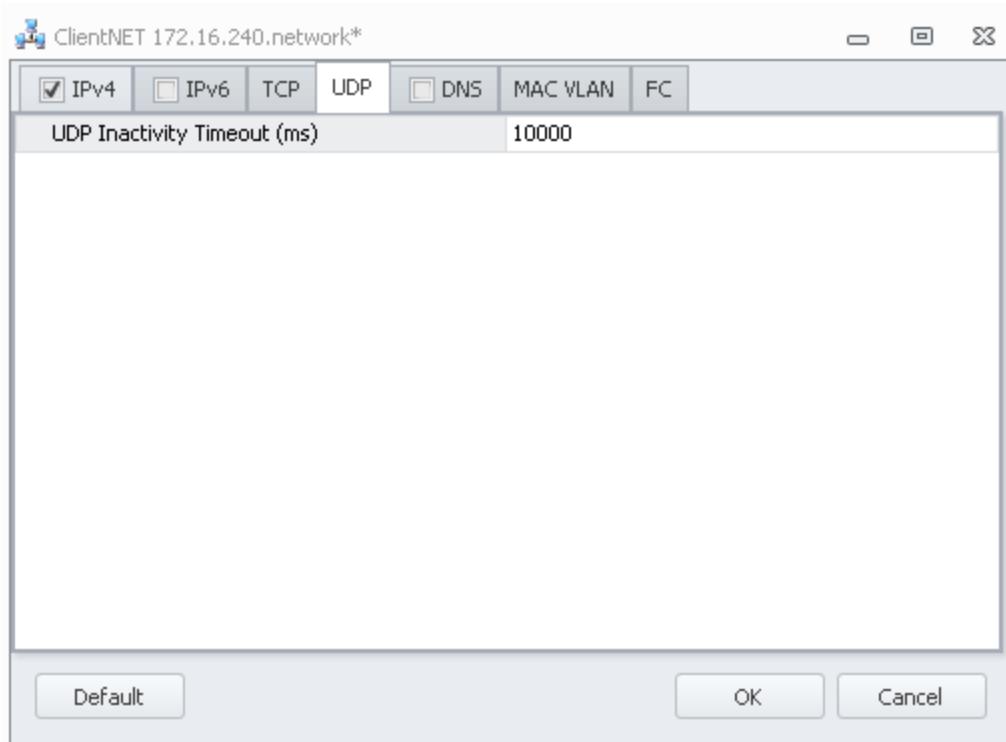
```

0	0	
=====		
=====		
=====		
TCP Connection Handshakes	Total	IPv4
IPv6		
=====		
=====		
Connections Attempted:	109	109
0		
-- DNS Resolutions Attempted:	106	n/a
n/a		
-- DNS Resolutions Succeeded:	106	n/a
n/a		
-- DNS Resolutions Failed:	0	n/a
n/a		
-- ARP/NDP Resolutions Attempted:	109	109
0		
-- ARP/NDP Resolutions Succeeded:	109	109
0		
-- ARP/NDP Resolutions Failed:	0	0
0		
-- SYN Handshakes Attempted:	109	109
0		
-- SYN Handshakes Timeout:	0	0
0		
-- SYN Handshakes Rejected:	0	0
0		
Connections Opened:	109	109
0		
Connections Failed:	0	n/a
n/a		
Rejected (invalid destination):	0	0
0		
=====		
=====		

UDP

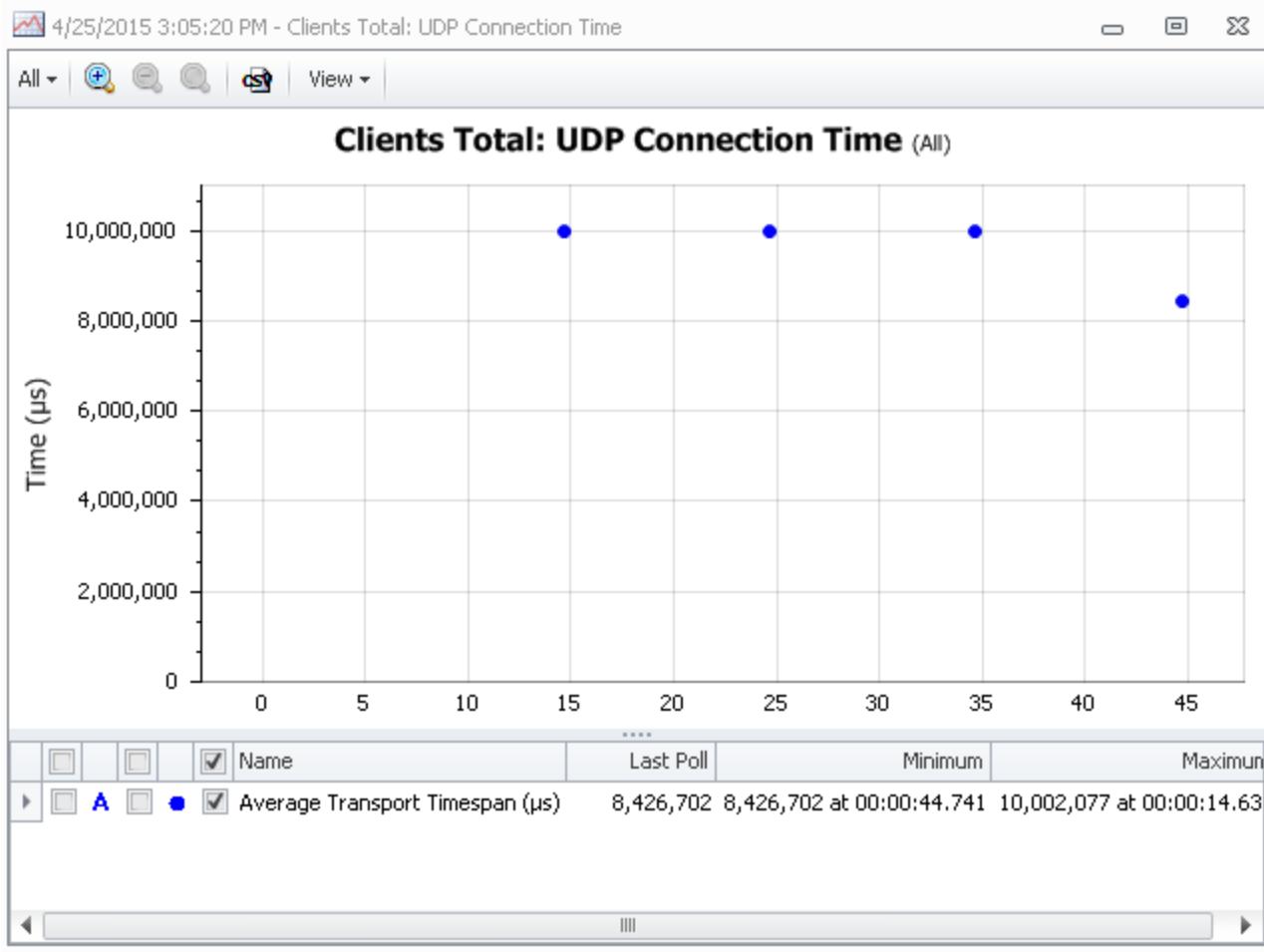
UDP Inactivity Timeout

Defined in the Network Profile UDP tab. Measured in milliseconds. Default is 1000 ms (1 second).

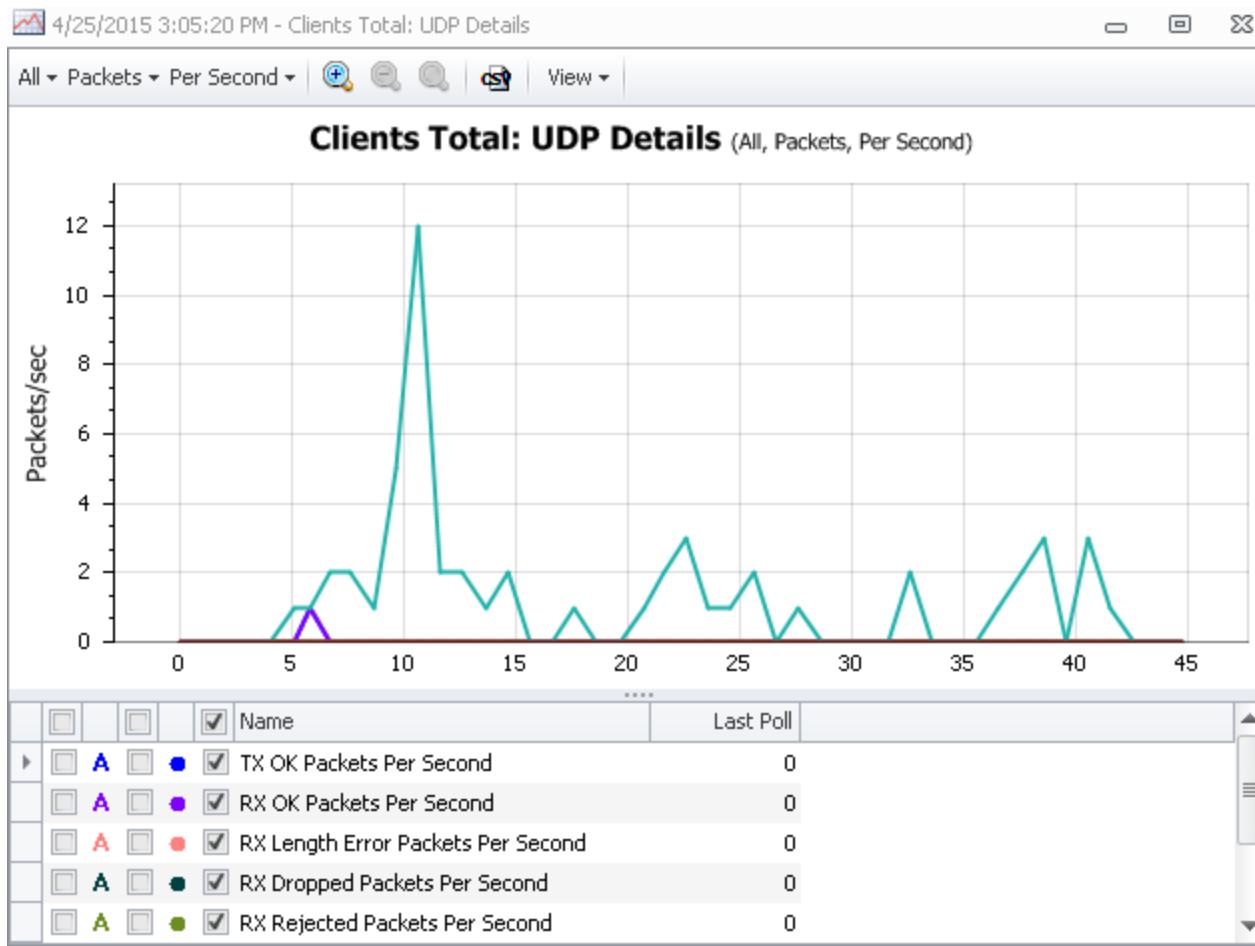


UDP Statistics Graphs

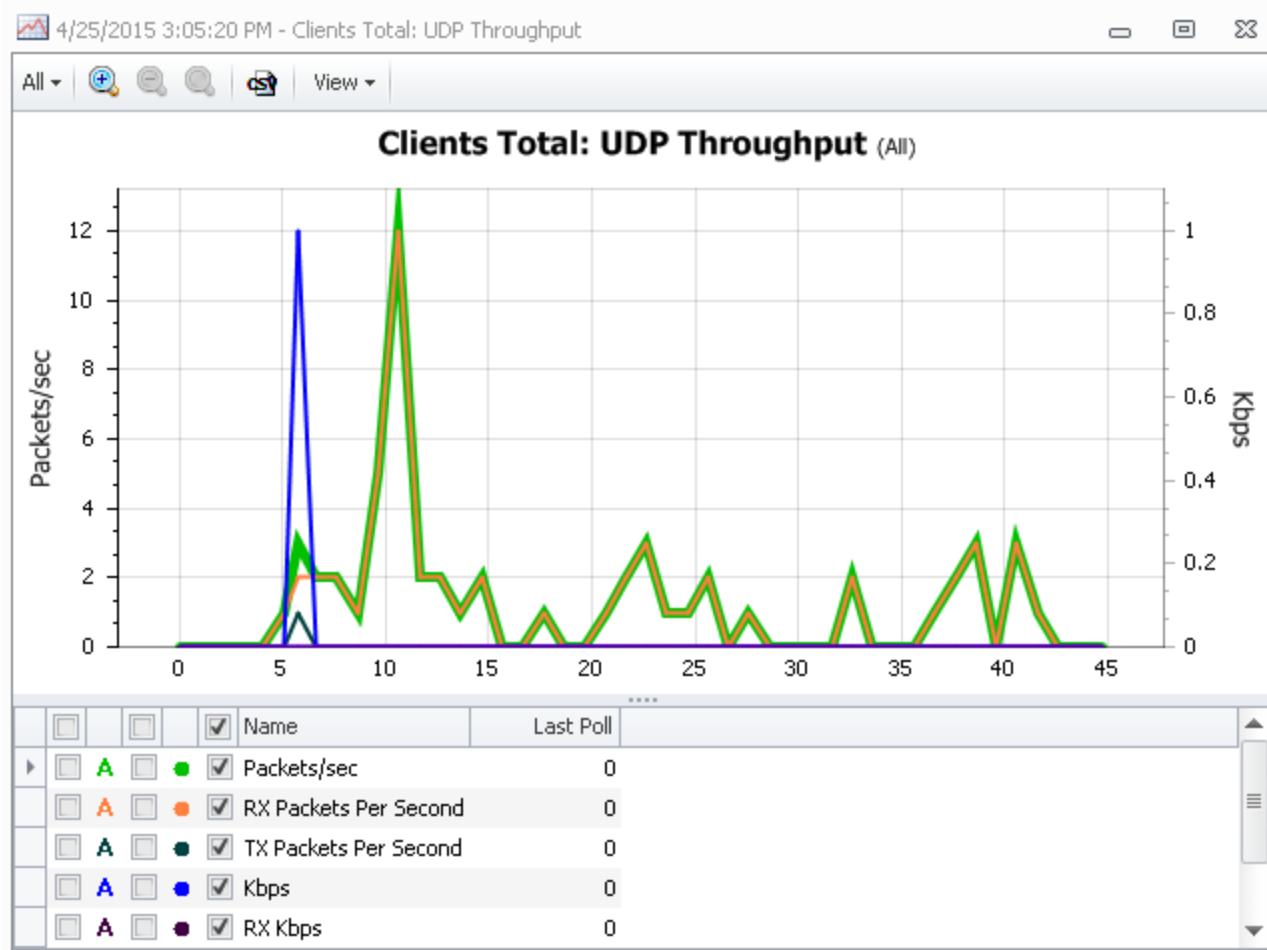
UDP Connection Time



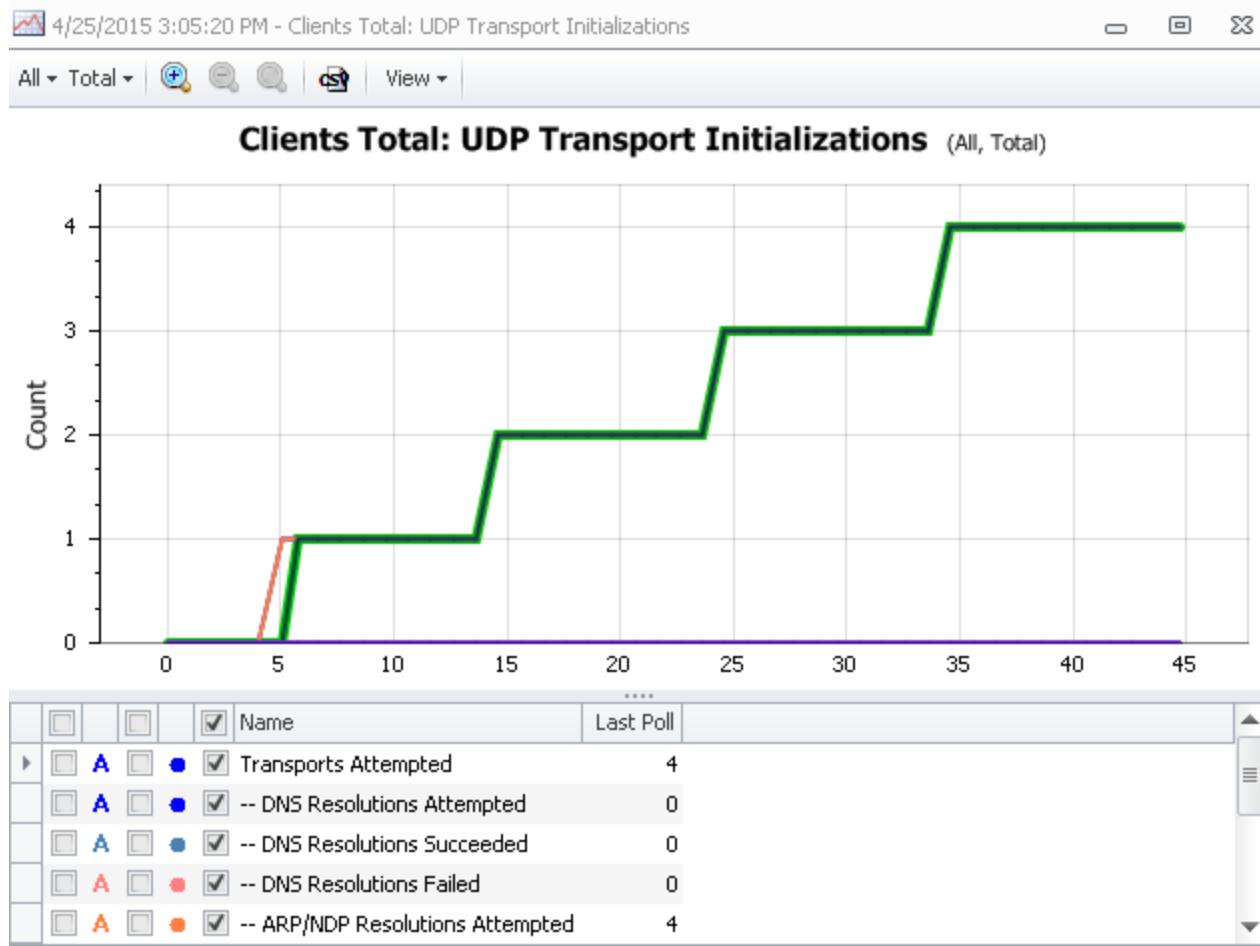
UDP Details



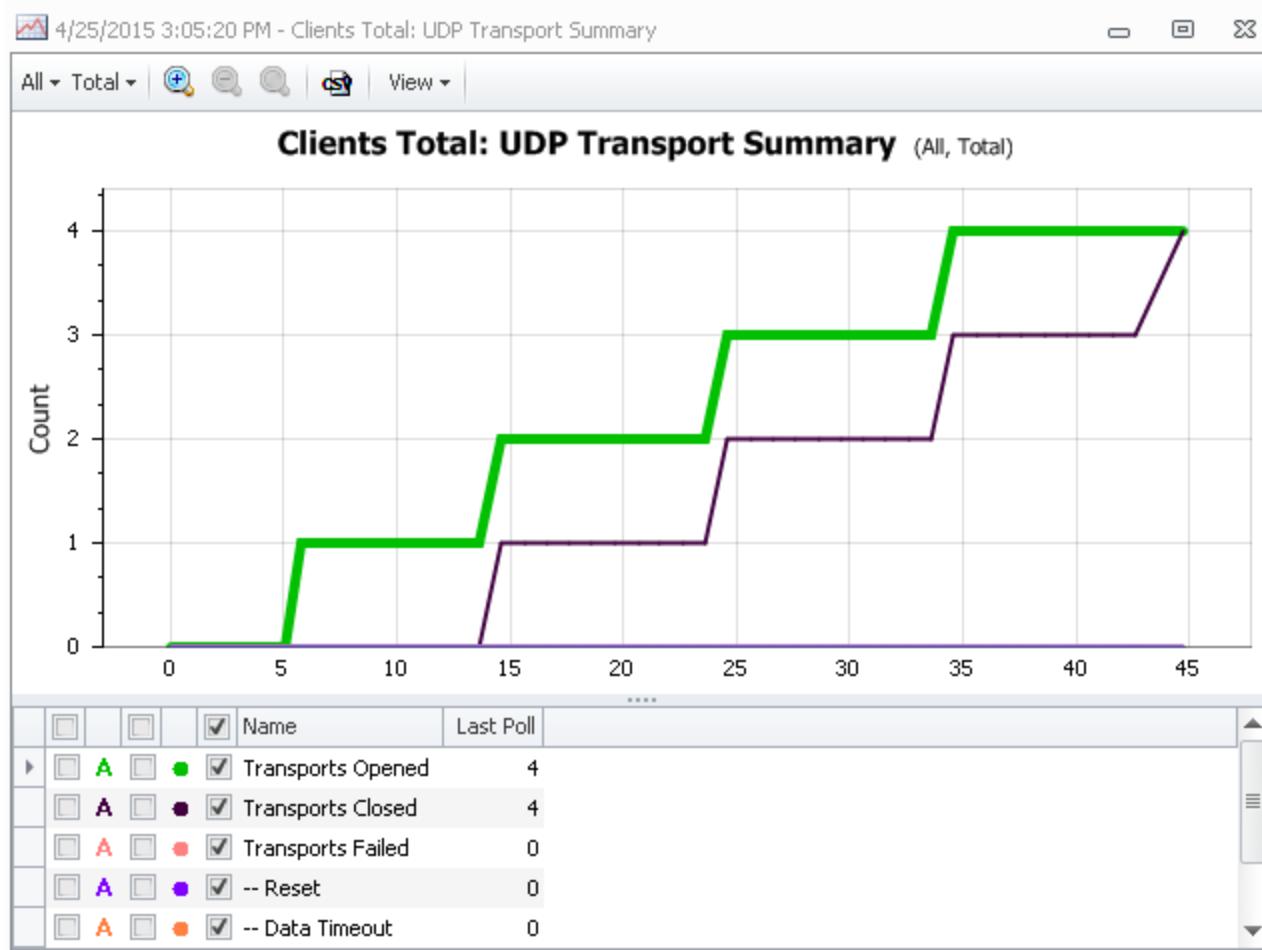
UDP Throughput



UDP Transport Initializations



UDP Transport Summary



UDP Client Log Statistics

UDP Transport Initializations	Total	IPv4	IPv6
<hr/>			
Transports Attempted:	4	4	0
-- DNS Resolutions Attempted:	0	n/a	n/a
-- DNS Resolutions Succeeded:	0	n/a	n/a
-- DNS Resolutions Failed:	0	n/a	n/a
-- ARP/NDP Resolutions Attempted:	4	4	0
-- ARP/NDP Resolutions Succeeded:	4	4	0
-- ARP/NDP Resolutions Failed:	0	0	0
-- Rejected by Peer:	0	0	0
Transports Opened:	4	4	0
Transports Failed:	0	n/a	n/a
Rejected (invalid destination):	56	36	20
<hr/>			
UDP Transport Summary	Total	IPv4	IPv6
<hr/>			
Transports Opened:	4	4	0
Transports Closed:	4	4	0
Transports Failed:	0	0	0
-- Reset:	0	0	0
-- Due to Throttle:	0	0	0
-- Data Timeout:	0	0	0
-- Inactivity Timeout:	0	0	0
Average Duration:	9608218	9608218	0 (microsec)
Minimum Duration:	8426702	8426702	0 (microsec)
Maximum Duration:	10002077	10002077	0 (microsec)
<hr/>			

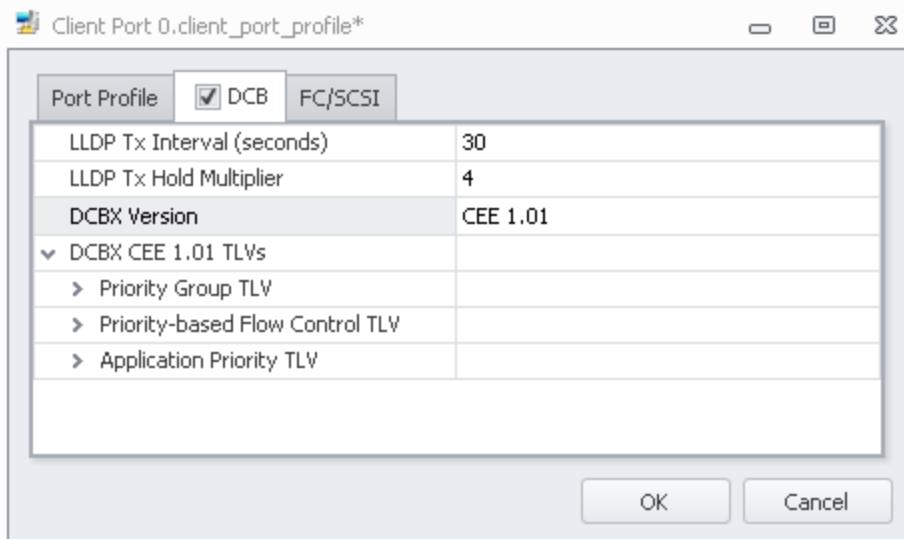
UDP Throughput Statistics	Total	IPv4	IPv6
<hr/>			
rx packets ok:	1	1	0
rx header bytes ok:	8	8	0
rx data bytes ok:	59	59	0
rx packets dropped:	0	0	0
rx header bytes dropped:	0	0	0
rx data bytes dropped:	0	0	0
rx packets throttled:	0	0	0
rx header bytes throttled:	0	0	0
rx data bytes throttled:	0	0	0
rx packets rejected:	0	0	0
rx header bytes rejected:	0	0	0
rx data bytes rejected:	0	0	0
rx packets invalid destination:	56	36	20
rx header bytes invalid destination:	448	288	160
rx data bytes invalid destination:	12221	10649	1572
rx packets header length error:	0	0	0
rx bytes header length error:	0	0	0
rx packets data length error:	0	0	0
rx bytes data length error:	0	0	0
tx packets ok:	1	1	0
tx header bytes ok:	8	8	0
tx data bytes ok:	43	43	0
tx packets canceled:	0	0	0
tx header bytes canceled:	0	0	0
tx data bytes canceled:	0	0	0
<hr/>			

Appendix: DCB/DCBx

Appendix: DCB/DCBX

Data Center Bridging and Data Center Bridging eXchange are Ethernet enhancements for Data Center environments. DCB and DCBX goals are to reduce loss due to queue overflow ("loss-less ethernet") and bandwidth allocation on links in the Data Center. See [IEEE 802.1Q work group page](#) for detailed information. The DCB controls should be left disabled unless the Project is targeted at testing or using DCB/DCBX capabilities of a DUT.

DCB/DCBX controls are accessed via the Logical Port Resource.



From this interface the Tester has access to LLDP Transmit (Tx) Interval and LLDP Tx Hold Multiplier.

The Tester can choose between two different versions of the DCBX protocol (CEE 1.01 the default) and IEEE (IEEE 802.1Qaz). The choice of DCBX version dictates the TLV choices presented.

DCBX version CEE 1.01

Client Port 0.client_port_profile*

Port Profile DCB FC/SCSI

LLDP Tx Interval (seconds)	30
LLDP Tx Hold Multiplier	4
DCBX Version	CEE 1.01
▼ DCBX CEE 1.01 TLVs	
▼ Priority Group TLV	
Enabled	True
Willing	True
Priority 0 PGID	1
Priority 1 PGID	1
Priority 2 PGID	1
Priority 3 PGID	0
Priority 4 PGID	2
Priority 5 PGID	0
Priority 6 PGID	0
Priority 7 PGID	0
PGID 0 BW %	50
PGID 1 BW %	0
PGID 2 BW %	50
PGID 3 BW %	0
PGID 4 BW %	0
PGID 5 BW %	0
PGID 6 BW %	0
PGID 7 BW %	0
> Priority-based Flow Control TLV	
> Application Priority TLV	

OK Cancel

Client Port 0.client_port_profile*

Port Profile	<input checked="" type="checkbox"/> DCB	FC/SCSI
LLDP Tx Interval (seconds)	30	
LLDP Tx Hold Multiplier	4	
DCBX Version	CEE 1.01	
▼ DCBX CEE 1.01 TLVs		
> Priority Group TLV		
▼ Priority-based Flow Control TLV		
Enabled	True	
Willing	True	
Priority 0	True	
Priority 1	True	
Priority 2	True	
Priority 3	True	
Priority 4	True	
Priority 5	True	
Priority 6	True	
Priority 7	True	
▼ Application Priority TLV		
Enabled	True	
Willing	True	
Priority 0	None	
Priority 1	None	
Priority 2	None	
Priority 3	None	
Priority 4	iSCSI	
Priority 5	None	
Priority 6	None	
Priority 7	None	

OK Cancel

DCBX version IEEE 802.1Qaz

Port Profile	<input checked="" type="checkbox"/> DCB	FC/SCSI
LLDP Tx Interval (seconds)	30	
LLDP Tx Hold Multiplier	4	
DCBX Version	IEEE 802.1Qaz TLVs	
▼ DCBX IEEE 802.1Qaz TLVs		
> ETS TLV		
> Priority-based Flow Control TLV		
> Application Priority TLV		

OK Cancel

Client Port 0.client_port_profile*

Port Profile	<input checked="" type="checkbox"/> DCB	FC/SCSI
LLDP Tx Interval (seconds)	30	
LLDP Tx Hold Multiplier	4	
DCBX Version	IEEE 802.1Qaz TLVs	
DCBX IEEE 802.1Qaz TLVs		
ETS TLV		
Enabled	True	
Willing	True	
Priority 0 TC	1	
Priority 1 TC	1	
Priority 2 TC	1	
Priority 3 TC	0	
Priority 4 TC	2	
Priority 5 TC	0	
Priority 6 TC	0	
Priority 7 TC	0	
TC 0 BW %	50	
TC 1 BW %	0	
TC 2 BW %	50	
TC 3 BW %	0	
TC 4 BW %	0	
TC 5 BW %	0	
TC 6 BW %	0	
TC 7 BW %	0	
> Priority-based Flow Control TLV		
> Application Priority TLV		

OK **Cancel**

Client Port 0.client_port_profile*

Port Profile	<input checked="" type="checkbox"/> DCB	FC/SCSI
LLDP Tx Hold Multiplier	4	
DCBX Version	IEEE 802.1Qaz TLVs	
DCBX IEEE 802.1Qaz TLVs		
ETS TLV		
Priority-based Flow Control TLV		
Enabled	True	
Willing	True	
Priority 0	True	
Priority 1	True	
Priority 2	True	
Priority 3	True	
Priority 4	True	
Priority 5	True	
Priority 6	True	
Priority 7	True	
Application Priority TLV		
Enabled	True	
Willing	True	
Priority 0	None	
Priority 1	None	
Priority 2	None	
Priority 3	None	
Priority 4	iSCSI	
Priority 5	None	
Priority 6	None	
Priority 7	None	

OK **Cancel**

DCB Statistics

See the [Advanced Concepts: Test Execution Rule](#) for a complete list of the statistics generated when DCB is enabled on a Logical Port.

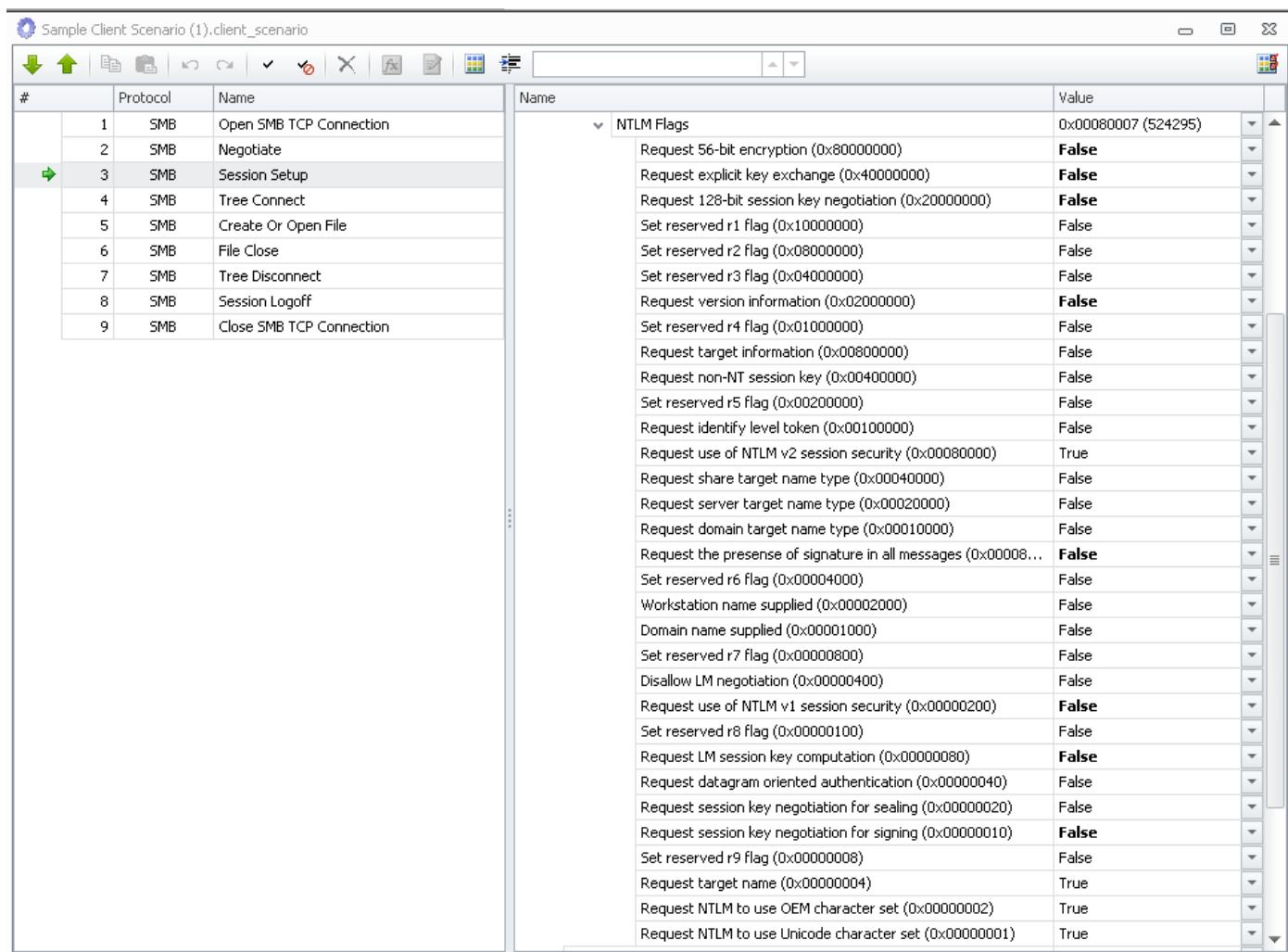
Appendix: NTLM Flags

Appendix: NTLM Flags

Protocols that support NTLM authentication provide the mechanisms to set/unset each of the 32 NTLM flags. The Load DynamiX protocols that support NTLM are:

- CIFS-SMB
- SMB 2/2.1
- SMB 3
- HTTP/HTTPS

In CIFS-SMB/SMB2/SMB3 protocols, the **Session Setup** Action provides access to the NTLM flags.

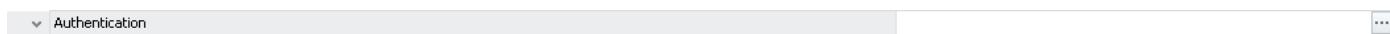


The screenshot shows the Load DynamiX software interface. On the left, a tree view displays a scenario named "Sample Client Scenario (1).client_scenario". Under this scenario, the "Session Setup" action is expanded, revealing its internal steps: Open SMB TCP Connection, Negotiate, Session Setup (which is selected), Tree Connect, Create Or Open File, File Close, Tree Disconnect, Session Logoff, and Close SMB TCP Connection. To the right of the tree view is a detailed configuration table for the "Session Setup" action. The table has two columns: "Name" and "Value". The "Name" column lists various NTLM flag names, and the "Value" column shows their current state as either "True" or "False". A dropdown menu next to the "Name" column is currently open, showing the option "NTLM Flags".

#	Protocol	Name	
1	SMB	Open SMB TCP Connection	
2	SMB	Negotiate	
3	SMB	Session Setup	
4	SMB	Tree Connect	
5	SMB	Create Or Open File	
6	SMB	File Close	
7	SMB	Tree Disconnect	
8	SMB	Session Logoff	
9	SMB	Close SMB TCP Connection	

Name	Value
Request 56-bit encryption (0x80000000)	False
Request explicit key exchange (0x40000000)	False
Request 128-bit session key negotiation (0x20000000)	False
Set reserved r1 flag (0x10000000)	False
Set reserved r2 flag (0x08000000)	False
Set reserved r3 flag (0x04000000)	False
Request version information (0x02000000)	False
Set reserved r4 flag (0x01000000)	False
Request target information (0x00800000)	False
Request non-NT session key (0x00400000)	False
Set reserved r5 flag (0x00200000)	False
Request identify level token (0x00100000)	False
Request use of NTLM v2 session security (0x00080000)	True
Request share target name type (0x00040000)	False
Request server target name type (0x00020000)	False
Request domain target name type (0x00010000)	False
Request the presence of signature in all messages (0x00008...)	False
Set reserved r6 flag (0x00004000)	False
Workstation name supplied (0x00002000)	False
Domain name supplied (0x00001000)	False
Set reserved r7 flag (0x00000800)	False
Disallow LM negotiation (0x00000400)	False
Request use of NTLM v1 session security (0x00000200)	False
Set reserved r8 flag (0x00000100)	False
Request LM session key computation (0x00000080)	False
Request datagram oriented authentication (0x00000040)	False
Request session key negotiation for sealing (0x00000020)	False
Request session key negotiation for signing (0x00000010)	False
Set reserved r9 flag (0x00000008)	False
Request target name (0x00000004)	True
Request NTLM to use OEM character set (0x00000002)	True
Request NTLM to use Unicode character set (0x00000001)	True

In the HTTP/HTTPS protocols, the NTLM flags are settable within each HTTP/HTTPS Action that supports authentication. The NTLM flags can be set within the Authentication dialog shown below that is opened when the  button on the Authentication field is clicked



HTTP Authentication Parameters*



Preemptive Authorization:		Passive Authorization:																															
<input checked="" type="radio"/> None <input type="radio"/> Basic <input type="radio"/> Digest <input type="radio"/> NTLM: NTLM <input type="radio"/> Negotiate: NTLM <input type="radio"/> Negotiate: NTLM+Kerberos <input type="radio"/> Negotiate: Kerberos+NTLM <input type="radio"/> Negotiate: Kerberos <input type="radio"/> Amazon S3		Enabled: Basic Digest NTLM: NTLM <input type="button" value="Enable"/> <input type="button" value="Disable"/> <input type="button" value="Priority Up"/> <input type="button" value="Priority Down"/>																															
		Disabled: Negotiate: NTLM Negotiate: Kerberos																															
Credentials:		NTLM Information:																															
Username: <input type="text"/> Password: <input type="password"/>		Domain Name: WORKGROUP Machine Name: <input type="text"/>																															
Kerberos Ticket Handle:		NTLM Flags: <table border="1"> <thead> <tr> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>NTLM Flags</td> <td>0x000080...</td> </tr> <tr> <td>Request 56-bit encryption (0x00000001)</td> <td>False</td> </tr> <tr> <td>Request explicit key exchange (0x00000002)</td> <td>False</td> </tr> <tr> <td>Request 128-bit session key (0x00000004)</td> <td>False</td> </tr> <tr> <td>Set reserved r1 flag (0x10000000)</td> <td>False</td> </tr> <tr> <td>Set reserved r2 flag (0x08000000)</td> <td>False</td> </tr> <tr> <td>Set reserved r3 flag (0x04000000)</td> <td>False</td> </tr> <tr> <td>Request version information (0x02000000)</td> <td>False</td> </tr> <tr> <td>Set reserved r4 flag (0x01000000)</td> <td>False</td> </tr> <tr> <td>Request target information (0x00800000)</td> <td>False</td> </tr> <tr> <td>Request non-NT session key (0x00400000)</td> <td>False</td> </tr> <tr> <td>Set reserved r5 flag (0x00200000)</td> <td>False</td> </tr> <tr> <td>Request identify level token (0x00100000)</td> <td>False</td> </tr> <tr> <td>Request use of NTLM v2 session security (0x00080000)</td> <td>True</td> </tr> </tbody> </table>		Name	Value	NTLM Flags	0x000080...	Request 56-bit encryption (0x00000001)	False	Request explicit key exchange (0x00000002)	False	Request 128-bit session key (0x00000004)	False	Set reserved r1 flag (0x10000000)	False	Set reserved r2 flag (0x08000000)	False	Set reserved r3 flag (0x04000000)	False	Request version information (0x02000000)	False	Set reserved r4 flag (0x01000000)	False	Request target information (0x00800000)	False	Request non-NT session key (0x00400000)	False	Set reserved r5 flag (0x00200000)	False	Request identify level token (0x00100000)	False	Request use of NTLM v2 session security (0x00080000)	True
Name	Value																																
NTLM Flags	0x000080...																																
Request 56-bit encryption (0x00000001)	False																																
Request explicit key exchange (0x00000002)	False																																
Request 128-bit session key (0x00000004)	False																																
Set reserved r1 flag (0x10000000)	False																																
Set reserved r2 flag (0x08000000)	False																																
Set reserved r3 flag (0x04000000)	False																																
Request version information (0x02000000)	False																																
Set reserved r4 flag (0x01000000)	False																																
Request target information (0x00800000)	False																																
Request non-NT session key (0x00400000)	False																																
Set reserved r5 flag (0x00200000)	False																																
Request identify level token (0x00100000)	False																																
Request use of NTLM v2 session security (0x00080000)	True																																

The full set of the NTLM flags are

FLAG	DEFAULT
Request 56 bit encryption	FALSE
Request explicit key exchange	FALSE
Request 128 bit session key negotiation	FALSE
Set reserved r1 flag	FALSE
Set reserved r2 flag	FALSE
Set reserved r3 flag	FALSE
Request version information	FALSE
Set reserved r4 flag	FALSE
Request target information	FALSE
Request non-NT session key	FALSE
Set reserved r5 flag	FALSE
Request identify level token	FALSE
Request NTLM v2 session security	TRUE
Request share target name type	FALSE
Request server target name type	FALSE

Request domain target name type	FALSE
Request presence of signature in all messages	FALSE
Set reserved r6 flag	FALSE
Workstation name supplied	FALSE
Domain name supplied	FALSE
Set reserved r7 flag	FALSE
Disallow LM negotiation	FALSE
Request NTLMv1 session security	FALSE
Set reserved r8 flag	FALSE
Request LM session key computation	FALSE
Request datagram oriented authentication	FALSE
Request session key negotiation for sealing	FALSE
Request session key negotiation for signing	FALSE
Set reserved r9 flag	FALSE
Request target name	TRUE
Request NTLM use OEM character set	TRUE
Request NTLM use Unicode character set	TRUE

Copyright © 2008-2017 Virtual Instruments Inc.

Appendix: Change Notify and Change Notify Cancel Actions

Appendix: SMB2/3 Change Notify and Change Notify Cancel Actions

Change Notify and Change Notify Cancel Actions unique behaviors

The **Change Notify** Action and the corresponding **Change Notify Cancel** Action are among the most unique Actions in the SMB2/3 Toolbox.

The **Change Notify** Action requests that changes to Directories on SMB2/3 servers be reported back to a Scenario and generates one or two responses from the server. The typical first response from the server indicates acceptance of the request or that the request failed for some reason. The typical second response (assuming that the first request was acceptance), received at some unspecified time later, indicates success (the Directory was changed) or that the request has been terminated by the server or canceled. When a **Change Notify** Action is pending, the Scenario must be prepared to receive the "CHANGE_NOTIFY" response whenever the server detects a change.

An SMB2/3 Scenario issuing a **Change Notify Cancel** Action may not receive a specific response to that Action from an SMB2/3 server. An SMB2/3 Scenario is expected to receive one response from a server for all other SMB2/3 commands (except **Change Notify** which might produce two responses). If the **Change Notify Cancel** Action is successful, the server will send a "CHANGE_NOTIFY" response containing "STATUS_CANCELLED" in response to a pending "CHANGE_NOTIFY" request. The **Change Notify** Action will be marked as Failed.

The **Change Notify Cancel** Action is the only Action in the SMB2/3 Toolbox that allows a Tester to change the contents of an SMB2/3 command SMB2 Header field. See **Change Notify Cancel** Action Input Parameters below for details.

Change Notify

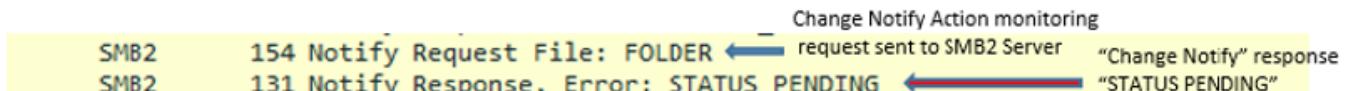
This Action requests an SMB2/SMB3 server to notify a Load DynamiX Scenario when the properties and/or contents of a Directory change. The Operations Flags and Completion Filter Properties (see below and [Appendix: Change Notify and Change Notify Cancel Actions](#) for details) control which Directory properties and content to monitor. The "CHANGE_NOTIFY" command is among the most unique commands in the SMB2 command set, in that a single "CHANGE_NOTIFY" request may receive one or two responses from an SMB2/3 server. The first response is either an error, indicating some kind of problem with the "CHANGE_NOTIFY" command that was sent, or a "STATUS_PENDING" response indicating acceptance of the command by the server and the beginning of the requested monitoring. The second response is a "CHANGE_NOTIFY" response containing either success or some error indication. See below and [Appendix: Change Notify and Change Notify Cancel Actions](#) for details.

To assist the Tester in creating Scenarios that mimic the use of SMB2/3 commands in the way that MS Windows uses them, Load DynamiX has introduced a special input parameter "Asynchronous" to the **Change Notify** Action, with two options (modes): "False" (Synchronous), and "True" (Asynchronous). This parameter controls the internal behavior of the Load DynamiX SMB2/3 Scenario when it executes a **Change Notify** Action (note: this "Asynchronous" parameter should not be confused with the SYNC / ASYNC variants of the SMB2 Packet Header, referenced in the "CHANGE NOTIFY" specification.) A link to the SMB2/3 protocol reference

material is provided in the [References and Terminology section](#).

In either mode, when the Load DynamiX Scenario sends the "CHANGE_NOTIFY" request to a server, the Scenario expects a "CHANGE_NOTIFY" response returned from that server containing a "STATUS_PENDING" error code.

The PCAP segment below demonstrates the expected return of a "CHANGE_NOTIFY" response containing a "STATUS_PENDING" error code.



If the "CHANGE_NOTIFY" response, containing some status indicator, is not received, for whatever reason - TCP error or server error, timeout, etc., the **Change Notify** Action will be marked as Aborted.

Note: If an event occurs on a server with an active "CHANGE_NOTIFY" request that is applicable to that request, within a very short time following the receipt of that request, the server may skip the "STATUS_PENDING" response altogether and respond immediately with an status code indicating the nature of the event. The indication could be "STATUS_SUCCESS", "STATUS_CANCELLED", etc., depending on the nature of the event.

Synchronous mode ("False") - default

In Synchronous mode, after receiving a "CHANGE_NOTIFY" response containing "STATUS_PENDING", the Scenario will wait until it receives a final "CHANGE_NOTIFY" response from the SMB2/3 server, containing either:

1. "STATUS_SUCCESS", indicating that a monitored change has occurred on the watched Directory
2. or, some error response ("STATUS_NOTIFY_CLEANUP", "STATUS_CANCELLED", etc.).

If the "CHANGE_NOTIFY" response is not received within the TCP Inactivity timeout window (when "Keep-Alive" is disabled), the Action and Scenario will abort.

Asynchronous mode ("True")

In Asynchronous mode, after receiving a "CHANGE_NOTIFY" response containing "STATUS_PENDING", the Scenario will continue executing, expecting a final "CHANGE_NOTIFY" response from the SMB2/3 server, containing either:

1. "STATUS_SUCCESS", indicating that a monitored change has occurred on the watched Directory
2. or, some error response ("STATUS_NOTIFY_CLEANUP", "STATUS_CANCELLED", etc.).

If the "CHANGE_NOTIFY" response is not received before the Scenario completes executing, the **Change Notify** Action will be marked as Failed.

Why use Asynchronous mode vs Synchronous mode?

In **Change Notify** Synchronous mode, the **Change Notify** Action causes the Scenario to pause until the server finally responds (other than "STATUS_PENDING") to the Action (see **Change Notify** Action Results below). Synchronous mode could be used to create an SMB2/3 Scenario

that will execute a specific set of Actions once the server responds to the **Change Notify** Action. Two real world examples:

1. A database sync process where any change to a database requires that other database-related files or storage are synchronized immediately.
2. A Windows Explorer window, displaying the content of a server Directory, reflects any change made to that Directory.

Synchronous mode allows a Scenario to more correctly emulate the behavior of the "CHANGE_NOTIFY" request as used by a Windows SMB2/3 Client. Synchronous mode allows a Tester to create a single Scenario, using SMB2 Actions (**Change Notify**, **Change Notify Cancel** and Directory Read/Write) and Scenario Control Actions (Events, Threads, If/Else If/Else/End If, etc.) to generate a variety of "CHANGE_NOTIFY" status code responses and execute specific Action sequences based on those codes.

Change Notify Asynchronous mode could be used to exercise the SMB2/3 "CHANGE_NOTIFY" server functionality itself (error handling, change detection, etc.). Asynchronous mode may be simpler to use in a Scenario but cannot be used to guarantee the Actions that will be executed when a event occurs that causes a "CHANGE_NOTIFY" response to be sent.

Change Notify Action Input Parameters

Asynchronous: See the detailed explanation above.

File Handle: The output handle of the **Create Directory** Action used to create or open the Directory that is to be monitored.

Credits Charged: See the SMB2 Credits discussion below.

Credits Requested: See the SMB2 Credits discussion below.

Operation Flags:

Watch Tree (True or False): Watch Tree == True will also monitor changes in any file or sub-directory contained in the Directory referenced by the Directory specified by File Handle (above). Watch Tree == False (default) will only monitor the Directory itself for changes as defined in the Completion Filter input.

Completion Filter:

Specifies which types of changes the Client wishes to receive "CHANGE_NOTIFY" responses for, such as Change of Directory Name, Change of Last Write, Change of Last Access, Change of Security, etc. As long as the Scenario is executing and the Change Notify is pending in the Server, the Scenario will receive "CHANGE_NOTIFY" responses for the properties set to True in the Completion Filter input. The complete list of Completion Filter properties is:

Completion Filter		0x00000009 (9)
Change of File Name	True	▼
Change of Directory Name	False	▼
Change of Attributes	False	▼
Change of File Size	True	▼
Change of Last Write	False	▼
Change of Last Access	False	▼
Change of Creation	False	▼
Change of EA	False	▼
Change of Security	False	▼
Stream Name Added	False	▼
Stream Size Changed	False	▼
Stream is Modified	False	▼

Change Notify Action Results

The **Change Notify** Action inputs determine the Directory and the Directory content/properties that are being monitored. When an SMB2/3 server responds to a **Change Notify** Action, it responds with a "CHANGE_NOTIFY" response. Included in that response is an error code:

- "STATUS_PENDING"- this server response indicates that the "CHANGE_NOTIFY" request has been accepted and Directory monitoring has begun,
- "STATUS_NOTIFY_CLEANUP" - if the server terminates the "CHANGE_NOTIFY" request before a monitored change has occurred, or
- "STATUS_CANCELLED"- if a **Change Notify Cancel** Action is sent to the SMB2/3 server before a change occurred, or
- some other error code indicating that there was a problem with the "CHANGE_NOTIFY" request itself, examples:
 - STATUS_INVALID_PARAMETER - some input to the **Change Notify** Action is incorrect
 - STATUS_ACCESS_DENIED - access to the Directory pointed to by File Handle is denied
 - STATUS_FILE_CLOSED - the Directory pointed to by the File Handle input is closed
 - STATUS_USER_SESSION_DELETED - the Scenario's user is no longer logged in

or

- "STATUS_SUCCESS" - the monitored change has occurred, information embedded in the "CHANGE_NOTIFY" response indicates the kind of change that occurred.

The following NetMon window shows a "CHANGE_NOTIFY" response containing a "STATUS_SUCCESS" return and the FileInfo data structure defining the change that was made ("the file was added to the directory").

Frame Summary

Find ▾

Frame Number	Protocol Name	Description
205	SMB2	SMB2:R CHANGE NOTIFY (0xf)

Frame Details

```

Frame: Number = 205, Captured Frame Length = 162, MediaType = ETHERNET
Ethernet: Etype = Internet IP (IPv4), DestinationAddress:[A0-36-9F-00-7F-09], SourceAddress:[00-50-56-A2-64-82]
Ipv4: Src = 172.16.2.253, Dest = 172.16.240.1, Next Protocol = TCP, Packet ID = 4637, Total IP Length = 148
Tcp: Flags=...AP..., SrcPort=Microsoft-DS(445), DstPort=Stock IXChange(527), PayloadLen=108, Seq=3085214048 - 
SMBOverTCP: Length = 104
SMB2: R CHANGE NOTIFY (0xf)
  SMBIdentifier: SMB
  SMB2Header: R CHANGE NOTIFY (0xf), TID=0x0000, MID=0x0005, PID=0x0000, SID=0x900000019
    StructureSize: 64 (0x40)
    CreditCharge: 0 (0x0)
    Status: 0x0, Facility = FACILITY_SYSTEM, Severity = STATUS_SEVERITY_SUCCESS, Code = (0) STATUS_SUCCESS
    Command: CHANGE NOTIFY (0xf)
    Credits: 0 (0x0)
    Flags: 0x3
    NextCommand: 0 (0x0)
    MessageId: 5 (0x5)
    AsyncId: 1 (0x1)
    SessionId: 17594601963545 (0x100090000019)
    Signature: Binary Large Object (16 Bytes)
  RChangeNotify:
    StructureSize: 9 (0x9)
    OutputBufferOffset: 72 (0x48)
    OutputBufferLength: 32 (0x20)
  FileInfoArray:
    FileInfo:
      NextEntryOffset: 0 (0x0)
      Action: (1) ADDED: The file was added to the directory.
      FileNameLength: 20 (0x14)
      FileName: FILE0.TEST

```

SMB2 Header indicating STATUS_SUCCESS

File Information Array returned in a Successful Change Notify Response

The **Change Notify** Action will be counted as successful if "STATUS_SUCCESS" is received. The **Change Notify** Action will be counted as failed if "STATUS_NOTIFY_CLEANUP" or "STATUS_CANCELLED" or any of the other error codes indicating that there was a problem with the "CHANGE_NOTIFY" request are received. "STATUS_PENDING" responses are not counted in Command or Action counts but do appear in Packet/Byte results (Tx/Rx counts, Tx/sec, Rx/Sec rates).

Change Notify Cancel

The **Change Notify Cancel** Action is used to cancel Directory monitoring initiated by a specific instance of a **Change Notify** Action. The Load DynamiX **Change Notify Cancel** Action uses the SMB2 "CANCEL" command to cancel a pending "CHANGE_NOTIFY" request. The "CHANGE_NOTIFY_CANCEL" command is unique among commands in the SMB2/3 command set in that an SMB2/3 Scenario sending a "CHANGE_NOTIFY_CANCEL" command to an SMB2/3 Server may not receive a response from the SMB2/3 Server for that command. An SMB2/3 Scenario is expected to receive at least one response from the server for all other SMB2/3 commands.

Most often, the **Change Notify Cancel** Action does not require a "CANCEL" response by the SMB2 server.

Specifically, after receiving the "CANCEL" request, the SMB2/SMB3 server will either:

1. (most often) send a "CHANGE_NOTIFY" response (to the prior "CHANGE_NOTIFY" request) with a "STATUS_CANCELLED" status error code, or
2. (rarely), not send any response at all (for example, when a related monitoring event has

- previously occurred and the "CHANGE_NOTIFY" request was already responded to), or
3. (in special situations), send a "CANCEL" response with an error status code.

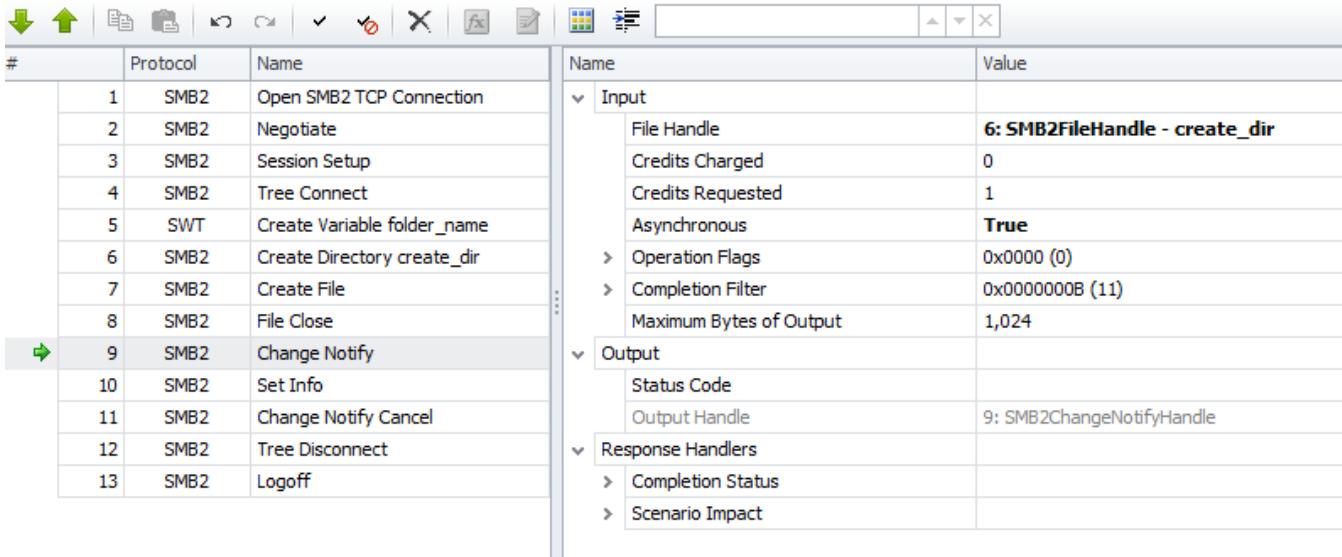
Note: Older versions of MS-SMB2 documentation contained incorrect content regarding the processing of "CANCEL" requests by SMB2 servers. Please use the most current MS-SMB2 documentation for correct content. A link to current SMB2/3 protocol reference material is provided in the [References and Terminology section](#).

The Load DynamiX Load Generation Appliance firmware does the best job possible dealing with the varied responses to the **Change Notify Cancel** Action described above. However, occasionally the Load DynamiX Appliance firmware might miscount some SMB2/3 "CANCEL" command statistics. Please contact support@loaddynamix.com if miscounted "CANCEL" command statistics regularly occur in Projects.

The following Scenario demonstrates the presence of a **Change Notify Cancel** Action in SMB2 Commands results.

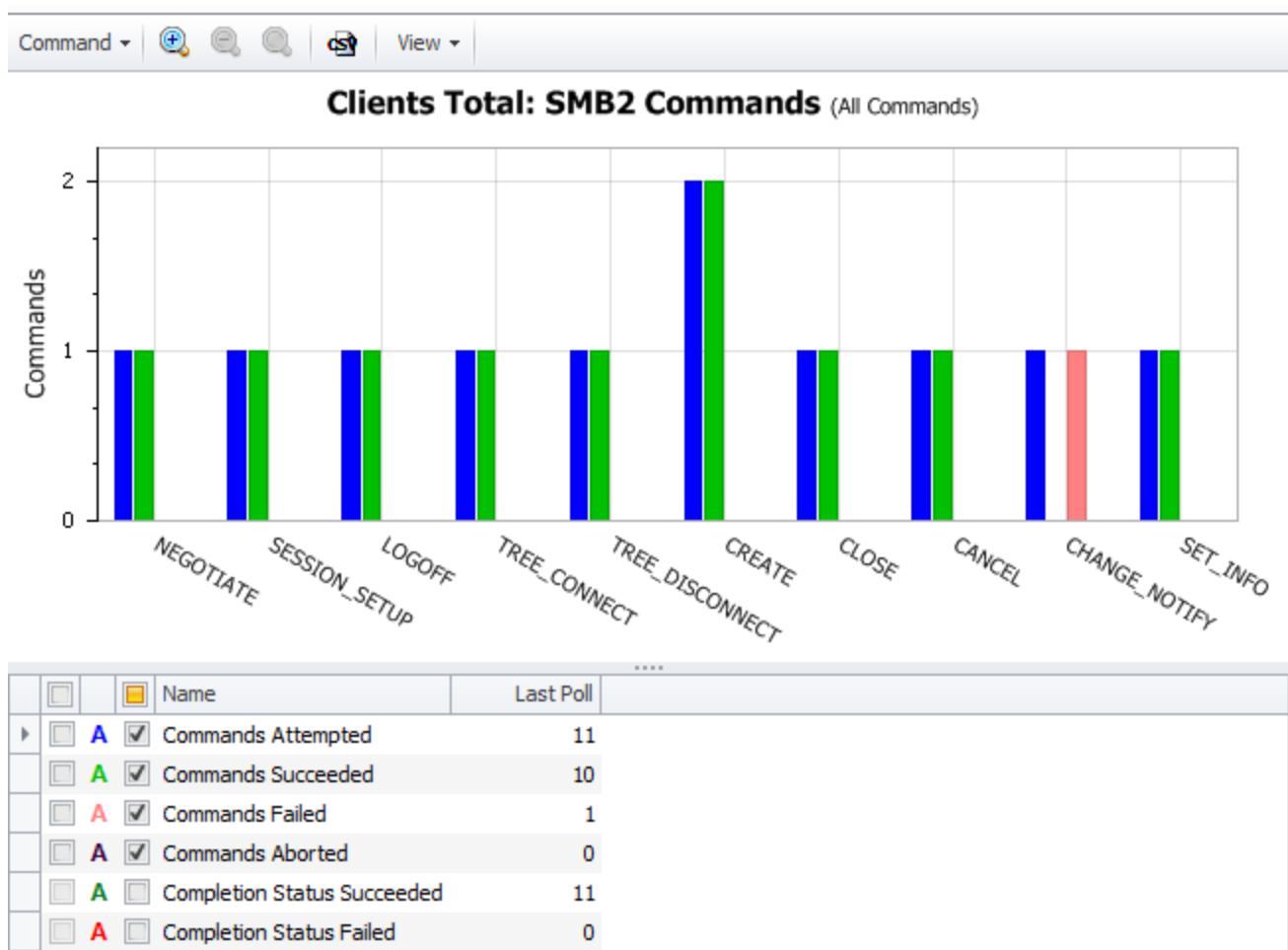
In this Scenario a **Change Notify Cancel** Action for a pending **Change Notify** Action is sent to an SMB2/3 server.

The **Change Notify** Action is canceled by that server.



#	Protocol	Name	Name	Value
1	SMB2	Open SMB2 TCP Connection	File Handle	6: SMB2FileHandle - create_dir
2	SMB2	Negotiate	Credits Charged	0
3	SMB2	Session Setup	Credits Requested	1
4	SMB2	Tree Connect	Asynchronous	True
5	SWT	Create Variable folder_name	> Operation Flags	0x0000 (0)
6	SMB2	Create Directory create_dir	> Completion Filter	0x0000000B (11)
7	SMB2	Create File	Maximum Bytes of Output	1,024
8	SMB2	File Close		
9	SMB2	Change Notify		
10	SMB2	Set Info	Status Code	
11	SMB2	Change Notify Cancel	Output Handle	9: SMB2ChangeNotifyHandle
12	SMB2	Tree Disconnect		
13	SMB2	Logoff		

The **Change Notify** Action is marked as failed due to the receipt of the **Change Notify Cancel** Action.



The Change Notify Cancel Action Input Parameters

The **Change Notify Cancel** Action takes as input the Output Handle of the **Change Notify** Action that is to be Canceled.

In the SMB2 Header inputs:

Credits Charged: See the SMB2 Credits discussion below.

Credits Requested: See the SMB2 Credits discussion below.

MessageID (Auto or Manual): When Auto is selected, the Load DynamiX Client fills in the appropriate ID value automatically. When Manual is selected, the Tester must provide the MessageID. The MessageID in the **Change Notify Cancel** Action must match the MessageID of the **Change Notify** Action to Cancel on the SMB2 Server.

AsyncID (Auto or Manual): When Auto is selected, the Load DynamiX Client fills in the appropriate ID value automatically. When Manual is selected, the Tester must provide the AsyncID. The AsyncID in the **Change Notify Cancel** Action must match the AsyncID of the **Change Notify** Action to Cancel on the SMB2 Server.

SessionID (Auto or Manual): When Auto is selected, the Load DynamiX Client fills in the appropriate ID value automatically. When Manual is selected, the Tester must provide the SessionID. The SessionID in the **Change Notify Cancel** Action must match the SessionID of the **Change Notify** Action to Cancel on the SMB2 Server. When Signing is enabled for a Scenario, the SessionID used in a **Change Notify Cancel** Action must match the SessionID

for a Pending **Change Notify** Action. If not, the **Change Notify Cancel** Action will fail. When Signing is enabled for a Scenario, if the SessionID of the **Change Notify Cancel** Action is set to 0, an error will be returned by the SMB2/3 server.

Signing (Auto or Force On or Force Off): When Auto is selected, the Load DynamiX Client signs the Action or not, automatically, depending on whether the Scenario indicated that Signing was required. When Force On is selected, Signing is forced to be On and the **Change Notify Cancel** Action will be signed. When Force Off is selected, Signing is forced to be Off and the **Change Notify Cancel** Action will not be signed.

SMB2 Header contents:

The **Change Notify Cancel** Action is the only SMB2 Action that Load DynamiX permits the Tester to modify the contents of the SMB2 Header portion of an SMB2 command packet other than setting the Credits Charged and Credits Requested header fields. The SMB2 Header fields that can be changed are MessageID, AsyncID, SessionID and Signing (see above for details).

Unless there are specific test goals related to the contents of these fields and server response, it is recommended that the default value of "Auto" be left in place as the input to these header fields.

Change Notify and **Change Notify Cancel** in use

Three examples of Change Notify and Change Notify Cancel behavior as viewed in MS NetMon output.

Example 1 Change Notify Cancel Action success - a **Change Notify** Action monitoring request canceled by a **Change Notify Cancel** Action.

In the NetMon output below, the **Change Notify Cancel** Action ("CANCEL" request) is sent to an SMB2 server, canceling the **Change Notify** Action (a "CHANGE_NOTIFY" request) sent earlier and resulting in a "STATUS_CANCELLED" error returned by the server in a "CHANGE_NOTIFY" response.

Description
SMB2:R NEGOTIATE (0x0), GUID={070EBB0D-C8B0-B6B8-4216-9BEE096E7769}
SMB2:C SESSION SETUP (0x1)
SMB2:R - NT Status: System - Error, Code = (22) STATUS_MORE_PROCESSING_REQUIRED SESSION SETUP (0x1), SessionFlags=0x0
SMB2:C SESSION SETUP (0x1)
SMB2:R SESSION SETUP (0x1), SessionFlags=0x0
SMB2:C TREE CONNECT (0x3), Path=\ 172.16.2.253\dump
SMB2:R TREE CONNECT (0x3), TID=0x1
SMB2:C CREATE (0x5), Sh(RWD), File=FOLDER_CN_1..028@#19
SMB2:R CREATE (0x5), FID=0x100000001(FOLDER_CN_1..028@#19)
SMB2:C CREATE (0x5), Sh(RWD), File=FOLDER_CN_1..028\FILE.TEST_1@#21
SMB2:R CREATE (0x5), FID=0x100000005(FOLDER_CN_1..028\FILE.TEST_1@#21)
SMB2:C CLOSE (0x6), FID=0x100000005(FOLDER_CN_1..028\FILE.TEST_1@#21)
SMB2:R CLOSE (0x6), File=FOLDER_CN_1..028\FILE.TEST_1@#21
SMB2:C CHANGE NOTIFY (0xf) ← Change Notify Action monitoring request sent to SMB2 Server
SMB2:R CHANGE NOTIFY (0xf) Interim Response, File=FOLDER_CN_1..028@#19 ← "Change Notify" response "STATUS PENDING"
SMB2:C SET INFORMATION (0x11), FID=0x100000001, Class=FileRenameInformation (10)
SMB2:R SET INFORMATION (0x11)
TCP:Flags=...A...., SrcPort=2000, DstPort=Microsoft-DS(445), PayloadLen=0, Seq=402780875, Ack=2185625013, Win=65535 (scale factor 0x8) = 16776960
ARP:Request, 172.16.1.83 asks for 172.16.93.100
BROWSER:Host Announcement, ServerName = QAWIN2008
DHCPV6:MessageType = SOLICIT
ARP:Request, 172.16.1.83 asks for 172.16.93.101
SMB2:C CANCEL (0xc) ← Change Notify Cancel Action sent to SMB2 Server
SMB2:C TREE DISCONNECT (0x4), TID=0x1
TCP:Flags=...A...., SrcPort=Microsoft-DS(445), DstPort=2000, PayloadLen=0, Seq=2185625013, Ack=402781019, Win=2053 (scale factor 0x8) = 525568
SMB2:R - NT Status: System - Error, Code = (288) STATUS_CANCELLED CHANGE NOTIFY (0xf) , File=FOLDER_CN_1..028@#19 ← "STATUS CANCELLED"
SMB2:R TREE DISCONNECT (0x4)
SMB2:C LOGOFF (0x2)
SMB2:R LOGOFF (0x2)

error response due to
"CANCEL" request
sent to SMB2 Server

Example 2 Change Notify Action failure - a Change Notify Action sent to an SMB2 server results in a "STATUS_NOTIFY_CLEANUP" error.

In the NetMon output below, the **Change Notify** Action ("CHANGE NOTIFY" request) returns the error "STATUS NOTIFY CLEANUP" in a "CHANGE NOTIFY" response due to the Scenario disconnecting ("TREE DISCONNECT" request in frame 36) from the Share before any "CHANGE NOTIFY" event occurred.

```

21      SMB2:C CHANGE NOTIFY (0xf) ← Change Notify Action monitoring request sent to SMB2 Server
22      SMB2:R CHANGE NOTIFY (0xf) Interim Response, File=FOLDER_CN_1..063@#15 ← "Change Notify" response "STATUS PENDING"
23      SMB2:C SET INFORMATION (0x11), FID=0x100000001, Class=FileRenameInformation (10)
24      SMB2:R SET INFORMATION (0x11)
25      TCP:Flags=...A..., SrcPort=2000, DstPort=Microsoft-DS(445), PayloadLen=0, Seq=1266814715, Ack=2414499249, W...
26      NbtNs:Registration Request for FILER3140-B <0x00> Workstation Service, 172.16.1.105
27      NbtNs:Registration Request for FILER3140-B <0x00> Workstation Service, 172.16.1.107
28      NbtNs:Registration Request for FILER3140-B <0x03> Messenger Service/Main name, 172.16.1.105
29      NbtNs:Registration Request for FILER3140-B <0x03> Messenger Service/Main name, 172.16.1.107
30      ARP:Request, 172.16.1.13 asks for 172.16.112.49
31      ARP:Request, 172.16.1.13 asks for 172.16.112.50
32      ARP:Request, 172.16.1.13 asks for 172.16.114.161
33      ARP:Request, 172.16.1.13 asks for 172.16.33.106
34      ARP:Request, 172.16.1.13 asks for 172.16.33.105
35      ARP:Request, 172.16.1.13 asks for 172.16.93.100
36      SMB2:C TREE DISCONNECT (0x4), TID=0x1
37      SMB2:R CHANGE NOTIFY (0xf) ← "STATUS NOTIFY CLEANUP" error response due to
38      SMB2:R TREE DISCONNECT (0x4) "TREE DISCONNECT REQUEST"
39      SMB2:C LOGOFF (0x2) request sent to SMB2 Server
40      SMB2:R LOGOFF (0x2)
41      TCP:Flags=...A...F, SrcPort=2000, DstPort=Microsoft-DS(445), PayloadLen=0, Seq=1266814859, Ack=2414499469, ...

```

Frame Details

- Frame: Number = 37, Captured Frame Length = 130, MediaType = ETHERNET
- Ethernet: Etype = Internet IP (IPv4), DestinationAddress:[00-15-17-CC-F7-E1], SourceAddress:[00-50-56-A2-64-82]
- Ipv4: Src = 172.16.2.253, Dest = 172.16.240.1, Next Protocol = TCP, Packet ID = 7753, Total IP Length = 116
- Tcp: Flags=...AP..., SrcPort=Microsoft-DS(445), DstPort=2000, PayloadLen=76, Seq=2414499249 - 2414499325, Ack=1266814787
- SMBOverTCP: Length = 72
- SMB2: R CHANGE NOTIFY (0xf)
 - SMBIdentifier: SMB
 - SMB2Header: R CHANGE NOTIFY (0xf), TID=0x0000, MID=0x0007, PID=0x0000, SID=0x10000011
 - StructureSize: 64 (0x40)
 - CreditCharge: 0 (0x0)
 - Status: 0x10B, Facility = FACILITY_SYSTEM, Severity = STATUS_SEVERITY_SUCCESS, Code = (267) STATUS_NOTIFY_CLEANUP
 - Command: CHANGE NOTIFY (0xf)

Example 3 Change Notify Action success - a Change Notify Action sent to an SMB2 server results in a "STATUS_SUCCESS" return code.

In the NetMon output below a **Change Notify** Action ("CHANGE NOTIFY" request) is sent to the SMB2 Server for Directory "Folder". A change is made to Directory "Folder" (frame 199) resulting in a "CHANGE NOTIFY" response indicating "STATUS_SUCCESS" that contains a data structure (FileInfo) that details what change was made to what (File "FILE0.TEST" was added to the folder "FOLDER")..

```

196      SMB2:C CHANGE NOTIFY (0xf) ← Change Notify Action monitoring request sent to SMB2 Server
197      SMB2:R CHANGE NOTIFY (0xf) Interim Response, File=FOLDER@#174 ← "Change Notify" response "STATUS PENDING"
198      TCP:Flags=...A..., SrcPort=Stock IXChange(527), DstPort=Microsoft-DS(445), PayloadLen=0, Seq=4265787315, Ack=3085214048, Win=65535 (scale factor 0x8) = 1.
199      SMB2:C CREATE (0x5), Sh(RWD), File=FOLDER\FILE0.TEST@#199
200      SMB2:C CREATE (0x5), Sh(RWD), File=FOLDER\FILE1.TEST@#200
201      SMB2:C CREATE (0x5), Sh(RWD), File=FOLDER\FILE2.TEST@#201
202      SMB2:R CHANGE NOTIFY (0xf) Interim Response, File=FOLDER@#184
203      TCP:Flags=...A..., SrcPort=Microsoft-DS(445), DstPort=ULP(522), PayloadLen=0, Seq=4056513177, Ack=1737581877, Win=251
204      SMB2:R CREATE (0x5), FID=0x4000000005(FOLDER\FILE2.TES@#201)
205      SMB2:R CHANGE NOTIFY (0xf) ← Change Notify response sent by SMB2 Server indicating STATUS_SUCCESS
206      SMB2:R CHANGE NOTIFY (0xf)

Frame Details
Frame: Number = 205, Captured Frame Length = 162, MediaType = ETHERNET
Ethernet: Etype = Internet IP (IPv4), DestinationAddress:[A0-36-9F-00-7F-09], SourceAddress:[00-50-56-A2-64-82]
Ipv4: Src = 172.16.2.253, Dest = 172.16.240.1, Next Protocol = TCP, Packet ID = 4637, Total IP Length = 148
Tcp: Flags=...AP..., SrcPort=Microsoft-DS(445), DstPort=Stock IXChange(527), PayloadLen=108, Seq=3085214048 - 
SMBOverTCP: Length = 104
SMB2: R CHANGE NOTIFY (0xf)
  SMBIdentifier: SMB
  SMB2Header: R CHANGE NOTIFY (0xf), TID=0x0000, MID=0x0005, PID=0x0000, SID=0x90000019
    StructureSize: 64 (0x40)
    CreditCharge: 0 (0x0)
    Status: 0x0, Facility = FACILITY_SYSTEM, Severity = STATUS_SEVERITY_SUCCESS, Code = (0) STATUS_SUCCESS
    Command: CHANGE NOTIFY (0xf)
    Credits: 0 (0x0)
    Flags: 0x3
    NextCommand: 0 (0x0)
    MessageId: 5 (0x5)
    AsyncId: 1 (0x1)
    SessionId: 17594601963545 (0x100090000019)
    Signature: Binary Large Object (16 Bytes)
  RChangeNotify:
    StructureSize: 9 (0x9)
    OutputBufferOffset: 72 (0x48)
    OutputBufferLength: 32 (0x20)
  FileInfoArray:
    FileInfo:
      NextEntryOffset: 0 (0x0)
      Action: (1) ADDED: The file was added to the directory.
      FileNameLength: 20 (0x14)
      FileName: FILE0.TEST

```

The screenshot shows a network traffic capture from NetworkMiner. It highlights the sequence of frames related to a 'Change Notify' operation. Frame 196 is a 'CHANGE NOTIFY' request sent to the SMB2 server. Frame 197 is an 'Interim Response' from the server. Frame 200 is a 'CREATE' request for a file named 'FILE0.TEST'. Frame 205 is the 'CHANGE NOTIFY' response from the server, which includes a 'SMB2 Header indicating STATUS_SUCCESS' and a 'FileInfo' array containing information about the newly added file. Red boxes and arrows point to specific fields in the response frame to explain their meaning.

Appendix: Virtual Appliance Constraints & Licensing

Appendix: Virtual Appliance Constraints, Licensing & License Server

Virtual Appliance License Concepts

Licensed Protocols: Like hardware Appliances, the Licenses used to control Virtual Appliances define the set of Protocols that the Virtual Appliance supports.

Licensed Appliance Instances: In addition to Protocols, the Virtual Appliance License also controls that number of Virtual Appliances that can be holding a License at any one time based on the number of Instances in the License and the per-port Constraints shown below (Standard and Plus). A Virtual Appliance License may contain allocations for both Standard and Plus Licenses or just Standard Licenses or just Plus Licenses.

For example, a License that allows 10 LDX-V instances could be configured for 5 Standard License instances and 5 Plus License instances or 10 Standard License instances or other combinations that add up to 10 instances. A License with 5 Standard License and 5 Plus Licenses allocated would enforce LDX-V operations such that at no time would more than 5 LDX-V instances be holding a Standard License or 5 LDX-V instances be holding Plus Licenses.

License Storage: Virtual Appliances do NOT maintain a copy of the License whereas hardware Appliances do. Virtual Appliance Licenses are stored in a special USB License Key ("dongle") and managed by a Virtual License Server.

Virtual Appliance (LDX-V) and License Server (LDX-VLS) Co-Requirements

VMware EXSi Server as host.

NTP Protocol for time and date synchronization between the Virtual Appliance and the Virtual License Server.

Virtual Appliance Constraints

The Load Dynamix Virtual Appliance is the Firmware of the Load Dynamix hardware Appliance running on a Virtual Machine on VMWare EXSi Server. The Virtual Appliance is not an exact duplicate of the Firmware that runs on the Load Dynamix hardware Appliance. Several Appliance capabilities are constrained and these constraints are controlled by the type of License that the Virtual Appliance uses. There are two categories of Virtual Appliance Licenses: Standard and Plus.

Capabilities	LDX-V Standard License	LDX-V Plus License
Concurrent Scenarios	1250	No Limit
Concurrent Connections	300	No Limit
Actions Per Second*	1250	No Limit
Bandwidth*	250Mbps	1Gbps
PCAP size	100mb linear, 32mb circular	100mb linear, 32mb circular
Test Duration	72 Hrs	72 Hrs

Client IP Addresses	256 for all Network Profiles in a Project	No Limit
Client MAC Addresses	100 for all Network Profiles in a Project	No Limit

* The observed Actions Per Second and Bandwidth depend on the compute, network and storage resources available on the Host.

The Virtual Appliance must be licensed and every protocol that is to be used by the Virtual Appliance must be included in that License.

LDX-V Licenses:

There are three types of Licenses:

Permanent: When a user licenses a Protocol, it is a Permanent License.

Temporary: If a user wishes to evaluate a protocol that they are not currently licensed for, a Temporary License can be issued that will allow the user to evaluate that new Protocol for a fixed period of time.

Emergency: If for some reason a user's license is not functioning as it should (ex: not allowing the use of a protocol that has been licensed), an Emergency License can be issued that will enable all Protocols for a fixed period of time.

One License feature that the Virtual Appliance has that the hardware Appliance does not is the maximum number of concurrently running Virtual Appliances. The feature constrains the number of Virtual Appliances that concurrently hold licenses. To maximize the use of Virtual Appliances, any Virtual Appliance instance that is not being used should be stopped on the EXSi Server that it is hosting it.

License Server Dongle:

The Virtual License Server Software requires the physical presence of a Virtual License Server Dongle which is a USB key used by the Virtual License Server Software to store the Licenses that it has been assigned. If the Dongle is not physically present then the Virtual License Server Software will not allow Virtual Appliances to execute. Any Virtual Appliances that were running at the time that the License Server Dongle was removed will continue to run until the current Project completes but will not allow new Projects to be executed.

Virtual License Server:

Key License Management tasks:

- Setting the Virtual License Server Software IP Address.
- Assigning the Virtual Appliance to a Virtual License Server.
- Adding a License to the Virtual License Server.

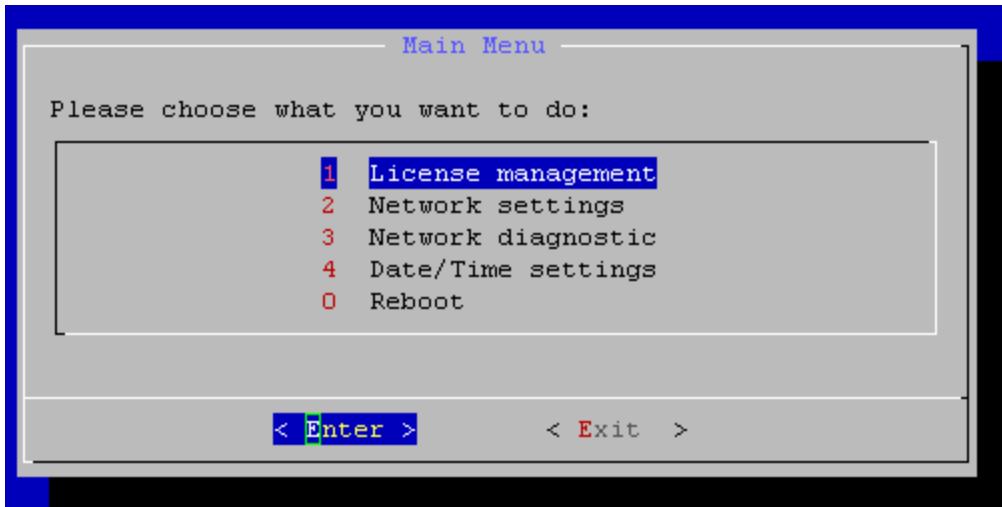
Other License management tasks exist

Task	Description
Remove License	Remove a license from a License Server. Once a license is removed, any of the Virtual Appliances that depend on that License will not be able to start any new Projects.

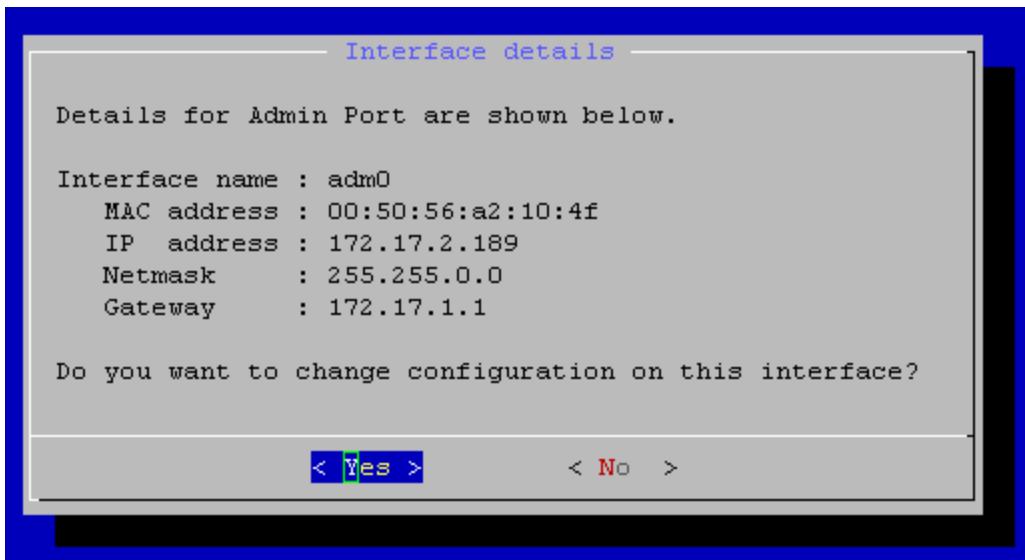
Revoke License Leases	Free up an existing License lease in a License Server for a specific Virtual Appliance. Allows other Virtual Appliances to acquire that License lease.
Emergency Mode	Add an Emergency License to a License Server if necessary

- Setting the Virtual License Server Software IP Address

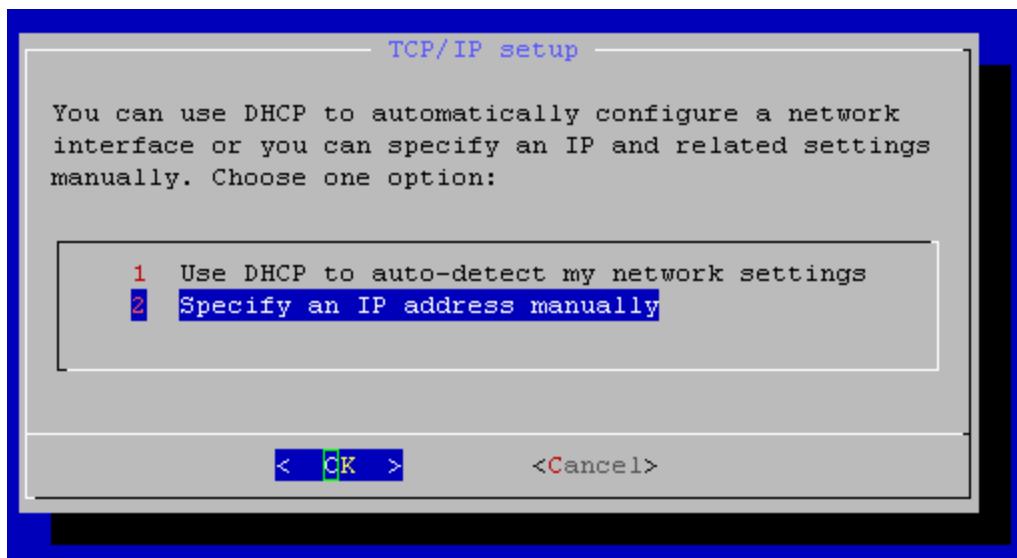
The Virtual Appliances require a single instance of a Virtual Appliance License Server which also runs on an EXSi Server. It is installed on the EXSi server and given an appropriate IP address and Licenses are installed on the License Server. The Virtual License Server is assigned its IP Address using the Virtual License Server Administrative Interface. To access the Virtual Appliance Administrative Interface, open the Console of the virtual machine using the VSphere client.



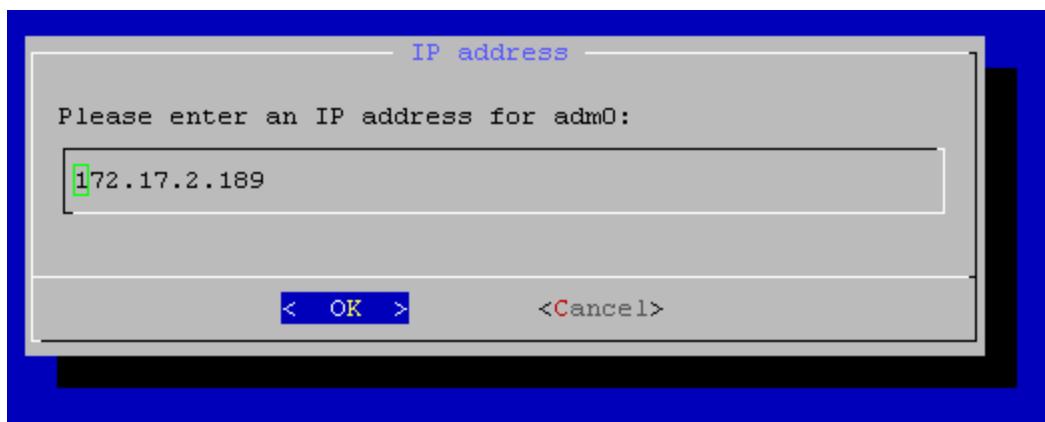
Select "Network Settings" and hit the Enter key.



Hit the Enter key to set the IP Address.



Select "Specify an IP address manually" because the Virtual License Server must have a fixed IP Address.



Enter the desired IP Address and click OK. Continue back to the Main Menu and close the Console session.

- Assigning the Virtual Appliance to a Virtual License Server

Virtual Appliances are made aware of the IP address of the Virtual License Server via the Virtual Appliance Administrative Interface.

To access the Virtual Appliance Administrative Interface, start an SSH session using the IP Address of the Virtual Appliance (or through opening the Console of the virtual machine using the VSphere client interface). Here the IP Address of the Virtual Appliance is 172.17.2.180.

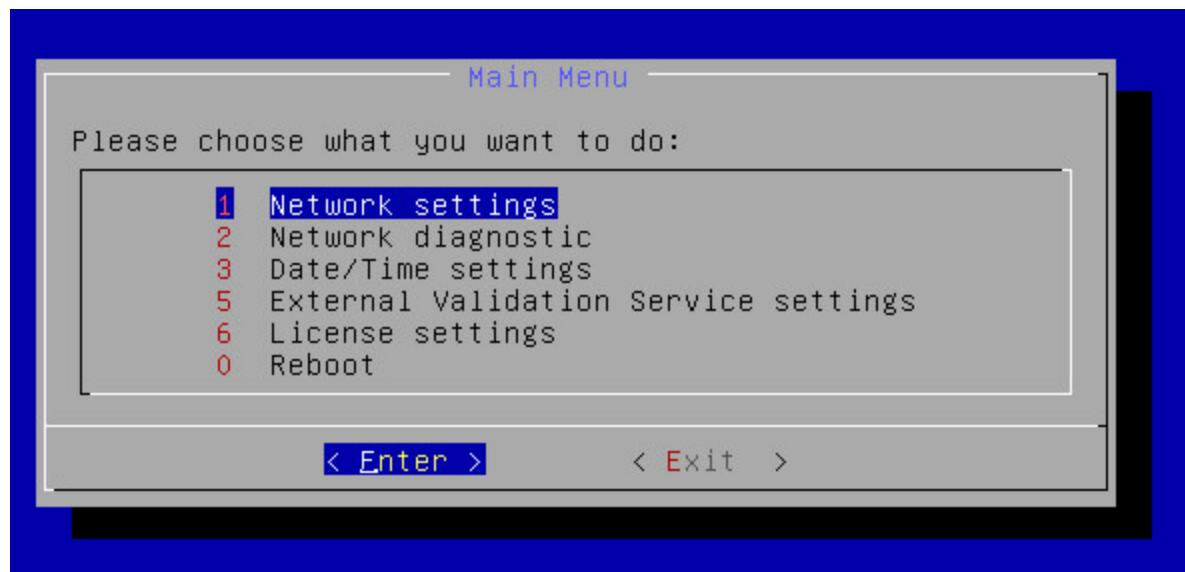
172.17.2.180 - PuTTY

v7 (c) 2009-2014 Load Dynamix

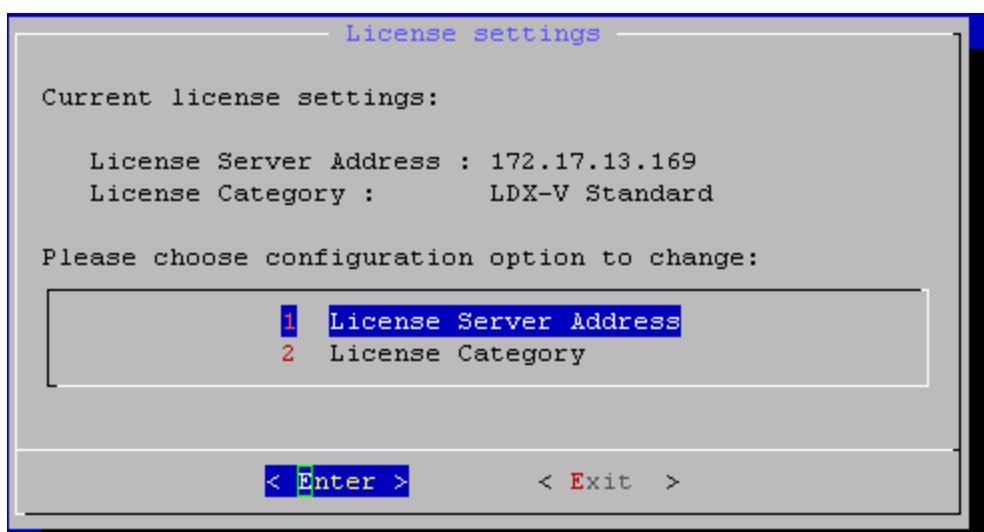
Hit Enter to
go to main menu.

```
+++ 0ooo oo      0      0000      0      00000
++++ 0ooo ooooo   0      0      0      0 0      0
+++++ 0oo oooooooooo   0      0      0 0      0 0      0
+++++ oooooooooooooooooo   0      0      0 0      0 0      0
+++++ oooooooooooooooooo   0      0      0 0000000 0      0
++++++      oooo 00000000 0000 0      0 00000
++++++      0000 0oo 0000000 00 00 00 00 00 00 00 00 00 00
+ ++++++ 0000000 oo 0000000 00 00 00 00 00 00 00 00 00 00 00
++++++      000000000 o 00 00 00 00 0000 0000 0000 0000 0000 0000
          000000000 00 00 00 0000 0000 00 00 0000 0000 00 0000
0000000000000000 00 00 00 00 0000 00 00 00 0000 00 00 00 00 00
00000000000000 0 00 00 00 00 0000 00000000 00 0 00 00 00 00 00
000000000 000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

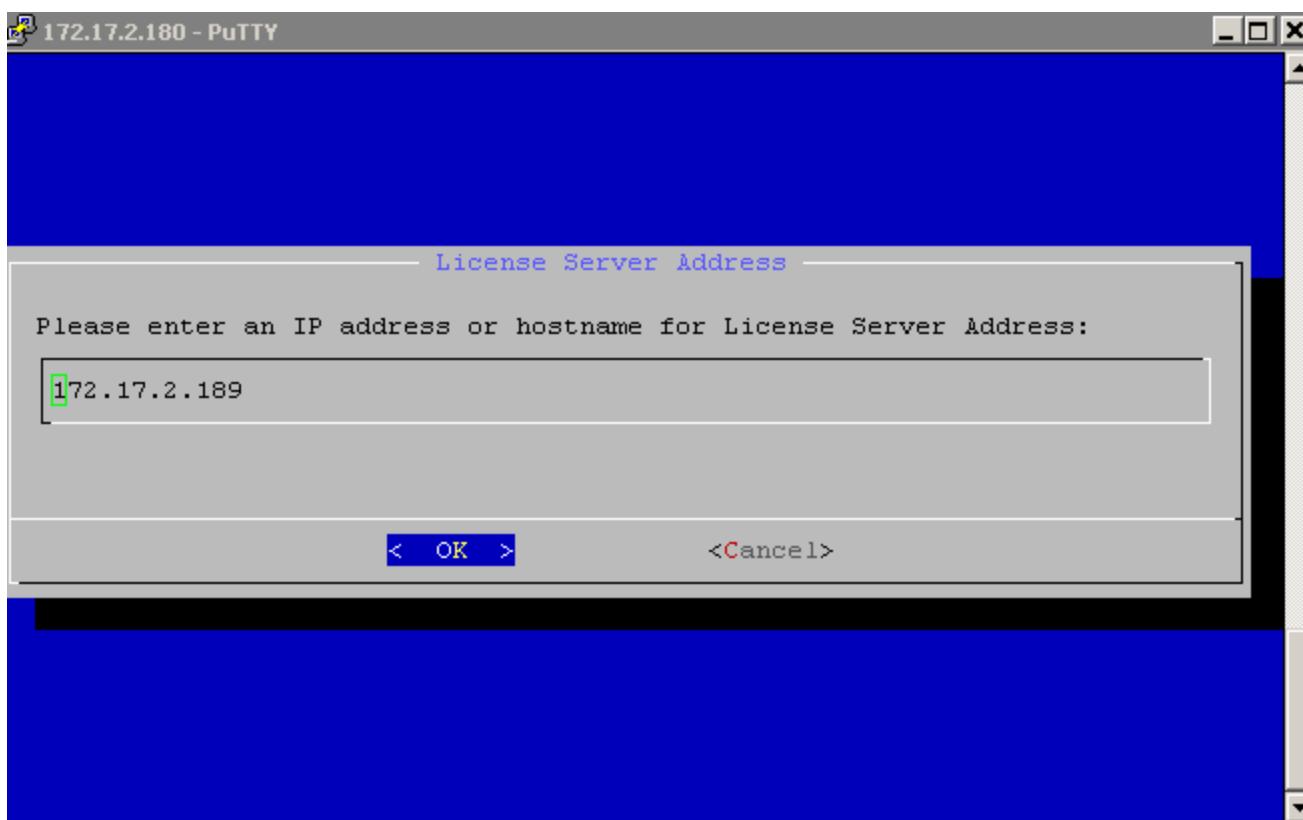
Hit the Enter key.



Use the down arrow to highlight "License Settings" and hit the Enter key.



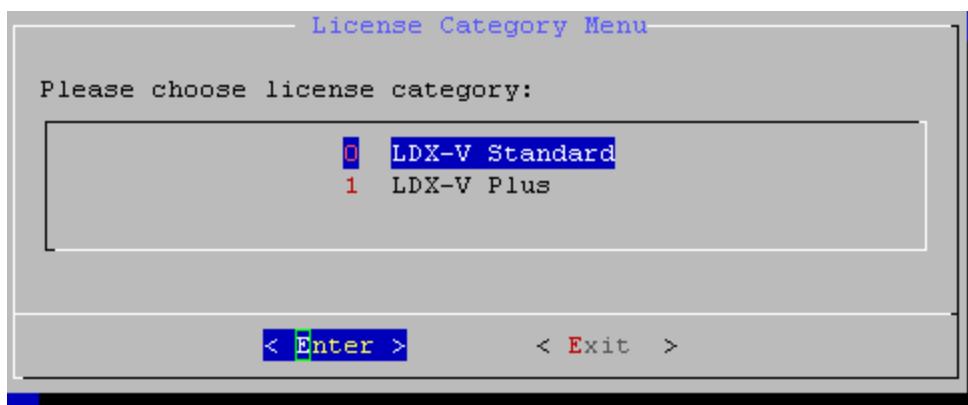
Assuming that an IP Address of the Virtual License Server is something other than 172.17.3.189, click <Enter> to go to the next screen.



Type in the IP Address of the Virtual License Server and hit the Enter key to save the new Virtual Server IP Address.

For a Virtual Appliance License, the user must also select the Category of the Virtual Appliance License: Standard or Plus.

Select 2 :License Category to get the following interface:



Select the Category of License desired (Virtual Appliance Licenses may contain both Standard and Plus Licenses) and click <Enter>.

Now continue back to the Main Menu and close the SSH session.

- Adding a License to the Virtual License Server

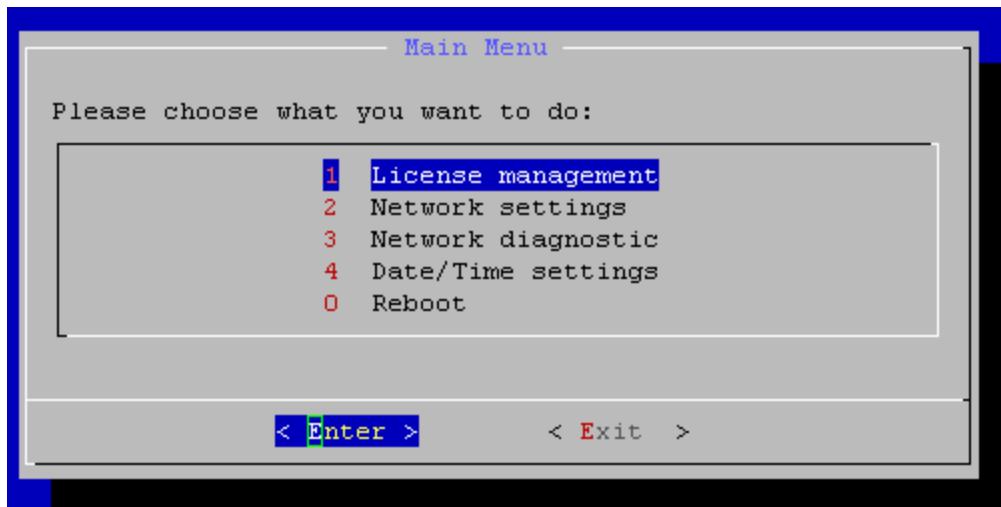
The first step in adding a new license to a Virtual License Server is to download the license from Load DynamiX onto a local filesystem (ex: C:\Documents\Temp on a Windows workstation) and then opening the license file with a text editor such as Notepad and Copying the license string in preparation for Pasting it into the Add License window below.

To add a license to a Virtual License Server, access the Virtual Appliance Administrative Interface by starting an SSH session using the IP Address of the Virtual License Server (or through opening the Console of the virtual machine using the VSphere client interface). Here the IP Address of the Virtual License Server is 172.17.2.189.

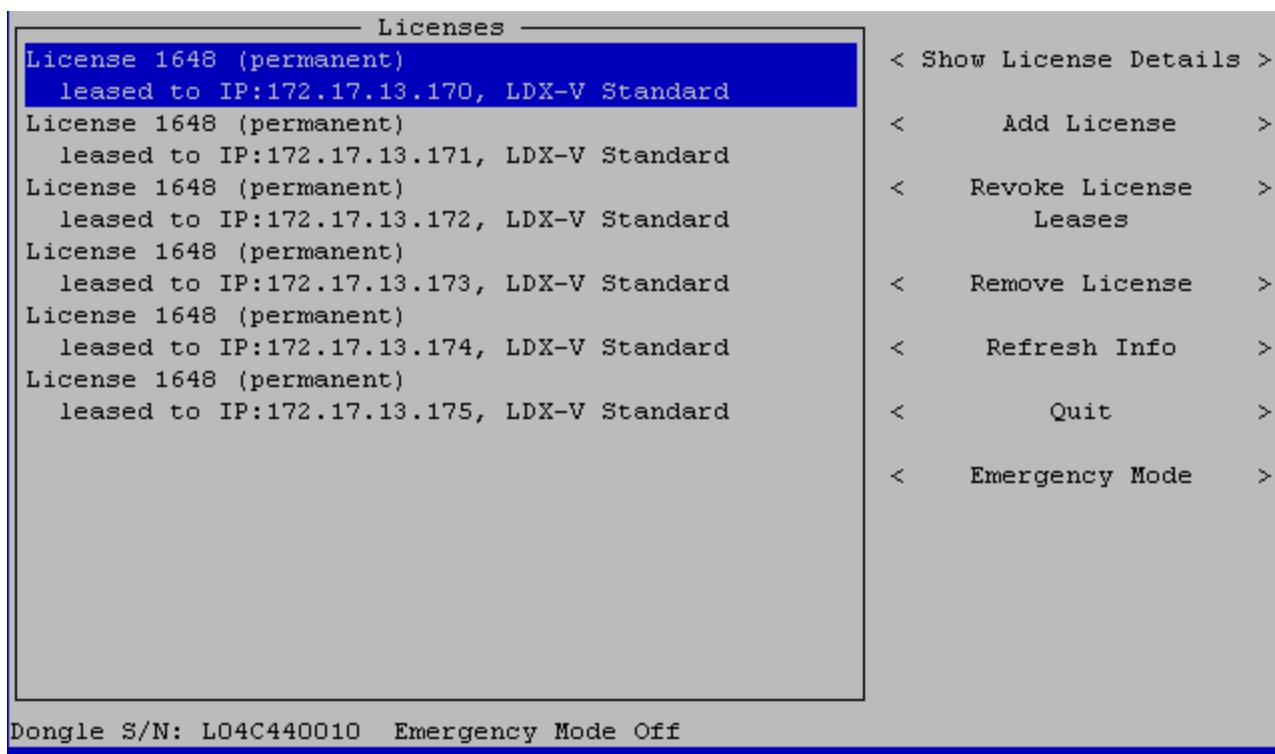
```
v5 (c) 2009-2014 Load DynamiX
Hit Enter to
go to main menu.

     +--+ 0000 00      0      0000      0      00000
     +---+ 0000 0000      0      0      0      0      0
     +----+ 000 00000000      0      0      0      0      0
     +----+ 0000000000000000      0      0      0      0      0
     +----+ 0000000000000000      0      0      0      0      0
     +----+ 0000 0000 0000      0      0      0      00000
     +----+ 0000 000      0      0000      0      0      0
+  +----+ 000000 00      000000 00      00 00      00      00      00 00 00 00
+-----+ 0000000000 0 00      00 00 00 00      0000      000      000 00 00 00 00
          0000000000 00      00      00      00 0000 00      00 00 00 00 00 00
          00000000000000 00      00      00      00 0000 00      00 00 00 00 00 00
          00000000000000 0 00      00      00      00 0000 0000 00      0 00 00 00 00
          00000000000000 00000000      00      00 00      00 00      00 00 00 00 00
```

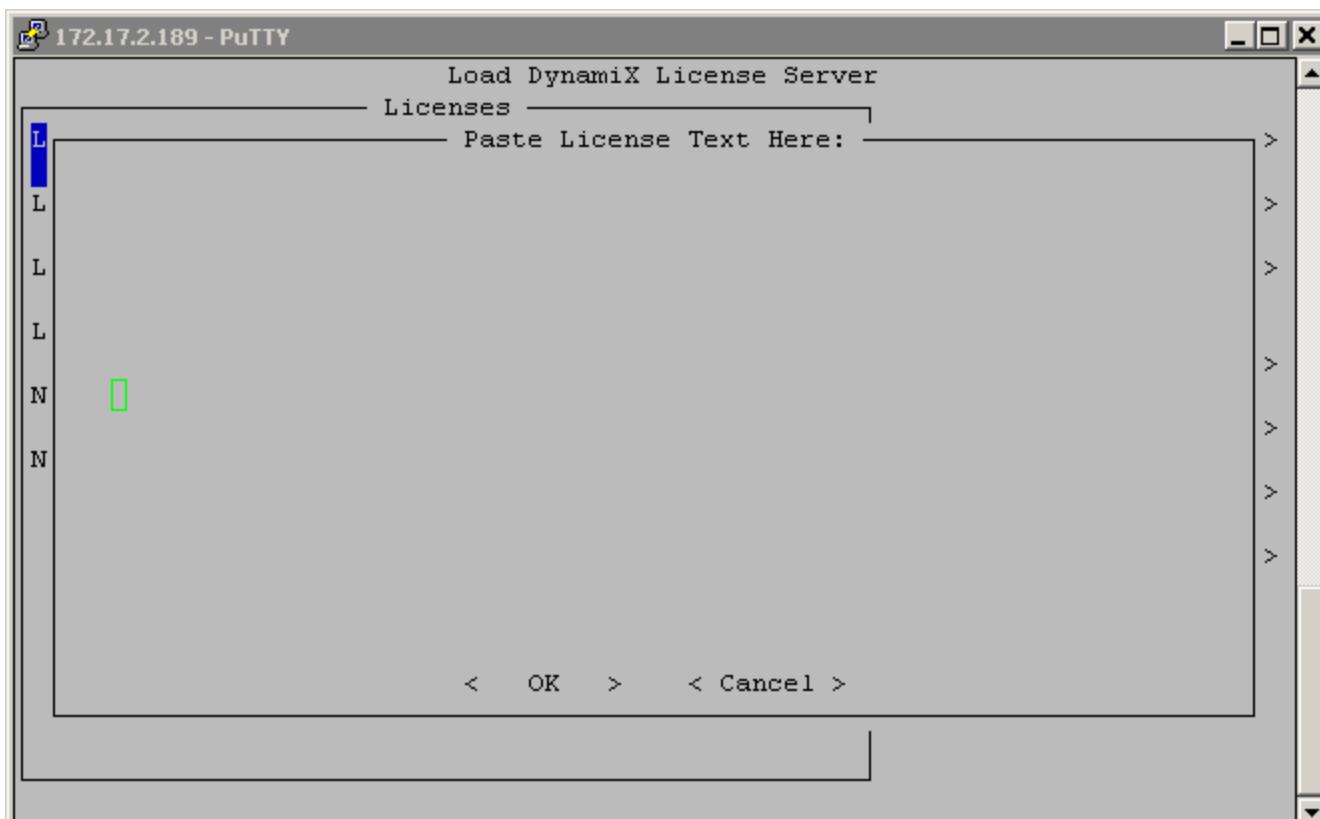
Hit the Enter key to get to the Main Menu



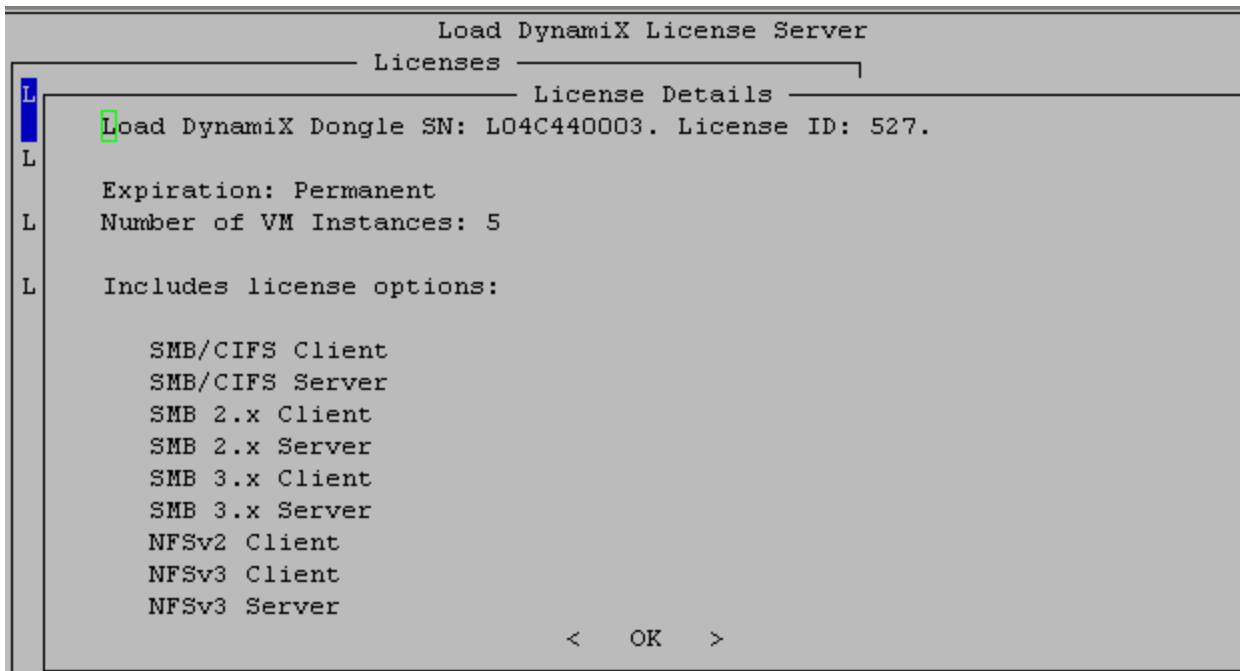
Hit the Enter Key to open the License Management interface



Use the Right Arrow key to get to the Menu items on the right then use the Down Arrow key to select Add License then hit the Enter key.



Copy (if not already copied from the text editor as noted above and) Paste the License information sent to you by Load DynamiX and click on OK to save.



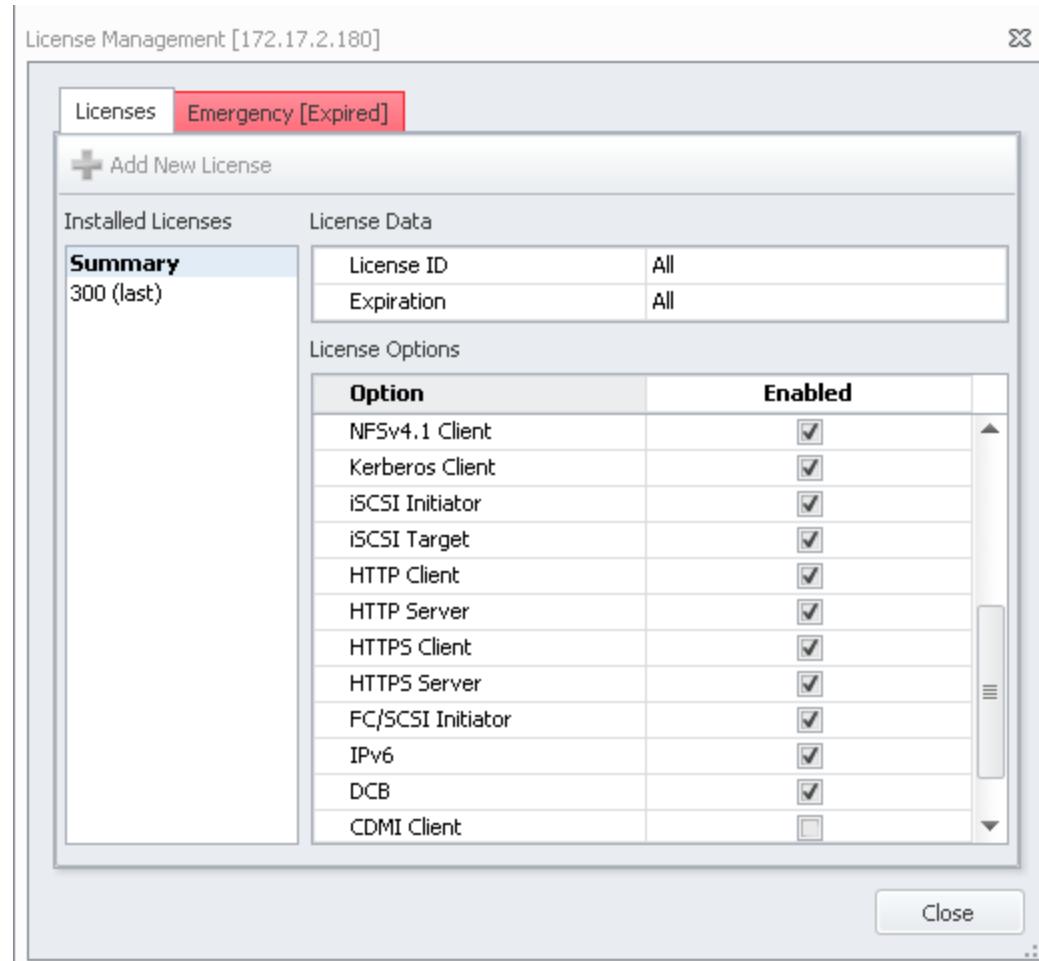
Return to the Main Menu and close the session.

Virtual License Viewed by TDE

The Virtual Appliance License used by a Virtual Appliance can be visualized by a Tester when running the TDE. In the Ports & Appliances window > Appliances Tab, is a button for License Management.

Highlight the Virtual Appliance IP address and click the License  button to see the Virtual

Appliance License. The License displayed below is the same as the license displayed above (both are license ID 300).



Reference: CIFS/SMB Commands and Behaviors

Reference: Load DynamiX CIFS-SMB Command and Behaviors

ACTION (SMB Protocol Command Name) (SMB Command hex value)
Close SMB TCP Connection
Open SMB TCP Connection
Compound Action
Session Requests
Echo (SMB_COM_ECHO) (0x2B)
Negotiate (SMB_COM_NEGOTIATE) (0x72)
Session Logoff (SMB_COM_LOGOFF_ANDX) (0x74)
Session Setup (SMB_COM_SESSION_SETUP) (0x73)
TRANS 2 Get File System Information (TRANS2_QUERY_FS_INFORMATION) (0x32/0x03)
Tree Connect (SMB_COM_TREE_CONNECT_ANDX) (0x75)
Tree Disconnect (SMB_COM_TREE_DISCONNECT) (0x71)
File Requests
Create Or Open File (SMB_COM_NT_CREATE_ANDX) (0xA2)
Create Temporary File (SMB_COM_CREATE_TEMPORARY) (0x0E)
File Close (SMB_COM_CLOSE) (0x04)
File Delete (SMB_COM_DELETE) (0x06)
File Flush (SMB_COM_FLUSH) (0x05)
File Lock/Unlock (SMB_COM_LOCKING_ANDX) (0x24)
File NT Rename (SMB_COM_NT_RENAME) (0xA5)
File Read (SMB_COM_READ_ANDX) (0x2E)
File Rename/Core (SMB_COM_RENAME) (0x07)
File Seek (SMB_COM_SEEK) (0x12)
File Write (SMB_COM_WRITE_ANDX) (0x2F)
NT Transact IOCTL (NT_TRANSACT_IOCTL) (0x0A/0x02)
NT Transact Query Security Descriptor (NT_TRANSACT_QUERY_SECURITY_DESC) (0x0A/0x06)
NT Transact Set Security Descriptor (NT_TRANSACT_SET_SECURITY_DESC) (0x0A/0x03)

TRANS 2 Get File Information (TRANS2_QUERY_FILE_INFORMATION) (0x32/0x07)**TRANS 2 Get Path Information** (TRANS2_QUERY_PATH_INFORMATION) (0x32/0x05)**TRANS 2 Set File Information** (TRANS2_QUERY_SET_FILE_INFORMATION) (0x32/0x08)**TRANS 2 Set Path Information** (TRANS2_QUERY_SET_PATH_INFORMATION) (0x32/0x06)

Directory Requests

Check Directory (SMB_COM_CHECK_DIRECTORY) (0x10)**Delete Directory** (SMB_COM_DELETE_DIRECTORY) (0x01)**Find Close** (TRANS2_FIND_CLOSE) (0x34)**Find First** (TRANS2_FIND_FIRST2) (0x32/0x01)**Find Next** (TRANS2_FIND_NEXT2) (0x32/0x02)**Get DFS Referral** (TRANS2_GET_DFS_REFERRAL) (0x32/0x11)**NT Transact Notify Change** (NT_TRANSACT_NOTIFY_CHANGE) (0xA0/0x04)

Other - LANMAN 1.0

File Close and Tree Disconnect (SMB_COM_CLOSE_AND_TREE_DISC) (0x31)**File Copy** (SMB_COM_COPY) (0x29)**File Lock and Read** (SMB_COM_LOCK_AND_READ) (0x13)**File Move** (SMB_COM_MOVE) (0x2A)**File Write and Close** (SMB_COM_WRITE_AND_CLOSE) (0x2C)**File Write and Unlock** (SMB_COM_WRITE_AND_UNLOCK) (0x14)**Get Path Information** (SMB_COM_QUERY_INFORMATION2) (0x23)**Open File/LANMAN** (SMB_COM_OPEN_ANDX) (0x2D)**Set Path Information** (SMB_COM_SET_INFORMATION2) (0x22)

Other - Core Dialect

Create Directory (SMB_COM_CREATE_DIRECTORY) (0x00)**Create File/Core** (SMB_COM_CREATE) (0x03)**Create New File/Core** (SMB_COM_CREATE_NEW) (0x0F)**File Lock/Core** (SMB_COM_LOCK_BYTE_RANGE) (0x0C)**File Read/Core** (SMB_COM_READ) (0x0A)**File Unlock/Core** (SMB_COM_UNLOCK_BYTE_RANGE) (0x0D)**File Write/Core** (SMB_COM_WRITE) (0x0B)**Get Path Info/Core** (SMB_COM_QUERY_INFORMATION) (0x08)

Open File (SMB_COM_OPEN) (0x02)
Process Exit (SMB_COM_PROCESS_EXIT) (0x11)
Set Path Info/Core (SMB_COM_SET_INFORMATION) (0x09)
Tree Connect/Core (SMB_COM_TREE_CONNECT) (0x70)

A link to CIFS-SMB protocol reference material is provided in the [References and Terminology section](#).

Creating Directories using CIFS-SMB Protocol

The CIFS-SMB protocol does not have a "CREATE DIRECTORY" command like the SMB2 protocol does. To create directories using CIFS-SMB, use the **Create Or Open File** Action with the following input field settings:

Path Name > Path == <Directory Name>
 Open Flags > Open Target Directory == False
 Create Options > Flag as Directory == True

CIFS-SMB Keep-Alive Messages

Load DynamiX Open SMB TCP Connection and Start SMB Server Actions allow the Tester to enable or disable the use of SMB Keep Alive messages by either the Client or Server. The default for SMB Clients is to Disable SMB Keep Alive messages and for SMB Servers to Enable SMB Keep Alive messages. The impact of enabling SMB Keep Alive for Client or Server is that whenever the TCP Inactivity Timer expires in an SMB session, an SMB Keep Alive message is sent to see if the connection is still alive. If no response is received, the connection will be closed.

CIFS-SMB File Write Action

Block Size cannot exceed 64Kbytes.

CIFS-SMB Session Setup Virtual Circuit

CIFS-SMB Session Setup Action contains an input field named Virtual Circuit. The default value for Virtual Circuit == 0 which has the behavior of closing any open virtual circuits/sessions for the IP address requesting the new session. The default value behavior is impactful in tests where there are a small number of available IP addresses but a reasonable number of valid Users. If it is necessary for the same IP address to have multiple sessions open with a CIFS-SMB server then the value of the Virtual Circuit input must be > 0 and must be unique for each unique Virtual Circuit

CIFS-SMB Multiplexing ("Credits")

In the CIFS-SMB Protocol, "Credits" is referred to as Multiplexing. The CIFS-SMB MaxMPXCount value is an indicator from the CIFS-SMB server as to how many outstanding operations the server will honor. The only place that Multiplexing is put into use in the Load DynamiX product is in the CIFS-SMB Negotiate Action where there is a field named "Honor MPX Count [True/False]". If the user sets the Honor MPX Count field to True then the Client will honor whatever the server specifies as the maximum number of outstanding requests. If it set to False then the Client will ignore input from the server regarding MaxMPXCount. It is recommended that for Compound Actions that result in large number of actions being send to a server, that Honor MPX Count be set == True.

CIFS-SMB Maximum Request Size

CIFS-SMB Negotiate Action has an input named Maximum Request Size. The field is preloaded with values of:

- Unlimited
- Auto
- 4356 (0x1104)
- 16644 (0x4104)
- 65535 (0xFFFF)
- 131071 (0x1FFFF)

This field is used to control the Maximum amount of data that can be requested in any one CIFS-SMB Action. Unlimited (default) indicates that maximum 4 byte value (4 Billion bytes) is requested. Auto indicates that the maximum needed is calculated and supplied. The rest are integer specifications of the Maximum Request Size.

CIFS/SMB Sample Projects

The screenshot shows the configuration dialog for the 'Client CIFS-SMB Write Kerberos.client_scenario'. The left pane displays a sequence of 13 actions:

#	Protocol	Name
1	Kerberos	Open Kerberos Connection
2	Kerberos	AS-REQ
3	Kerberos	TGS-REQ
4	Kerberos	Close Kerberos Connection
5	SMB	Open SMB TCP Connection
6	SMB	Negotiate
7	SMB	Session Setup
8	SMB	Tree Connect
9	SMB	Create Or Open File
10	SMB	File Write
11	SMB	File Close
12	SMB	Tree Disconnect
13	SMB	Session Logoff

The right pane shows the configuration details for action 2 (AS-REQ):

Name	Value
Connection Handle	Default
Client principal	= @UP(0,A) + @STRING(@\$STORAGE.QAD)
Password	= @UP(0,B)
Service principal	krbtgt/\$STORAGE.QAD@\$STORAGE.QAD
Proxiable	False
Renewable	False
Forwardable	False
Anonymous	False
Output	
Output Handle	2: Kerberos TGT

The screenshot shows the configuration dialog for the 'Client CIFS-SMB Read Rate.client_scenario'. The left pane displays a sequence of 8 actions:

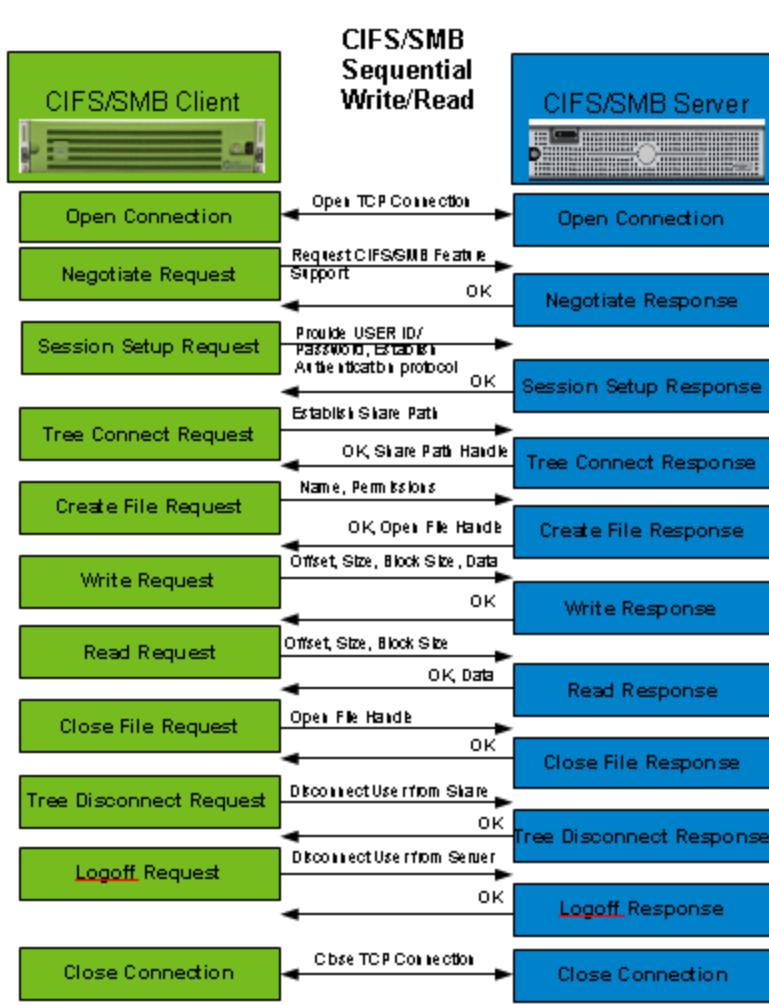
#	Protocol	Name
1	SMB	Open SMB TCP Connection
2	SMB	Negotiate
3	SMB	Session Setup
4	SMB	Compound Action
...	SMB	Tree Connect
...	SMB	Create Or Open File
5	SMB	Compound Action
...	SMB	File Read
...	SMB	File Close
6	SMB	Tree Disconnect
7	SMB	Session Logoff
8	SMB	Close SMB TCP Connection

The right pane shows the configuration details for action 5 (Compound Action):

Name	Value
Input	
Header Flags	Unicode
Enabled	Enabled
Response Handlers	
Completion Status	
Scenario Impact	

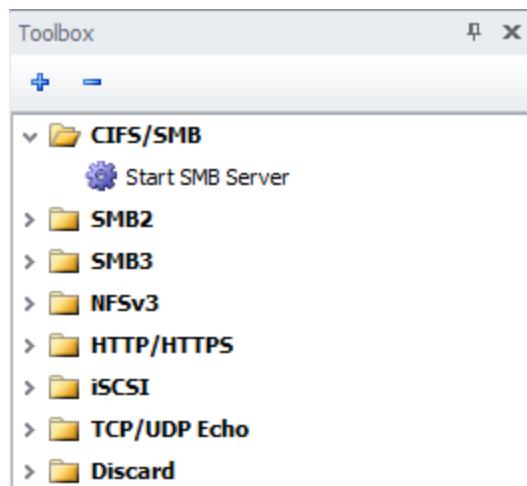
CIFS/SMB Project Flow

The following is the CIFS/SMB client/server interaction flow for a simple sequential write read Project.



CIFS-SMB Start Server Action

The Load DynamiX Appliance firmware supports a CIFS-SMB server emulation. The CIFS-SMB server is instantiated in a Server Scenario using the Start SMB Server Action and providing, at a minimum, an IP address for the Server.



CIFS-SMB Statistics

See the [Advanced Concepts: Test Execution Rules](#) for a complete list of the statistics generated when CIFS-SMB Actions are used in a Scenario.

Reference: SMB2 Commands and Behaviors

Reference: Load DynamiX SMB2 Commands and Behaviors

ACTION (SMB2 Protocol Command Name) (SMB2 Command Hex Value)

Close SMB2 TCP Connection

Open SMB2 TCP Connection

Compound Action

Session Requests

Echo (SMB2 ECHO) (0x000D)

Logoff (SMB2 LOGOFF) (0x0002)

Negotiate (SMB2 NEGOTIATE) (0x0000)

Session Setup (SMB2 SESSION_SETUP) (0x0001)

Tree Connect (SMB2 TREE_CONNECT) (0x0003)

Tree Disconnect (SMB2 TREE_DISCONNECT) (0x0004)

File Requests

Create File [file default settings] (SMB2 CREATE) (0x0005)

Durable File Connect (SMB2 Create) (0x0005)

File Close (SMB2 CLOSE) (0x0006)

File Flush (SMB2 FLUSH) (0x0007)

File Lock (SMB2 LOCK) (0x000A)

File Read (SMB2 READ) (0x0008)

File Write (SMB2 WRITE) (0x0009)

Query Info (SMB2 QUERY_INFO) (0x0010)

Set Info (SMB2 SET_INFO) (0x0011)

Directory Requests

Create Directory [directory default settings] (SMB2 CREATE) (0x0005)

Change Notify (SMB2 CHANGE_NOTIFY) (0x0015)

Change Notify Cancel (SMB2 CANCEL) (0x000C)

Query Directory (SMB2 QUERY_DIRECTORY) (0x000E)

IOCTL Requests

Get DFS Referral Request (SMB2 IOCTL DFS_REFERRAL) (0x000B/0x00060194)

IOCTL Request (SMB2 IOCTL) (0x000B)

Read Hash Request (SMB2 IOCTL SRV_READ_HASH) (0x000B/0x001441bb)

Resiliency Request (SMB2 IOCTL NETWORK_RESILIENCY_REQUEST) (0x000B/0x001401D4)

Resume Key Request (SMB2 IOCTL SRV_REQUEST_RESUME_KEY) (0x000B/0x00140078)

Server Side Copy (SMB2 IOCTL SRV_COPYCHUNK_COPY) (0x000B/0x001440F2)

A link to the SMB2.0/3.0 protocol reference material is provided in the [References and Terminology section](#).

Olocks and Leases

Load DynamiX support for CIFS-SMB and SMB2 commands include support for both Olocks (Opportunistic Locks in CIFS-SMB and SMB2) and Leases (SMB2.1). Olocks and Leases both provide support for efficient caching of server file data by clients and better performance in high latency networks. Leases States are somewhat more configurable than Oplock Levels.

Olocks

Olock Levels

- Batch - used by clients that want to do all file operations (open, close, read, write, etc) locally and keep the file open on the server regardless of the file's state on the client.
- Exclusive - A Read/Write lock. Allows the client to Read and Write data to a file locally knowing that no other clients have requested access to that same file. Also called Level I in some CIFS-SMB documentation.
- Level II - Allows caching of Read requests (only supported in SMB2 create file requests).

When selecting Olock Level there are only the three choices possible in the SMB2 protocol

Create File Action and two choices possible in CIFS-SMB **Create or Open File** Action.

CIFS-SMB and SMB2/2.1 Servers will send the client an Olock Break request when the Server needs to downgrade or eliminate the current Olock Level held by the client. To get Olock Batch Level behavior, both Batch and Exclusive Levels must be selected. When Olock Level other than None is selected in the SMB2 **Create File** Action, the Tester is given options on how that Scenario is to respond to Olock Break messages from the server.

CIFS-SMB scenarios can choose how to respond to Olock Breaks from the server: None (do nothing), Acknowledge, Abort Scenario, Close File.

SMB2 scenarios can choose how to respond to Olock Breaks from the server based on the kind of Break received: If level II is available or if no Olock level is available.

Leases

Leases are a feature of the SMB2.1 protocol and extend the Olock features by adding somewhat greater flexibility to the levels of caching locks that are available and what the response to the Lease Break is. To use Leases, the SMB2 Negotiate command for a Scenario must request SMB2.1 support. Lease fields in a **Create File** request are enabled by setting the Olock Level field to Lease. When a Lease is requested, the file create command provides the Lease State desired, what to do when a Lease Break is received and whether or not to Acknowledge receipt of Lease Breaks before doing whatever is specified in the Action on Lease Break field (Yes or No).

Lease State:

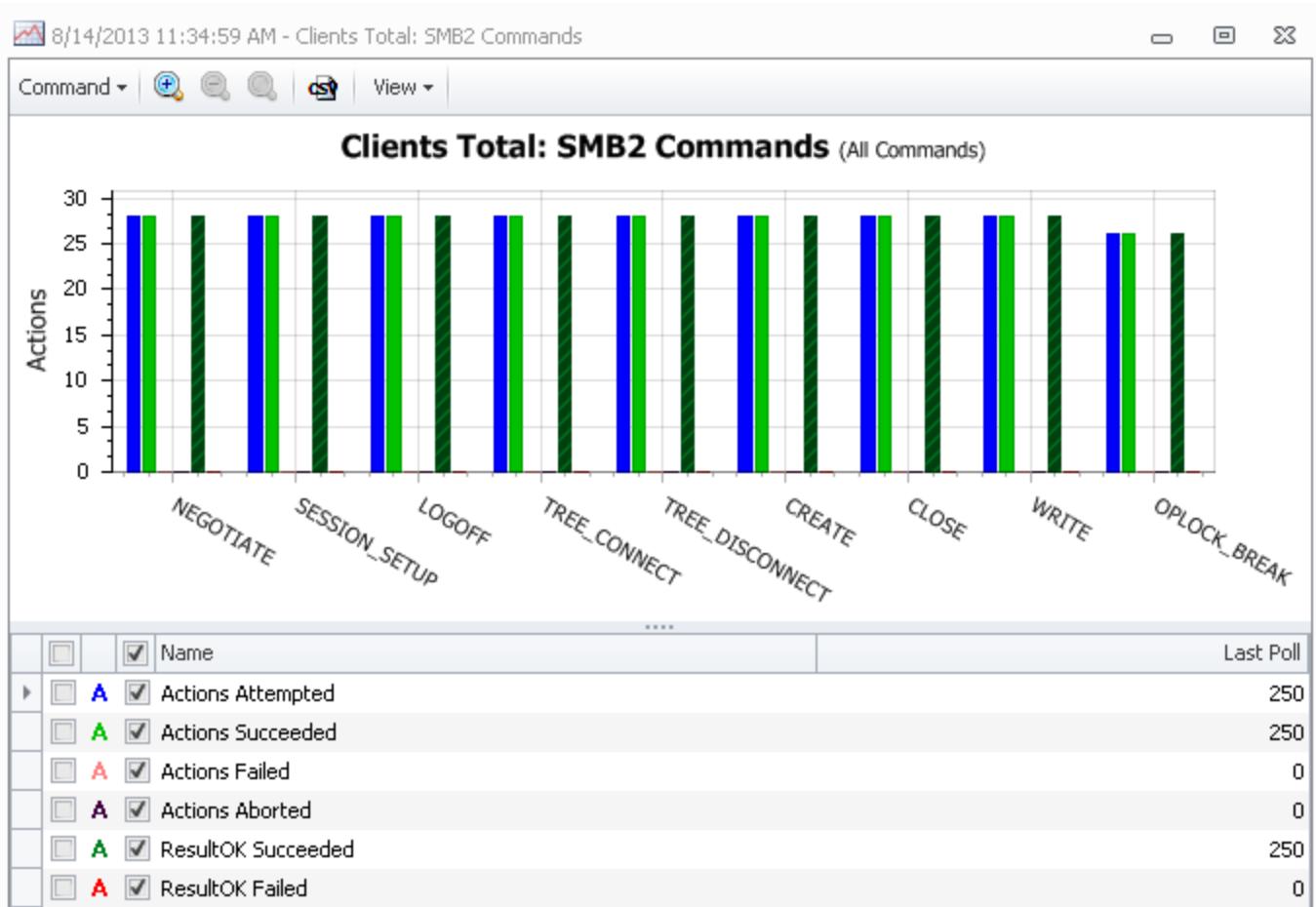
- Read Caching - like the Level II Olock
- Write Caching - when combined with Read Caching is like an Exclusive Olock
- Handle Caching - when combined with Read Caching and Write Caching like a Batch Olock

The SMB 2.1 Server will send the client an Lease Break request when the Server needs to downgrade or eliminate the current Lease State held by the client. The Load DynamiX Scenario can choose what to do when the Lease Break is received (either continue or abort the Scenario) and whether to Acknowledge the Lease Break or not (Yes or No).

Allowable Lease State combinations for Windows SMB2.1 Servers are Read Caching, Read + Handle Caching, Read + Write Caching, Read + Write + Handle Caching and None (all States set to False). Hex values for these combinations are 0x0 (none), 0x07 (read+write+handle), 0x05 (read+write), 0x03 (read+handle), and 0x01 (read). It is possible to enter other hex values in this field but the results when interacting with a Windows SMB2.1 server will be an invalid parameters error and undetermined with other SMB2.1 servers.

Statistics

Oplock Breaks and Lease Breaks received by Scenarios are captured in SMB and SMB2 Command statistics as if they were individual commands. Every Oplock or Lease Break received the command graph will appear on the commands graph as if it was a successfully executed command as well as in the Client Log file. The screenshot below shows how Oplocks are displayed in the SMB2 command graph.



Extended SMB2 Command Support

As the SMB2 Protocol continues to evolve, Load DynamiX will continue to add support for new SMB2 features by implementing specific Actions to address those SMB2 features or updating existing Actions. The following new SMB2 Actions have been added to the supported SMB2 Client Actions.

Extended SMB2 Actions	Operation
Change Notify (and Change Notify Cancel)	Watch for changes to a directory or to a whole directory tree based on some criteria (e.g. change of name, change of size, last write date/time, etc). Cancel watch for changes. See below and Appendix: Change Notify and Change Notify Cancel Actions for details.
Server Side Copy	Initiate a server-side copy of one file to another specifying source and target offsets and the copy length
Resume Key Request	Precursor operation to a Server Side Copy to establish the Handle to the source file of the copy
Resiliency Request	Request that a file be "resilient" (will be kept open for a time period should the client disconnect)
Durable File Connect	Request to open a file that has previously been opened with the Request Durability property set to True in Create File
Read Hash Request	Request Hash Information that contains a set of Hashes that can be used to retrieve the contents of a specific file using the branch cache.

The following existing Actions have been extended

Extended SMB2 Actions	Extension
Set Info	Added Security information that can be set
Query Info	Added Security information that can be queried for
Create File	Ability to set Request Durability to True (files that remain open across Client disconnects)

Action Details

Change Notify

This Action requests an SMB2/SMB3 server to notify a Load DynamiX Scenario when the properties and/or contents of a Directory change. The Operations Flags and Completion Filter Properties (see below and [Appendix: Change Notify and Change Notify Cancel Actions](#) for details) control which Directory properties and content to monitor. The "CHANGE_NOTIFY" command is among the most unique commands in the SMB2 command set, in that a single "CHANGE_NOTIFY" request may receive one or two responses from an SMB2/3 server. The first response is either an error, indicating some kind of problem with the "CHANGE_NOTIFY" command that was sent, or a "STATUS_PENDING" response indicating acceptance of the command by the server and the beginning of the requested monitoring. Typical server behavior is for the second response to be a "CHANGE_NOTIFY" response containing either success or some error indication. See below and [Appendix: Change Notify and Change Notify Cancel Actions](#) for details.

To assist the Tester in creating Scenarios that mimic the use of SMB2/3 commands in the way that MS Windows uses them, Load DynamiX has introduced a special input parameter "Asynchronous" to the **Change Notify** Action, with two options (modes): "False" (Synchronous), and "True" (Asynchronous). This parameter controls the internal behavior of the Load DynamiX SMB2/3 Scenario when it executes a **Change Notify** Action (note: this "Asynchronous" parameter should not be

confused with the SYNC / ASYNC variants of the SMB2 Packet Header, referenced in the "CHANGE_NOTIFY" specification.) A link to the SMB2/3 protocol reference material is provided in the [References and Terminology section](#).

In either mode, when the Load DynamiX Scenario sends the "CHANGE_NOTIFY" request to a server, the Scenario expects a "CHANGE_NOTIFY" response returned from that server containing a "STATUS_PENDING" error code. If the "CHANGE_NOTIFY" response, containing some status indicator, is not received, for whatever reason - TCP error or server error, timeout, etc., the **Change Notify** Action will be marked as Aborted.

Note: If an event occurs on a server with an active "CHANGE_NOTIFY" request that is applicable to that request, within a very short time following the receipt of that request, the server may skip the "STATUS_PENDING" response altogether and respond immediately with a status code indicating the nature of the event. The indication could be "STATUS_SUCCESS", "STATUS_CANCELLED", etc., depending on the nature of the event.

Synchronous mode ("False") - default

In Synchronous mode, after receiving a "CHANGE_NOTIFY" response containing "STATUS_PENDING", the Scenario will wait until it receives a final "CHANGE_NOTIFY" response from the SMB2/3 server, containing either:

1. "STATUS_SUCCESS", indicating that a monitored change has occurred on the watched Directory
2. or, some error response ("STATUS_NOTIFY_CLEANUP", "STATUS_CANCELLED", etc.).

If the "CHANGE_NOTIFY" response is not received within the TCP Inactivity timeout window (when "Keep-Alive" is disabled), the Action and Scenario will abort.

Asynchronous mode ("True")

In Asynchronous mode, after receiving a "CHANGE_NOTIFY" response containing "STATUS_PENDING", the Scenario will continue executing, expecting a final "CHANGE_NOTIFY" response from the SMB2/3 server, containing either:

1. "STATUS_SUCCESS", indicating that a monitored change has occurred on the watched Directory
2. or, some error response ("STATUS_NOTIFY_CLEANUP", "STATUS_CANCELLED", etc.).

If the "CHANGE_NOTIFY" response is not received before the Scenario completes executing, the **Change Notify** Action will be marked as Failed.

Why use Asynchronous mode vs Synchronous mode?

In **Change Notify** Synchronous mode, the **Change Notify** Action causes the Scenario to pause until the server finally responds (other than "STATUS_PENDING") to the **Change Notify** Action (see **Change Notify** Action Results below). Synchronous mode could be used to create an SMB2/3 Scenario that will execute a specific set of Actions once the server responds to the **Change Notify** Action. Two real world examples:

1. A database sync process where any change to a database requires that other database-related files or storage are synchronized immediately.
2. A Windows Explorer window, displaying the content of a server Directory, reflects any change made to that Directory.

Synchronous mode allows a Scenario to more correctly emulate the behavior of the "CHANGE_NOTIFY" request as used by a Windows SMB2/3 Client. Synchronous mode allows a Tester to create a single Scenario, using SMB2 Actions (**Change Notify**, **Change Notify Cancel** and Directory Read/Write) and Scenario Control Actions (Events, Threads, If/Else If/Else/End If, etc.) to generate a variety of "CHANGE_NOTIFY" status code responses and execute specific Action sequences based on those codes.

Change Notify Asynchronous mode could be used to exercise the SMB2/3 "CHANGE_NOTIFY" server functionality itself (error handling, change detection, etc.). Asynchronous mode may be simpler to use in a Scenario but cannot be used to guarantee the Actions that will be executed when an event occurs that causes a "CHANGE_NOTIFY" response to be sent.

Change Notify Action Input Parameters

Asynchronous: See the detailed explanation above.

File Handle: The output handle of the **Create Directory** Action used to create or open the Directory that is to be monitored.

Credits Charged: See the SMB2 Credits discussion below.

Credits Requested: See the SMB2 Credits discussion below.

Operation Flags:

Watch Tree (True or False): Watch Tree == True will also monitor changes in any file or sub-directory contained in the Directory referenced by File Handle (see above). See [Appendix: Change Notify and Change Notify Cancel Actions](#) for details.

Completion Filter:

Specifies which types of changes the Client wishes to receive "CHANGE_NOTIFY" responses for. See [Appendix: Change Notify and Change Notify Cancel Actions](#) for details.

Change Notify Action Results

The **Change Notify** Action inputs determine the Directory that is being monitored and, the Directory properties that are being monitored. When an SMB2/3 server responds to a **Change Notify** Action, it responds with a "CHANGE_NOTIFY" response. Included in that response packet is an error code:

- "STATUS_PENDING" - this server response indicates that the "CHANGE_NOTIFY" request has been accepted and Directory monitoring has begun,
- "STATUS_NOTIFY_CLEANUP" - if the server terminates the "CHANGE_NOTIFY" request before a monitored change has occurred, or
- "STATUS_CANCELLED" - if a **Change Notify Cancel** Action is sent to the SMB2/3 server before a change occurred, or
- (other some error code indicating that there was a problem with the "CHANGE_NOTIFY" request itself)

or

- "STATUS_SUCCESS" - the monitored change has occurred, information embedded in the "CHANGE_NOTIFY" response indicates the kind of change that occurred.

The **Change Notify** Action will be counted as successful if "STATUS_SUCCESS" is received. The **Change Notify** Action will be counted as failed if "STATUS_NOTIFY_CLEANUP" or "STATUS_CANCELLED" or any of the other error codes indicating that there was a problem with the "CHANGE_NOTIFY" request are received. "STATUS_PENDING" responses are not counted in Command or Action counts but do appear in Packet/Byte results (Tx/Rx counts, Tx/sec, Rx/Sec rates).

See [Appendix: Change Notify and Change Notify Cancel Actions](#) for details.

Change Notify Cancel

The **Change Notify Cancel** Action is used to cancel Directory monitoring initiated by a specific instance of a **Change Notify** Action. The Load DynamiX **Change Notify Cancel** Action uses the SMB2 "CANCEL" command to cancel a pending "CHANGE_NOTIFY" request. If the **Change Notify Cancel** Action is successful, the Scenario will receive a "CHANGE_NOTIFY" response containing the error code "STATUS_CANCELLED" from the SMB2/3 server and the **Change Notify** Action will be marked as Failed in the Results Explorer data.

Change Notify Cancel Action Input Parameters

The **Change Notify Cancel** Action takes as input the Output Handle of the **Change Notify Action** that is to be Canceled. In the SMB2 Header inputs:

Credits Charged: See the SMB2 Credits discussion below.

Credits Requested: See the SMB2 Credits discussion below.

MessageID (Auto or Manual): When Auto is selected, the Load DynamiX Client fills in the appropriate MessageID value automatically. When Manual is selected, the Tester must provide the MessageID. See [Appendix: Change Notify and Change Notify Cancel Actions](#) for details.

AsyncID (Auto or Manual): When Auto is selected, the Load DynamiX Client fills in the appropriate AsyncID value automatically. When Manual is selected, the Tester must provide the AsyncID. See [Appendix: Change Notify and Change Notify Cancel Actions](#) for details.

SessionID (Auto or Manual): When Auto is selected, the Load DynamiX Client fills in the appropriate SessionID value automatically. When Manual is selected, the Tester must provide the SessionID. See [Appendix: Change Notify and Change Notify Cancel Actions](#) for details.

Signing (Auto or Force On or Force Off): When Auto is selected, the Load DynamiX Client signs the Action or not, automatically, depending on whether the Scenario indicated that Signing was required. Force On and Force Off allow the Tester to directly control the signing behavior of the **Change Notify Cancel** Action. See [Appendix: Change Notify and Change Notify Cancel Actions](#) for details.

Change Notify Cancel Action results

The Load Dynamix SMB2/3 Scenario may not receive a response from an SMB2/3 server when a "CHANGE_NOTIFY_CANCEL" command is sent. An SMB2/3 Client is expected to receive at least one response from the server for all other SMB2/3 commands.

Specifically, after receiving a "CANCEL" request, the SMB2/3 server will either:

1. (most often) send a "CHANGE_NOTIFY" response (to the "CHANGE_NOTIFY" request that is being canceled) with a "STATUS_CANCELLED" status code, or
2. (rarely), not send any response at all (for example, when a related monitoring event has previously occurred and the "CHANGE_NOTIFY" request was already responded to), or
3. (in special situations), send a "CANCEL" response with an error status code.

Note: Older versions of MS-SMB2 documentation contained incorrect content regarding the processing of "CANCEL" requests by SMB2 servers. Please use the most current MS-SMB2 documentation for correct content. A link to current SMB2/3 protocol reference material is provided in the [References and Terminology section](#).

The Load DynamiX Load Generation Appliance firmware does the best job possible dealing with the varied responses to the **Change Notify Cancel** Action described above. However, occasionally the Load DynamiX Appliance firmware might miscount some SMB2/3 "CANCEL" command statistics. Please contact support@loaddynamix.com if miscounted "CANCEL" command statistics regularly occur in Projects.

See [Appendix: Change Notify and Change Notify Cancel Actions](#) for details regarding the **Change Notify Cancel** Action.

Contact support@loaddynamix.com for more details regarding the **Change Notify Cancel** and **Change Notify** Actions.

Resiliency Request

This Action is used to tell a SMB2 server how long a specified open file is to remain Resilient (open across Client disconnects).

Properties:

File Handle - The file Handle for the open file that will be made Resilient.

Timeout - The period of time (in milliseconds) during which the file identified by File Handle will be made Resilient.

Resume Key Request

This action is used to create the File Handle for the Source file for a Server Side Copy Action.

Properties:

File Handle - The file Handle for the open file that will be the source file for the copy.

Resume Key Handle - the Handle that will be used by the Server Side Copy Action as the Handle for the source of the copy operation.

Server Side Copy

This action is used to tell a SMB2 server initiate a server side copy of data or "FSCTL_SRV_COPYCHUNK". This is issued as an IOCTL Request by the Load DynamiX client. The Load DynamiX Client will then wait for the "SRV_COPYCHUNK_RESPONSE" from the server. Once it is received the action will be complete.

Properties:

Source Resume Key Handle - The source file for the copy, this handle is obtained from a "Resume Key Request" action issued earlier in the scenario.

Destination File Handle - The destination file for the copy, the handle is obtained from a previously opened file.

Source and Target Offset - The starting position to start the copy from the source and position to start copying to the destination.

Length - The amount of data to copy

Durable File Connect

This action is used to tell a SMB2 server open a file that had previously been opened or created with a connection that been closed for some reason.

Properties:

Durable File Handle - The file Handle for a file that was opened previously with Request Durability set equal to True.

Read Hash Request

This action allows to request Hash Information that contains a set of Hashes that can be used to retrieve the contents of specific files using the branch cache.

Properties:

Hash Version - the Branch Cache version 1 or 2

Hash Retrieval Type - Hash Based or File Based

Length - maximum length, in bytes, of the hash information returned in response

Offset - the offset of the data to be retrieved, in bytes, from the beginning of the hash information

SMB2 Keep-Alive Messages

Load DynamiX Open SMB2 TCP Connection and Start SMB2 Server Actions allow the Tester to enable or disable the use of SMB2 Keep Alive messages by either the Client or Server. The default for SMB2 Clients is to Disable SMB2 Keep Alive messages and for SMB2 Servers to Enable SMB2 Keep Alive messages. The impact of enabling SMB2 Keep Alive for Client or Server is that whenever the TCP Inactivity Timer expires in an SMB2 session, an SMB2 Keep Alive message is sent to see if the connection is still alive. If no response is received, the connection will be closed.

SMB2 Credits

A number of outstanding simultaneous requests that the client can have on a particular connection is determined by the number of credits granted to the client by the server.

All discussion below only makes sense if you are planning to make a test with outstanding requests by using one of the following in your scenario:

- Async or Thread blocks with SMB2 commands inside
- SMB2 Compound requests
- SMB2 commands which required multiple credits:
 - SMB2 Writes or SMB2 Reads with Block Size bigger than 64KB
 - SMB2 Query Directory with Output Size bigger than 64KB
 - etc
- Commands which can generate multiple outstanding requests internally (build in NOR support).

Load Dynamix introduced two additional fields into each SMB2 action for give tester control on SMB2 Credits flow:

SMB2 Header	
Credits Charged	Automatic
Credits Requested	Automatic

Credits Charged: Default value == Automatic.

Automatic means that LDX calculates number of credits to be charged correspondingly to MS documentation requirements, dependently on action executed:

The CreditCharge of an SMB2 operation is computed from the payload size (the size of the data within the variable-length field of the request) or the maximum size of the response.

$$\text{CreditCharge} = (\max(\text{SendPayloadSize}, \text{Expected ResponsePayloadSize}) - 1) / 65536 + 1$$

User can overwrite Automatic by setting any integer value to simulate positive and negative tests. In that case please note that accordingly to MS-SMB2 documentation:

In the SMB 2.0.2 dialect, this field MUST NOT be used and MUST be reserved. The sender MUST set this to 0, and the receiver MUST ignore it. In the SMB 2.1 dialect, this field indicates the number of credits that this request consumes.

Credits Requested: Default value == Automatic.

Meaning of Automatic values described below inside of SMB2 Negotiate Action : Credits Pool Size section.

User can overwrite Automatic by setting any integer value to simulate positive and negative tests. In that case please note that accordingly to MS-SMB2 documentation:

To maintain its current number of credits, the client MUST set CreditRequest to the number of credits that it will consume in sending this request, as specified in sections 3.2.4.1.5 and 3.2.4.1.6. To increase or decrease this number, the client MUST request the server to grant more or fewer credits than will be consumed by the current request.

The client MUST NOT decrease its credits to zero, and SHOULD request a sufficient number of credits to support implementation-defined local requirements.

SMB2 Negotiate Action : Credits Pool Size

The SMB2 **Negotiate** Action has an input field named **Credit Pool Size**, it's default value == 31. The purpose of this field is to always try to keep this number of credits from SMB2 Server. Every next SMB2 Action with **Credits Requested** field predefined *Automatic* mode will try to fill in from SMB2 Server the number of Credits to obtain **Credit Pool Size** in total. In this case Client is always charged by credits for executing any further action.

SMB2 Negotiate Action: Honor Credits

Default value == True.

NOTE: Honor Credits == True will not allow client to send more outstanding requests than in Credit Pool specified. It might affect Load Profile settings and number of concurrently executed threads, block sizes etc. See details below.

The SMB2 **Negotiate** Action has an input field named Honor Credits. The purpose of this field is to inform the SMB2 Client whether to honor Credit Requested responses from SMB2 servers where outstanding requests are concerned:

- *False* - credit responses from Servers are fundamentally ignored. It means that even if Charge Limit was reached, Load Dynamix SMB2 Client will continue sending:
 - All asynchronous commands (inside Async blocks, Thread blocks, NOR etc);
 - Compound requests will be sent full - contains all actions in one compound request as they are specified in scenario;
 - All SMB2 commands which might use multiple credits (Writes, Reads, Query Directories etc.) will be sent with parameter values (Block size, Output size) as it is specified in scenario.
- *True* - credit responses from Servers are taken into account. If number of outstanding requests needed to be sent for executing Action is greater than current number of credits granted by Server to Client, then Client will not send more requests than allowed.
It's applicable to:

- Actions out of limit of available credits inside of Async blocks will be suspended until new credits will be granted;
- No more Actions than number of available credits inside of Thread blocks will be executed simultaneously, the rest of Actions from other Thread blocks without enough granted credits will be suspended;
- The number of sent NOR will be no more than number of available credits;
- Compound requests will be divided into parts equal to the number of credits currently available for the client;
- Write/Read/Query Directory work correctly only for requests less than 64Kb.

SMB2 Negotiate Action Capabilities Input

The SMB2 **Negotiate** Action has an input field named Capabilities. The purpose of this field is for the Client to inform the SMB2 Server of six specific capabilities that it is interested in using (see screenshot above):

- Distributed File System
- Leases
- Large_MTU
- Multi-Channel (Windows 8/SMB 3.0 feature)
- Persistent Handles (Windows 8/SMB 2.2 feature)
- Directory Leases (Windows 8/SMB 2.2 feature)
- Encryption

The default values for these capabilities is False. Regardless of the setting of these flags, the SMB2 server will respond with what it supports by setting the corresponding bit in its Negotiate response to a "1" if it supports the capability.

The screenshot shows a software interface for managing SMB2 sessions. On the left, a tree view lists session actions: Open SMB2 TCP Connection, Negotiate, Session Setup, Tree Connect, Create File, Begin Loop, File Write, End Loop, File Close, Tree Disconnect, and Logoff. The 'Negotiate' node is selected. On the right, a detailed configuration pane displays session settings and capabilities. The 'Input' section includes 'Connection Handle' set to 'Default'. The 'Capabilities' section shows a value of '0x00000000 (0)'. The 'Output' section includes 'Status Code', 'Dialect', and 'Server Capabilities' fields.

#	Protocol	Name
1	SMB2	Open SMB2 TCP Connection
2	SMB2	Negotiate
3	SMB2	Session Setup
4	SMB2	Tree Connect
5	SMB2	Create File
6	SWT	Begin Loop
7	SMB2	File Write
8	SWT	End Loop
9	SMB2	File Close
10	SMB2	Tree Disconnect
11	SMB2	Logoff

Name	Value
Global SMB2/3 Session Settings	
Honor Credits	True
Credits Pool Size	31
Maximum NetBIOS Size	Unlimited
Input	
Connection Handle	Default
SMB2 Header	
Dialects	
SMB 2.0.2	True
SMB 2.1	False
SMB 3.0	False
SMB 3.0.2	False
SMB 3.1.1	False
Packet Signing	Disabled
Capabilities	0x00000000 (0)
Client GUID	Automatic
Output	
Status Code	
Dialect	
Server Capabilities	

Negotiate Action Capabilities Caveats

- The Load DynamiX Client sends all 6 bits == 1 if all values are set to True (although Wireshark and Microsoft NetMon only show the DFS, Leases and Large_MTU flag values) because the other features are not known by the current versions of Wireshark and NetMon.
- The Load DynamiX Client does not do anything with the information that the server returns in the Negotiate response Capabilities field so it is up to the Tester to know what Capabilities the server supports. A server that supports the SMB2.1 protocol may support more of these features than a server that only supports the SMB 2.002 protocol.
- Large_MTU must be set to **True** if a Client wants to send or request chunks larger than 64kb.

SMB2 Negotiate Maximum Request Size

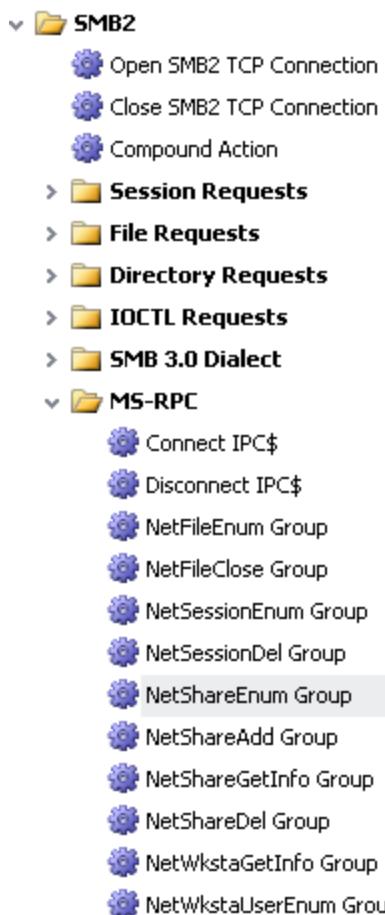
SMB2 **Negotiate** Action has an input named Maximum Request Size. The field is preloaded with values of:

- Unlimited
- Auto
- 4096 (0x1104)
- 16384 (0x4104)
- 65535 (0xFFFF)
- 131071 (0x1FFFF)

This field is used to control the Maximum amount of data that can be requested in any one SMB2 Action. Unlimited (default) indicates that maximum 4 byte value (4 Billion bytes) is requested. Auto indicates that the maximum needed is calculated and supplied. The rest are integer specifications of the Maximum Request Size.

SMB2 MSRPC Actions

The Load DynamiX TDE and Appliance support a limited set of MSRPC/SMB2 commands. The MSRPC protocol is a remote procedure call protocol which largely provides administrative interfaces to Servers and Workstations via the SMB2 protocol. The MSRPC protocol is complex and somewhat obscure so Load DynamiX has taken the approach of building Group Actions (think of them as command macros) that combine several MSRPC commands into a single Load DynamiX Action. Only the Group Actions are usable in Load DynamiX Scenarios. The Toolbox entries for MSRPC/SMB2 Actions looks like



The MSRPC/SMB2 Actions must be used in an SMB2 Scenario and the Scenario must authenticate itself as Administrator to the target SMB2 device.

The screenshot shows the scenario editor with a sequence of actions:

#	Protocol	Name
1	SWT	Create Variable
2	SWT	Create Variable
3	#	.4B Group
4	SMB2	Open SMB2 TCP Connection
5	SMB2	Negotiate
6	SMB2	Session Setup
7	MSRPC/SMB2	Connect IPC\$
8	MSRPC/SMB2	NetShareGetInfo Group
9	MSRPC/SMB2	Disconnect IPC\$
10	SMB2	Logoff
11	SMB2	Close SMB2 TCP Connection
12	SWT	Delay Execution

The 'Session Setup' action is selected. The properties panel on the right shows:

- Input**
 - Connection Handle: Default
 - Credits Charged: 0
 - Credits Requested: 1
- Authentication**
 - GSSAPI: Disable
 - Authentication method: NTLM only
- NTLM authentication options**
 - Domain Name: WORKGROUP
 - Machine Name:
 - User Name: **Administrator**
 - Password: **adminpassword**

The first MSRPC/SMB2 Group Action to be executed after the Session Setup is the **Connect IPC\$** Action. This Action connects the Scenario to the MSRPC services on the SMB2 server that the Scenario is authenticated with.

#	Protocol	Name
1	SWT	Create Variable
2	SWT	Create Variable
3	#	.48 Group
4	SMB2	Open SMB2 TCP Connection
5	SMB2	Negotiate
6	SMB2	Session Setup
7	MSRPC/SMB2	Connect IPC\$
8	MSRPC/SMB2	NetShareGetInfo Group
9	MSRPC/SMB2	NetWkstaGetInfo Group
10	MSRPC/SMB2	NetWkstaUserEnum Group
11	MSRPC/SMB2	NetFileEnum Group

Name	Value
Input	
Session Handle	Default
Credits Charged	0
Credits Requested	1
Share Name	
Input Format	Auto
Output	
Output Handle	7: SMB2ShareHandle
Response Handlers	
Completion Status	
Scenario Impact	

Following the Connect IPC\$ Action, the Scenario may issue any of the se MSRPC Actions:

- **NetShareGetInfo**: Returns various information about that Share from the Server

#	Protocol	Name
1	SWT	Create Variable
2	SWT	Create Variable
3	#	.48 Group
4	SMB2	Open SMB2 TCP Connection
5	SMB2	Negotiate
6	SMB2	Session Setup
7	MSRPC/SMB2	Connect IPC\$
8	MSRPC/SMB2	NetShareGetInfo Group

Name	Value
Input	
Share Handle	Default
Credits Charged	0
Credits Requested	1
Server	=@UP(1, F)
Share	DUMP
Level	0

- **NetWkstaGetInfo**: Returns various pieces of information about that computer (e.g. Windows version)

#	Protocol	Name
1	SWT	Create Variable
2	SWT	Create Variable
3	#	.48 Group
4	SMB2	Open SMB2 TCP Connection
5	SMB2	Negotiate
6	SMB2	Session Setup
7	MSRPC/SMB2	Connect IPC\$
8	MSRPC/SMB2	NetShareGetInfo Group
9	MSRPC/SMB2	NetWkstaGetInfo Group

Name	Value
Input	
Share Handle	Default
Credits Charged	0
Credits Requested	1
Server	=@UP(1, F)
Level	101

- **NetWkstaUserEnum**: Enumerate a list of users logged into that Windows computer.

The screenshot shows a tool interface with two main panes. The left pane is a table with columns: #, Protocol, and Name. It lists 11 operations: Create Variable (x2), .48 Group (highlighted in green), Open SMB2 TCP Connection, Negotiate, Session Setup, Connect IPC\$, NetShareGetInfo Group, NetWkstaGetInfo Group, and NetWkstaUserEnum Group (highlighted in blue). The right pane shows configuration parameters for the selected operation (.48 Group). The 'Input' section includes fields for Share Handle (Default), Credits Charged (0), Credits Requested (1), Server (= @UP(1, F)), and Level (0). There is also an 'Output' section with an Output Handle field set to 11: MSRPCFileIDHandle.

- **NetFileEnum Group**: Enumerate a list of open files for the named user at the specified path. Returns a handle that can be used by **NetFileClose** to close open files.

The screenshot shows a tool interface with two main panes. The left pane is a table with columns: #, Protocol, and Name. It lists 11 operations: Create Variable (x2), .48 Group (highlighted in green), Open SMB2 TCP Connection, Negotiate, Session Setup, Connect IPC\$, NetShareGetInfo Group, NetWkstaGetInfo Group, NetWkstaUserEnum Group, and NetFileEnum Group (highlighted in blue). The right pane shows configuration parameters for the selected operation (NetFileEnum Group). The 'Input' section includes fields for Share Handle (Default), Credits Charged (0), Credits Requested (1), Server (= @UP(0, A)), Path (\\"), User (USER001), and Level (3). The 'Output' section includes an Output Handle field set to 11: MSRPCFileIDHandle.

- **NetFileClose Group**: Close selected files (FileIDIndex) from list of open files returned in FileID Handle from **NetFileEnum Group** Action.

The screenshot shows a tool interface with two main panes. The left pane is a table with columns: #, Protocol, and Name. It lists 12 operations: Create Variable (x2), .48 Group (highlighted in green), Open SMB2 TCP Connection, Negotiate, Session Setup, Connect IPC\$, NetShareGetInfo Group, NetWkstaGetInfo Group, NetWkstaUserEnum Group, NetFileEnum Group (highlighted in blue), and NetFileClose Group (highlighted in blue). The right pane shows configuration parameters for the selected operation (NetFileClose Group). The 'Input' section includes fields for Share Handle (Default), Credits Charged (0), Credits Requested (1), Server (= @UP(0, A)), FileID Handle (Default), and FileIDIndex (= @UP(0, B)).

- **NetSessionEnum Group:** Enumerate open Sessions on the Server that the Client is connected to.

#	Protocol	Name
1	SWT	Create Variable
2	SWT	Create Variable
3	#	.4B Group
4	SMB2	Open SMB2 TCP Connection
5	SMB2	Negotiate
6	SMB2	Session Setup
7	MSRPC/SMB2	Connect IPC\$
8	MSRPC/SMB2	NetShareGetInfo Group
9	MSRPC/SMB2	NetWkstaGetInfo Group
10	MSRPC/SMB2	NetWkstaUserEnum Group
11	MSRPC/SMB2	NetFileEnum Group
12	MSRPC/SMB2	NetFileClose Group
13	MSRPC/SMB2	NetSessionEnum Group

Name	Value
Input	
Share Handle	Default
Credits Charged	0
Credits Requested	1
Server	172.16.1.44
Client	172.17.240.1
User	Administrator
Level	0

- **NetSessionDel Group:** Close (Delete) open Sessions on the Server that the Client is connected to.

#	Protocol	Name
1	SWT	Create Variable
2	SWT	Create Variable
3	#	.4B Group
4	SMB2	Open SMB2 TCP Connection
5	SMB2	Negotiate
6	SMB2	Session Setup
7	MSRPC/SMB2	Connect IPC\$
8	MSRPC/SMB2	NetShareGetInfo Group
9	MSRPC/SMB2	NetWkstaGetInfo Group
10	MSRPC/SMB2	NetWkstaUserEnum Group
11	MSRPC/SMB2	NetFileEnum Group
12	MSRPC/SMB2	NetFileClose Group
13	MSRPC/SMB2	NetSessionEnum Group
14	MSRPC/SMB2	NetSessionDel Group

Name	Value
Input	
Share Handle	Default
Credits Charged	0
Credits Requested	1
Server	172.16.1.44
Client	172.17.240.1
User	Administrator

- **NetShareEnum Group:** Enumerate Shares on the Server that the Client is connected to.

#	Protocol	Name
1	SWT	Create Variable
2	SWT	Create Variable
3	#	.4B Group
4	SMB2	Open SMB2 TCP Connection
5	SMB2	Negotiate
6	SMB2	Session Setup
7	MSRPC/SMB2	Connect IPC\$
8	MSRPC/SMB2	NetShareGetInfo Group
9	MSRPC/SMB2	NetWkstaGetInfo Group
10	MSRPC/SMB2	NetWkstaUserEnum Group
11	MSRPC/SMB2	NetFileEnum Group
12	MSRPC/SMB2	NetFileClose Group
13	MSRPC/SMB2	NetSessionEnum Group
14	MSRPC/SMB2	NetSessionDel Group
15	MSRPC/SMB2	NetShareEnum Group

Name	Value
Input	
Share Handle	Default
Credits Charged	0
Credits Requested	1
Server	172.16.1.44
Level	0

- **NetShareDel Group:** Delete the named Share from the Server.

#	Protocol	Name
1	SWT	Create Variable
2	SWT	Create Variable
3	#	.4B Group
4	SMB2	Open SMB2 TCP Connection
5	SMB2	Negotiate
6	SMB2	Session Setup
7	MSRPC/SMB2	Connect IPC\$
8	MSRPC/SMB2	NetShareGetInfo Group
9	MSRPC/SMB2	NetWkstaGetInfo Group
10	MSRPC/SMB2	NetWkstaUserEnum Group
11	MSRPC/SMB2	NetFileEnum Group
12	MSRPC/SMB2	NetFileClose Group
13	MSRPC/SMB2	NetSessionEnum Group
14	MSRPC/SMB2	NetSessionDel Group
15	MSRPC/SMB2	NetShareEnum Group
16	MSRPC/SMB2	NetShareDel Group

- **NetShareAdd Group:** Add the named Share to the Server.

#	Protocol	Name
1	SWT	Create Variable
2	SWT	Create Variable
3	#	.4B Group
4	SMB2	Open SMB2 TCP Connection
5	SMB2	Negotiate
6	SMB2	Session Setup
7	MSRPC/SMB2	Connect IPC\$
8	MSRPC/SMB2	NetShareGetInfo Group
9	MSRPC/SMB2	NetWkstaGetInfo Group
10	MSRPC/SMB2	NetWkstaUserEnum Group
11	MSRPC/SMB2	NetFileEnum Group
12	MSRPC/SMB2	NetFileClose Group
13	MSRPC/SMB2	NetSessionEnum Group
14	MSRPC/SMB2	NetSessionDel Group
15	MSRPC/SMB2	NetShareEnum Group
16	MSRPC/SMB2	NetShareDel Group
17	MSRPC/SMB2	NetShareAdd Group

The final Group Action to execute is the MSRPC/SMB2 **Disconnect IPC\$** Action which disconnects the Scenario from the MSRPC services on the target SMB2 device.

The screenshot shows the Load DynamiX configuration interface. On the left, a list of actions is displayed in a table:

#	Protocol	Name
1	SWT	Create Variable
2	SWT	Create Variable
3	#	.4B Group
4	SMB2	Open SMB2 TCP Connection
5	SMB2	Negotiate
6	SMB2	Session Setup
7	MSRPC/SMB2	Connect IPC\$
8	MSRPC/SMB2	NetShareGetInfo Group
9	MSRPC/SMB2	NetWkstaGetInfo Group
10	MSRPC/SMB2	NetWkstaUserEnum Group
11	MSRPC/SMB2	NetFileEnum Group
12	MSRPC/SMB2	NetFileClose Group
13	MSRPC/SMB2	NetSessionEnum Group
14	MSRPC/SMB2	NetSessionDel Group
15	MSRPC/SMB2	NetShareEnum Group
16	MSRPC/SMB2	NetShareDel Group
17	MSRPC/SMB2	NetShareAdd Group
18	MSRPC/SMB2	Disconnect IPC\$

On the right, the properties for the selected action (Disconnect IPC\$) are shown in a detailed view:

Name	Value
Input	
Share Handle	Default
Credits Charged	0
Credits Requested	1
Response Handlers	
Completion Status	
Scenario Impact	

Each of these Group Actions is delivered to the Load DynamiX Appliance as a set of MSRPC Actions that are executed sequentially to accomplish the desired effect.

MSRPC/SMB2 Interoperability

- Server and Share input fields accept all Functions including @Variable, @UP(), ...
- MSRPC/SMB2 Group Actions may be interspersed with other SMB2 Actions in the same Scenario.

MSRPC/SMB2 Caveats

- No more than one set of MSRPC/SMB2 Group Actions can be executing at any one time. Otherwise, some sets of the MSRPC/SMB2 Actions will fail.
- Windows MSRPC services seem to ignore the Server and Share input fields of the **NetShareGetInfo** Group Action and the Server input field of the **NetWkstaGetInfo** and **NetWkstaUserEnum** Group Actions.

SMB2 Sample Projects

SMB2 Sequential Write Read

Client: SMB2 Payload Rate.client_scenario

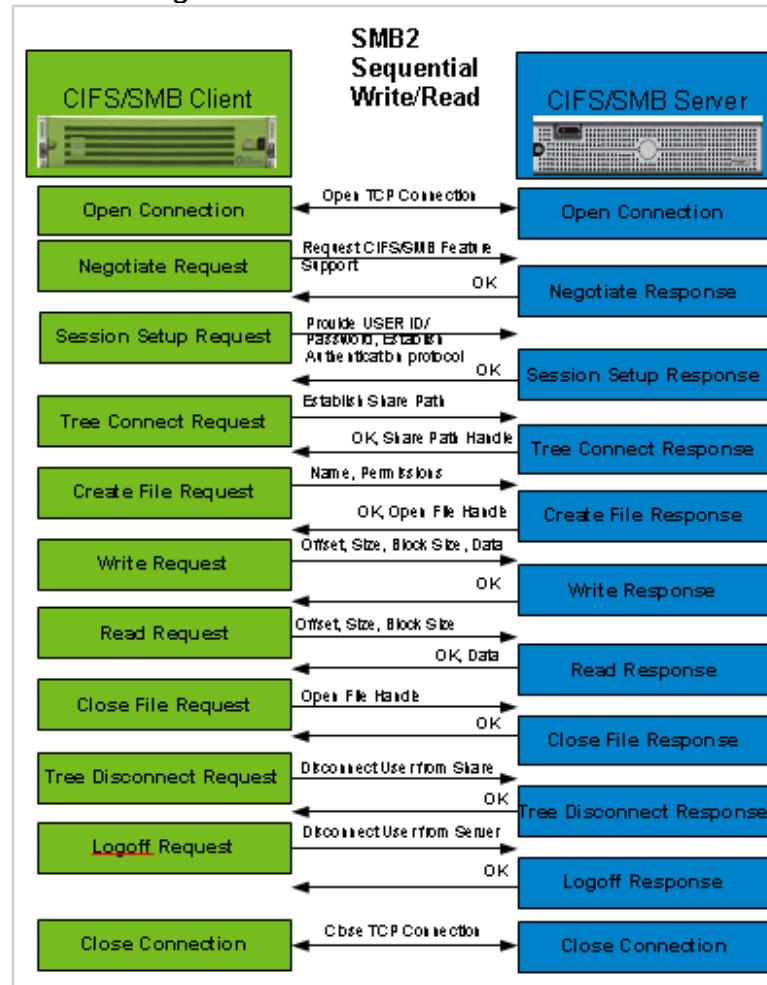
#	Protocol	Name
1	SMB2	Open SMB2 TCP Connection
2	SMB2	Negotiate
3	SMB2	Session Setup
4	SMB2	Tree Connect
5	SMB2	Create File
6	SWT	Begin Loop
7	SMB2	File Write
8	SWT	End Loop
9	SWT	Begin Loop
10	SMB2	File Read
11	SWT	End Loop
12	SMB2	File Close
13	SMB2	Tree Disconnect
14	SMB2	Logoff

SMB2 Multi-Credit Write Read

#	Protocol	Name
1	SMB2	Open SMB2 TCP Connection
2	SMB2	Negotiate
3	#	Ask server to grant 9 credits for use in writes and reads
4	SMB2	Session Setup
5	SMB2	Tree Connect
6	SMB2	Create File
7	SWT	Begin Loop
8	#	Write 100K bytes, Credits Changed = 1 + (10000 - 1)/65536 = 2
9	SMB2	File Write
10	#	Read 100K bytes, Credits Changed = 1 + (10000 - 1)/65536 = 2
11	SMB2	File Read
12	#	Write 32000 bytes, Credits Changed = 1 + (10000 - 1)/65536 = 1
13	SMB2	File Write
14	#	Read 32000 bytes, Credits Changed = 1 + (10000 - 1)/65536 = 1
15	SMB2	File Read
16	SWT	End Loop
17	SMB2	File Close
18	SMB2	Tree Disconnect
19	SMB2	Logoff

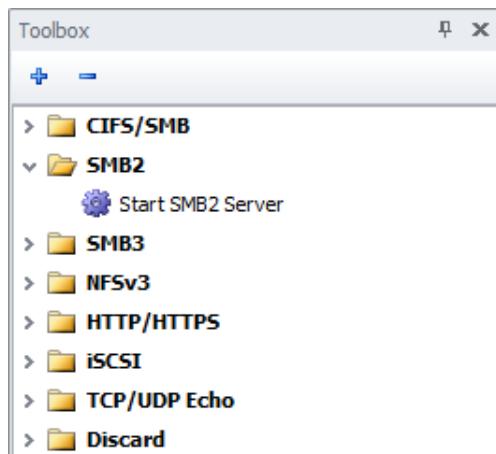
SMB2 Project Flow

The following is the SMB2 client/server interaction flow for a simple sequential write/read Project.



SMB2 Start Server Action

The Load DynamiX Appliance firmware supports a SMB2 server emulation. The SMB2 server is instantiated in a Server Scenario using the Start SMB2 Server Action and providing, at a minimum, an IP address for the Server.



SMB2 Statistics

See the [Advanced Concepts: Test Execution Rules](#) for a complete list of the statistics generated when SMB2 Actions are used in a Scenario.

Reference: SMB 3.0 Commands and Behaviors

Reference: Load DynamiX SMB 3.0 Commands and Behaviors

A link to the SMB2.0/3.0 protocol reference material is provided in the [References and Terminology section](#).

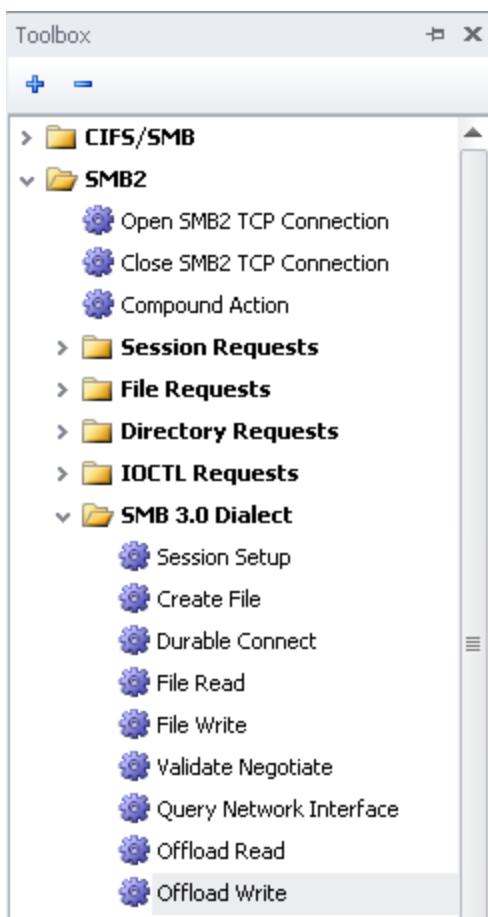
SMB3.0 Dialect Support

The SMB3.0 Dialect is a set of extensions to the SMB2 protocol. The currently supported SMB3.0 extensions are:

- Multi-Channel communications in which Read and Write operations to a single file can be spread out over several channels (really several TCP connections) to the Server on which the file is being Read or Written. Multi-Channel support is provided in three Actions: Session Setup, File Read and File Write. Currently the Tester must know which interfaces are available from a Server and hard code them into the Open SMB2 TCP Connection Action to create the Multi-Channel test. The user can alternatively link interfaces/IP addresses returned by the Query Network Interface Action to a handle and index available in the Open SMB2 TCP Connection Action. See below for how to use the Query Network Interface Action results to access the IP addresses of the interfaces that the SMB3 server returns.
- A new Signing algorithm referred to as SMB3 Signing.
- Several new IOCTL-type commands:
 - One to validate that SMB3.0 support has been negotiated (Validate Negotiate)
 - One to query the Server's available Channels (Query Network Interface).
 - One to initiate the Copy Offload Read process (Offload Read)
 - One to initiate the Copy Offload Write process (Offload Write)
- Persistent Handles v2 support in the SMB2/3.0 Create File Action.
- Directory Leasing (Lease v2 context) support in the SMB2/3.0 Create File Action.
- Application Instance ID support in the SMB2/3.0 Create File Action.

SMB3.0 Dialect Toolbox

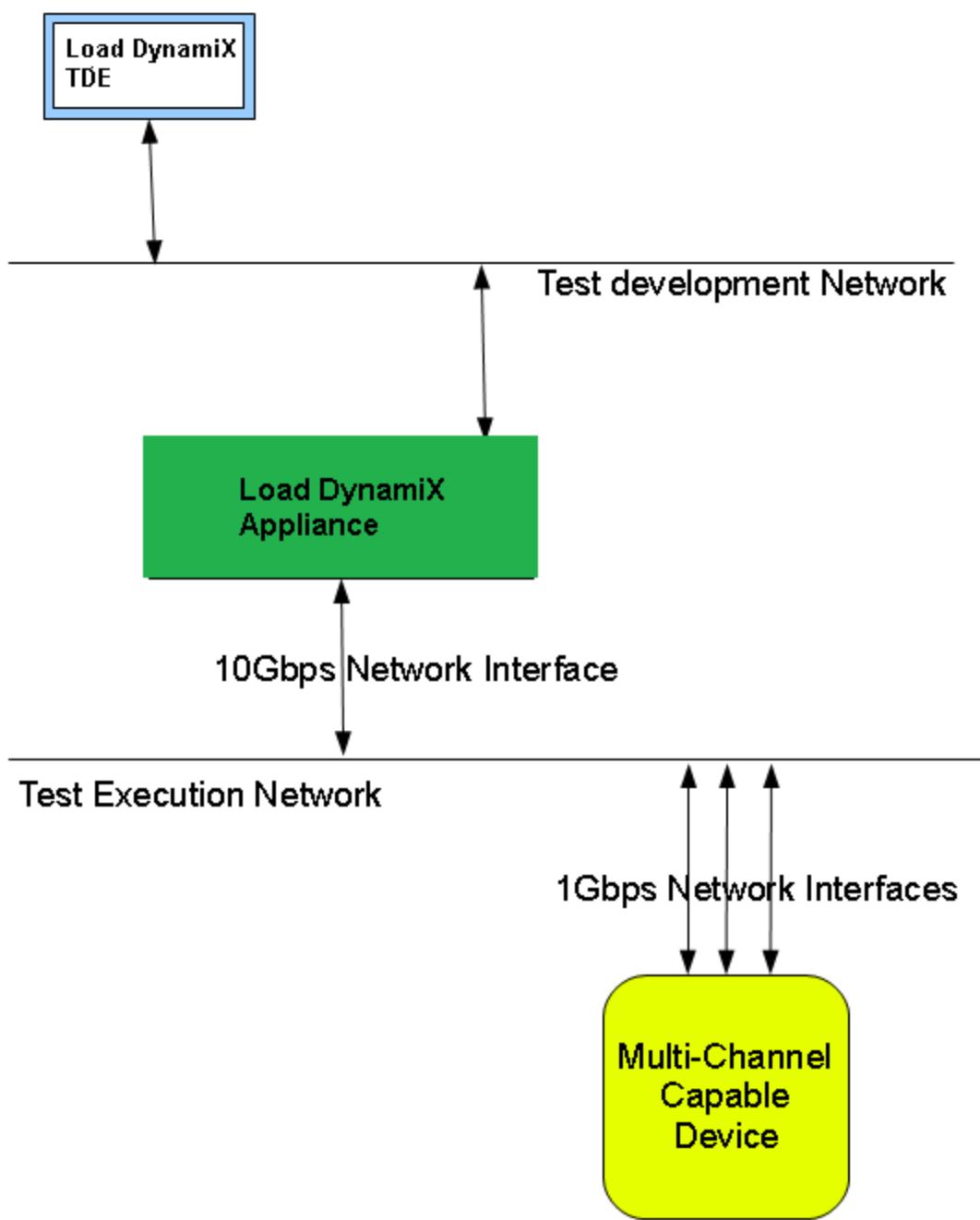
The SMB3.0 Dialect Actions are in a sub-folder in the SMB2 Toolbox:



Multi-Channel

Multi-Channel communications between Client and Server is performance enhancing. It allows a Server (DUT) to use more than one channel (IP Address or Network Interface) to exchange (read or write) data with a Client. This feature allows a Client with a high performance Network Interface (e.g. the Load DynamiX Appliance 10Gbps ports) to communicate with a Server using several 1Gbps Network Interfaces with potentially higher performance than if a single 1Gbps interface was used.

Conceptually the environment would look like



The 10Gbps Network interface on the Load DynamiX Appliance could read or write data to the Multi-Channel Capable Device over 10 1Gbps Network Interfaces if that is what the device supports. The Multi-Channel process requires that the Client extract from the target device the channels it has available and choose to use them if the workload demands higher throughput.

The Load DynamiX support for Multi-Channel provides the IOCTL Action (**Query Network Interface**) to extract the necessary interface information but does not use it directly. The Tester must know in advance what Network Interfaces a device has available (or have gotten that info from the response to the **Query Network Interface** Action using NetMon v3.4 to parse it in a PCAP file) and make use of that information to create the Scenario(s) that will use those interfaces. See below for how to use the **Query Network Interface** Action results to access the IP addresses of the interfaces that the SMB3 server returns.

A example of a Multi-Channel Scenario is:

The first Negotiate Action requests SMB 2.1, SMB 3.0 and/or SMB 3.1.1 support. Then comes the usual Session Setup and Tree Connect to establish the credentials and share. Validate Negotiate requests information from the Server regarding supported protocols and capabilities but this information is not used by the Load DynamiX Scenario. Query Network Interface requests a list of available interfaces but this information is not used directly by the Load DynamiX Scenario, the Tester must know this information up front when the Scenario is created. See below for how to use the Query Network Interface Action results to access the IP addresses of the interfaces that the SMB3 server returns.

The second **Open SMB2 TCP Connection**, **Negotiate** and **Session Setup** create the second channel. At this point, the Scenario has created two channels with which to Read or Write data to the Server. The **SMB2 File Write** Action will write data to the first (or initial) channel and the **SMB2/3.0 File Write** and **File Read** Actions will Write or Read data to the Server over the second channel.

If the Server had 10 network interfaces available, the Tester could create a Scenario that opened 9 additional channels to the Server and could move data across those 9 channels simultaneously using the Load DynamiX Thread Blocks capability.

SMB3.0 Session Setup, Create File, File Read, File Write Actions

Four SMB2 Actions have been updated to take advantage of SMB3.0 Dialect capabilities:

- **Create File**: Adds new Durable Connection input fields, new Leasing input fields.
- **Session Setup**: Adds new Session Flags/Session Handle input
- **File Read**: Adds Session Handle to be able to select the second or Nth session of a Multi-Channel test.
- **File Write**: Adds Session Handle to be able to select the second or Nth session of a Multi-Channel test.

SMB3.0 Signing

SMB3.0 Signing is a new signing algorithm. In Multi-Channel environments, the second Session Setup in a Multi-Channel Scenario (see example above) MUST be signed (even if signing is not enabled in the first Negotiate Action) and it Must be signed with the SMB3.0 Signing algorithm. If (in the example above), the first Negotiate Action had Signing enabled and the SMB3.0 Dialect was negotiated with the Server, then all signing would be done with the SMB3.0 Signing algorithm. The SMB3 Client supports SMB3 PACKET SIGNING using the AES-HMAC-128 algorithm. The SIGNING Algorithm that will be used between the client and SMB3 Server when Packet Signing is Enabled is the same for SMB2 and SMB3 dialects.

IOCTLs

Validate Negotiate: This Action requests the Server to provide information regarding which SMB2 Protocols and Dialects are supported (2.002, 2.1, 3.0, 3.1.1) and which Capabilities are supported (Distributed File System, Leasing, Large MTU (Multi-Credit), Multi-Channel, Persistent Handles, Directory Leasing, Encryption). The Load DynamiX Client ignores the information returned by this Action.

Query Network Interface: This Action requests that the target server return a list of its available network interfaces (channels). The Load DynamiX Client allows the Tester to iterate through the interface information (IP address, IP address type) received when Query Network Interface is executed. A server will return a record containing four kinds of information relating to its interfaces when this command is executed:

Interface #	Interface IP Address	Interface IP Address Type	Interface Speed
-------------	----------------------	---------------------------	-----------------

Interface # - assigned by the server

Interface IP Address - an IPv4 or IPv6 address

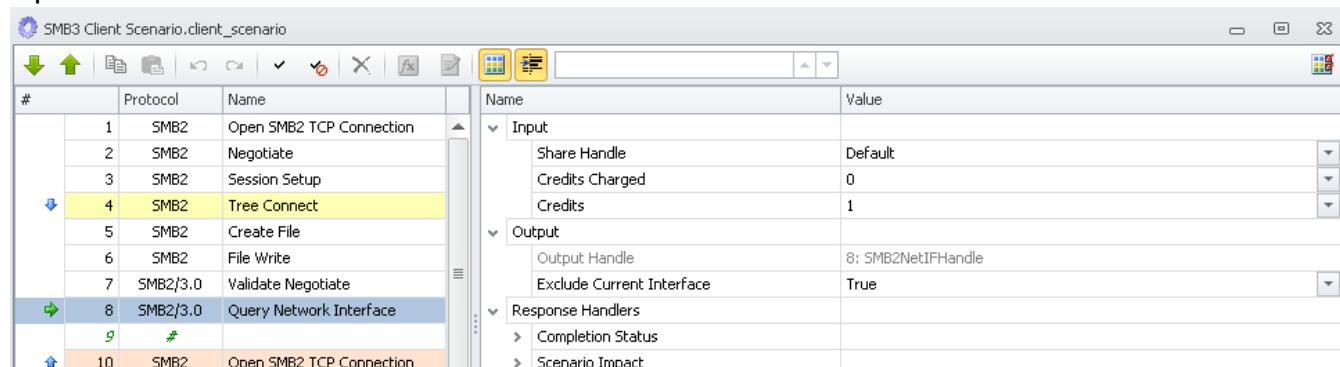
Interface IP Address Type - v4 or v6

Interface Speed - 1G, 10G, (recorded not used by Load DynamiX)

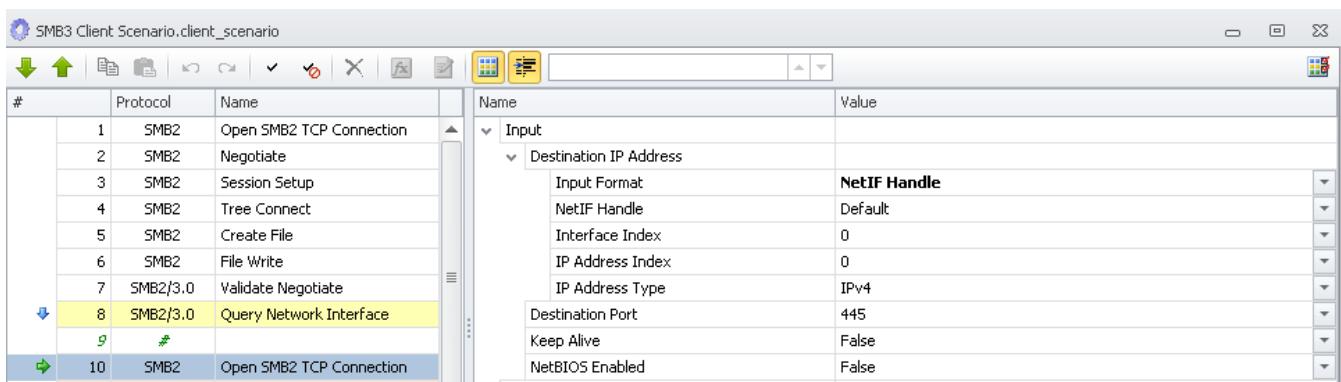
As these records are received, the Load DynamiX Client builds a 4 x N table of interface information (example of a single server with 4 interfaces with IPv4, IPv6 or both addresses):

Client Index	Interface #	Interface IP Address	Interface IP Address Type	Interface Speed
0	33	10.0.0.49	FD00::172:16:1:55	1G
1	39		FD00::172:16:1:60 FD00::172:55:100:30	10G
2	71	192.168.1.23		1G
3	14	172.17.100.10 172.17.150.0	FD00::172:16:1:99	

Clients may access this information using the SMB2/3.0 Query Network Interface and SMB2 Open TCP Connection Actions.



The **Query Network Interface** command extracts the interface information and makes it available to the **SMB2 Open TCP Connection** Action through the NetIF handle. If the NetIF handle is selected as the input format, then Testers may select the Interface index, IP Address index and the IP Address Type.



At run time, the selected fields are extracted from the table of Interface information and passed to the SMB2 Open TCP Connection Action.

Copy Offload - it is possible on some Windows servers to have cooperating servers offload operations like file copy from the Client. The Copy Offload feature allows a Client to initiate a file copy that is performed by the cooperating servers. On the Load DynamiX product, the Copy Offload feature is implemented by two Actions: Offload Read and Offload Write.

Offload Read: This is the Action that identifies the source connection and file to be copied.

Offload Write: This is the Action that identifies the target connection and file for the copy.

In the example Scenario below, an **Copy Offload** is done within a single server. A connection to a server is opened in line 1 and a file from that server is created in line 10. This is the source of the copy and the handle for this file is passed as input the Offload Read. The **Offload Write Action** takes the token from the **Offload Read Action** and the file handle from the **Create File Action** in line 22, and performs the Copy. Then the target file is then read to verify it contains the data that was expected (Seeded Random(0) data). Finally the Source and Destination files are closed and the server connection closed.

smb3 offload write.client_scenario*

#	Protocol	Name
1	SMB2	Open SMB2 TCP Connection
2	SMB2	Negotiate
3	SMB2	Session Setup
4	SMB2	Tree Connect
5	#	file size stored in this variable
6	SWT	Create Variable
7	#	create source file
8	SMB2	Create File
9	#	write file, filesize = var(8), known data
10	SMB2	File Write
11	#	obtain token for future offload write
12	SMB2/3.0	Offload Read
13	#	create target file
14	SMB2/3.0	Create File
15	#	set file pointer to end of file
16	SMB2	Set Info
17	#	execute offload write
18	SMB2/3.0	Offload Write
19	#	validate written data
20	SMB2	File Read
21	SMB2	File Close
22	SMB2	File Close
23	SMB2	Tree Disconnect

smb3 offload write.client_scenario*

#	Protocol	Name	
1	SMB2	Open SMB2 TCP Connection	
2	SMB2	Negotiate	
3	SMB2	Session Setup	
4	SMB2	Tree Connect	
5	#	file size stored in this variable	
6	SWT	Create Variable	
7	#	create source file	
8	SMB2	Create File	
9	#	write file, filesize = var(8), known data	
10	SMB2	File Write	
11	#	obtain token for future offload write	
12	SMB2/3.0	Offload Read	
13	#	create target file	
14	SMB2/3.0	Create File	
15	#	set file pointer to end of file	
16	SMB2	Set Info	
17	#	execute offload write	
18	SMB2/3.0	Offload Write	

Name	Value
Input	
Offload Token Handle	12: SMB2OffloadTokenHandle
Target File Handle	14: SMB2FileHandle
Credits Charged	0
Credits	1
File Offset	0
Transfer Offset	0
Copy Length	1,024

Name	Value
Response Handlers	
Completion Status	
Scenario Impact	

Persistent Handles v2

V2 of the Durable Handle capability supports new input fields including a Durable Timeout (how long will the Durable Connection be possible) and Durable Flags in which the Persistent

characteristic of the handle is defined (True/False). False is the default value which means that Durable Handle is NOT persistent.

Directory Leasing v2

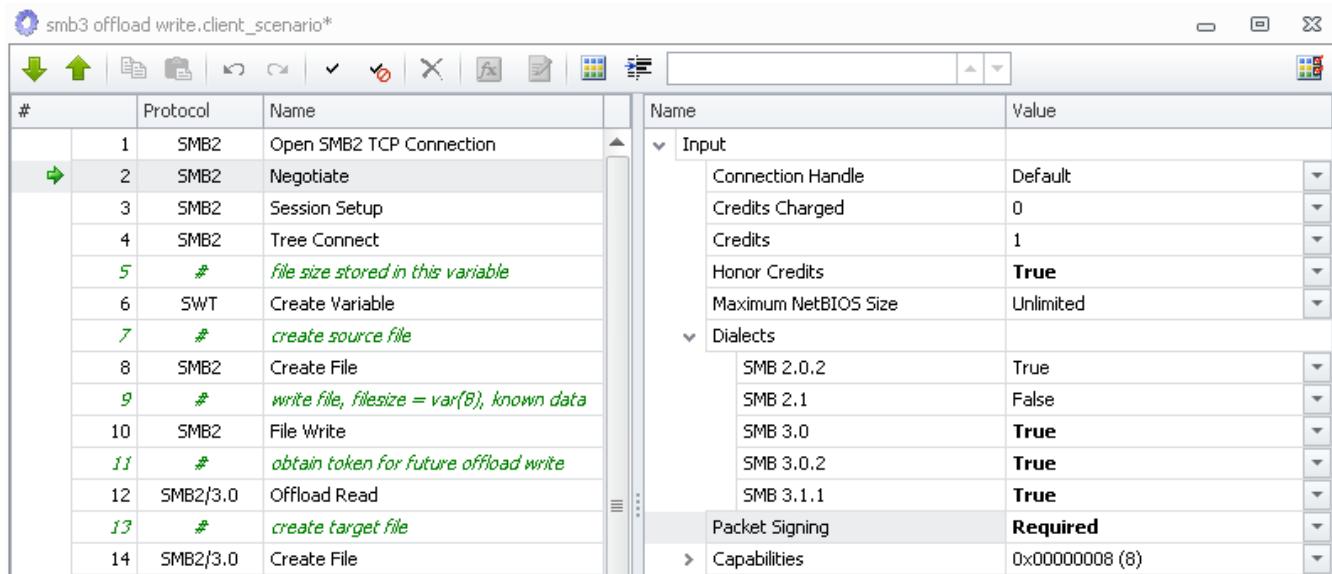
V2 of the Lease capability supports Leasing of Directories. Additional input fields are present in the SMB2/3.0 Create File Action (Lease Flags:Parent Lease key and Lease Version:V1 or V2).

Application Instance ID

An Input field in the SMB 2/3.0 **Create File** Action [True/False] - sends a GUID value for Application Instance ID if set to True. Sends no value if False. GUID is generated so that it will be unique in each instance of the scenario in which it is used.

SMB3 Client

The only change/addition to the SMB2 Actions is the addition of 3.1.1 Dialect support in the SMB2 **Negotiate** Action. All Client SMB3.1.1 behaviors are triggered by the presence of the 3.1.1 dialect set to True in the **Negotiate** Action.



The SMB3 Client software supports the following 3.1.1 features:

Pre-Authentication Integrity

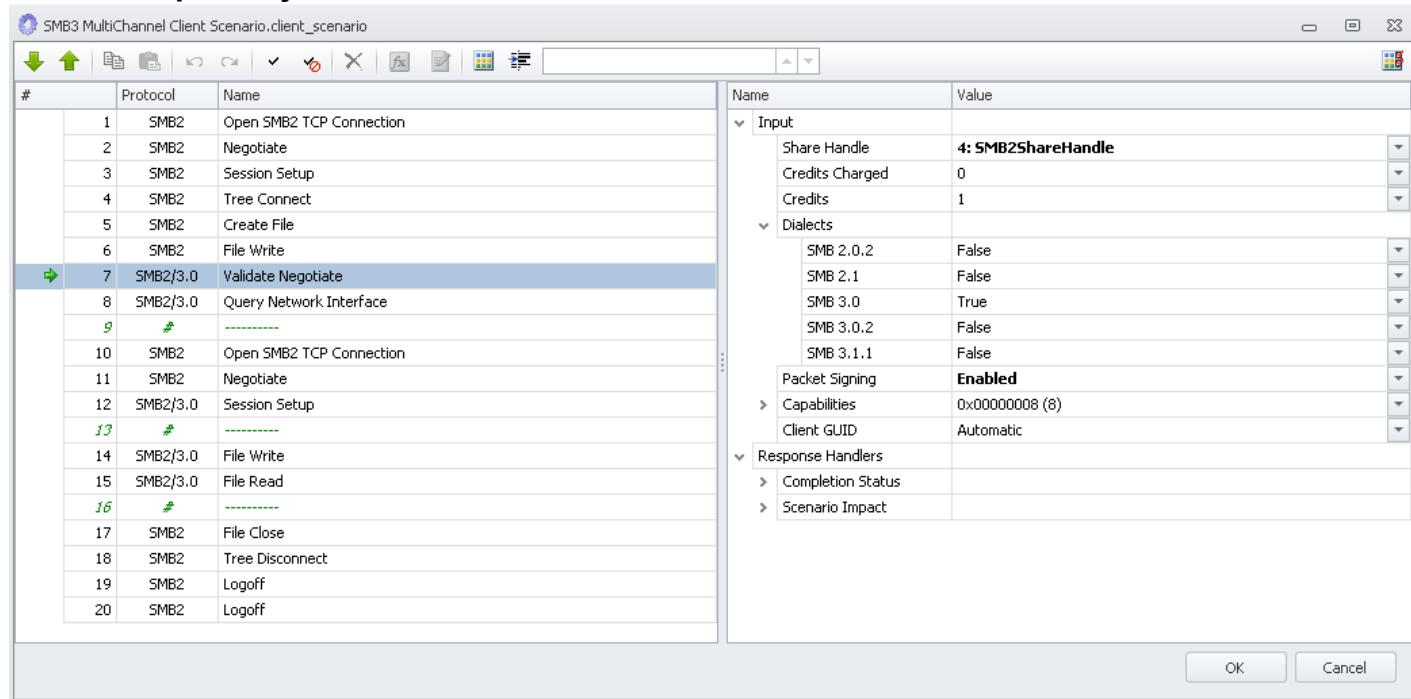
Pre-authentication integrity provides improved protection from a man-in-the-middle attacker tampering with SMB's connection establishment and authentication messages. Pre-Auth integrity verifies all the "negotiate" and "session setup" exchanges used by SMB with a strong cryptographic hash (SHA-512). Using SMB signing on top of an SMB 3.1.1 session protects you from an attacker tampering with any packets. Using SMB encryption on top of an SMB 3.1.1 session protects you from an attacker tampering with or eavesdropping on any packets.

SMB Encryption Improvements

SMB Encryption, introduced with SMB 3.0, used a fixed cryptographic algorithm: AES-128-CCM. SMB 3.1.1 offers a mechanism to negotiate the crypto algorithm per connection, with options for AES-128-CCM and AES-128-GCM. The Load DynamiX

SMB3 Client offers support for both encryption algorithms and allows the Server to specify which of the two algorithms to use. The Load Dynamix **Start SMB3 Server** Action specifies if SMB3 Encryption is to be used and whether it is in Shared or Session mode.

SMB3 Sample Project

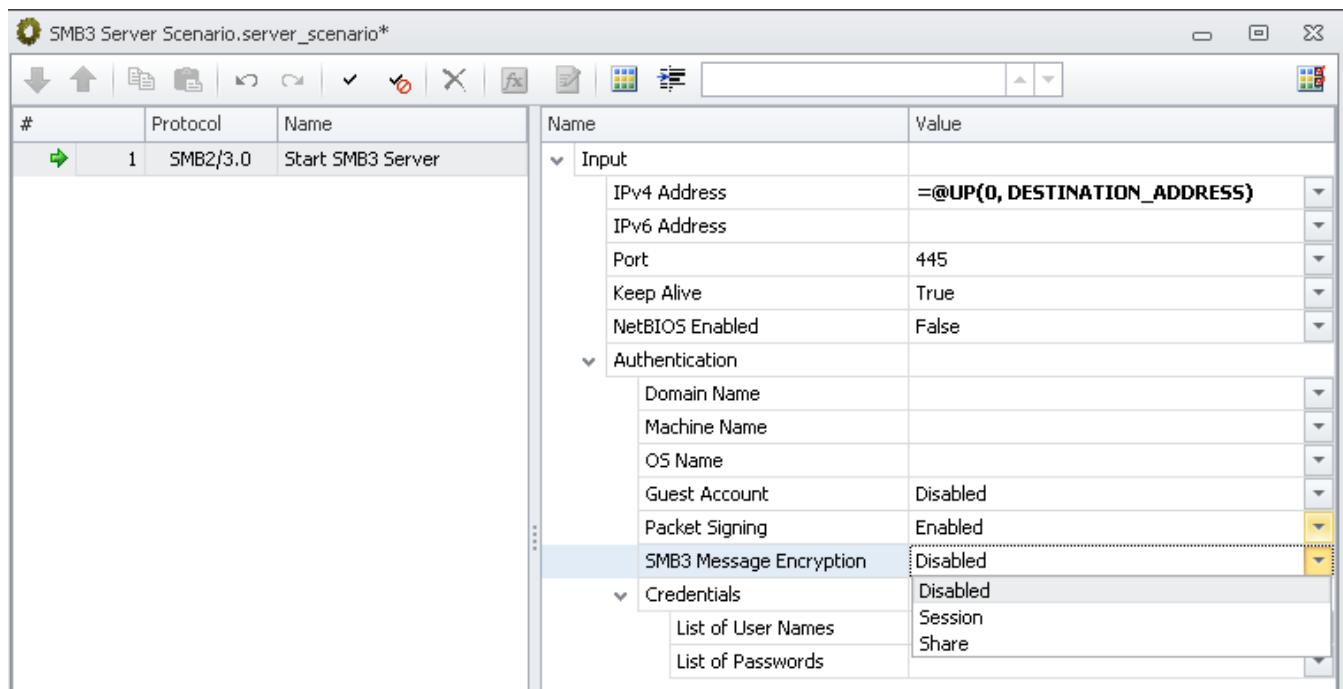


SMB3 Server

The Load Dynamix Firmware provides minimal SMB3 server support via the **Start SMB3 Server** Action. The SMB3 Server supports a limited set of the SMB3 Client Actions and features supported by Load Dynamix. The SMB3 Server also supports SMB3 PACKET SIGNING using the AES-HMAC-128 algorithm. The SMB3 Server supports the following SMB2 Dialects: 2.02, 2.1, 3.0 and 3.1.1. The SIGNING Algorithm that will be used between the client and SMB3 Server when Packet Signing is Enabled is the same for SMB2 and SMB3 dialects. The Load Dynamix **Start SMB3 Server** Action specifies if SMB3 Encryption is to be used and whether it is in Shared (send encryption required bit in the Tree Connect request) or Session mode (send encryption required bit in the **Session Setup** request).

The SMB3 Server's response to the Validate Negotiate Action depends on the SMB2 Dialects that were Negotiated. If SMB2 Dialects have been Negotiated that do not support the Validate Negotiate command then the SMB3 server response is Invalid Parameter.

The SMB3 Server supports all of the SMB2 features that the SMB2 Server supports and a limited subset of the SMB3 Client Actions.



Copyright © 2008-2017 Virtual Instruments Inc.

Reference: NFSv2 Command List**Reference: Load DynamiX NFS version 2 command list**

ACTION
Portmapper
Close Portmapper TCP Connection
Get Program Port
Null
Open Portmapper TCP Connection
Mount
Close Mount TCP Connection
Mount Directory
Null
Open Mount TCP Connection
Unmount Directory
File System
Close TCP Connection
Create Directory
Create File
Create Link
Create Symbolic Link
Get File Attributes
Get Filesystem Attributes
Lookup
Null
Open NFS TCP Connection
Read Directory
Read File
Read Symbolic Link
Remove Directory
Remove File
Rename

Set File Attributes

Write File

A links to NFSv2 protocol reference material is provided in the [References and Terminology section](#).

See the [Advanced Concepts: Test Execution Rules](#) for a complete list of the statistics generated when NFSv2 Actions are used in a Scenario.

Copyright © 2008-2017 Virtual Instruments Inc.

Reference: NFSv3 Commands and Behaviors

Reference: Load DynamiX NFS version 3 Commands and Behaviors

ACTION
Portmapper
Close Portmapper TCP Connection
Get Program Port
Null
Open Portmapper TCP Connection
Mount
Close Mount TCP Connection
Mount Directory
Null
Open Mount TCP Connection
Unmount Directory
File System
Access
Close NFS TCP Connection
Commit File
Create Device
Create Directory
Create File
Create Link
Create RPCSEC GSS Context
Create Symbolic Link
Destroy RPCSEC GSS Context
Get Dynamic File System Info
Get File Attributes
Get Posix Info
Get Static File System Info
Lookup

Null
Open NFS TCP Connection
Read Directory
Read Directory Extended
Read File
Read Symbolic Link
Read Symbolic Link and Lookup
Remove Directory
Remove File
Rename
Set File Attributes
Write File
Server
Start NFSv3 Server

A link to NFSv3 protocol reference material is provided in the [References and Terminology section](#).

NFSv3 Asynchronous Read and Write Operations

NFSv3 **Read File** and **Write File** operations can be segmented into multiple operations that are issued asynchronously. To do this use the Number of Outstanding Requests feature of the NFSv3 **Read File** and **Write File** Actions. Below are two screen shots showing the Read and Write Actions with the Number of Outstanding Requests field. The allowed range of values for the Number of Outstanding Requests field is 1 to 128. Values < 1 or > 128 will result in an error that will prevent saving the Project.

The default value of Number of Outstanding Requests is 1 which results in same behavior as before Number of Outstanding Requests was introduced. The generic behavior of a Read or Write with Number of Outstanding Requests > 1 is that the operation (Read or Write) is broken into up to N independent operations of size Number of Bytes per Request where

$N = (\text{Number of Bytes}/\text{Number of Bytes per Request}) \text{ or Number of Outstanding Requests}$,

whichever is smaller and these N operations are all issued at the same time. As these requests complete, new operations are issued until the total number represented by (Number of Bytes/Number of Bytes per Request) has been completed. Any errors encountered during the processing of these requests will cause the entire Action to fail.

In the **Read File** Action below, Number of Bytes per Request == 1024, Number of Bytes == 10240 so the result of the division is 10. Number of Outstanding Requests is 5 so N == 5 and initially 5 requests will be sent in parallel to the target NFS server and as any one of these 5 requests completes, the remaining requests will be issued.

Client NFSv3 Read.client_scenario*

#	Protocol	Name
1	PMAPv2	Open Portmapper TCP Connection
2	PMAPv2	Null
3	PMAPv2	Get Program Port
4	MNTv3	Open Mount TCP Connection
5	MNTv3	Null
6	MNTv3	Mount Directory
7	PMAPv2	Get Program Port
8	NFSv3	Open NFS TCP Connection
9	PMAPv2	Close Portmapper TCP Connection
10	MNTv3	Close Mount TCP Connection
11	NFSv3	Null
12	NFSv3	Access
13	NFSv3	Create File
14	NFSv3	Lookup
15	SWT	Begin Loop
16	NFSv3	Read File

Similarly, in the **Write File** Action below, the result of the division of Number of Bytes per Request / Number of Bytes == 10 and the Number of Outstanding Requests == 8, so 8 Write requests will be issued initially and then as the 8 complete, the remaining two requests will be issued.

Client NFSv3 Write.client_scenario*

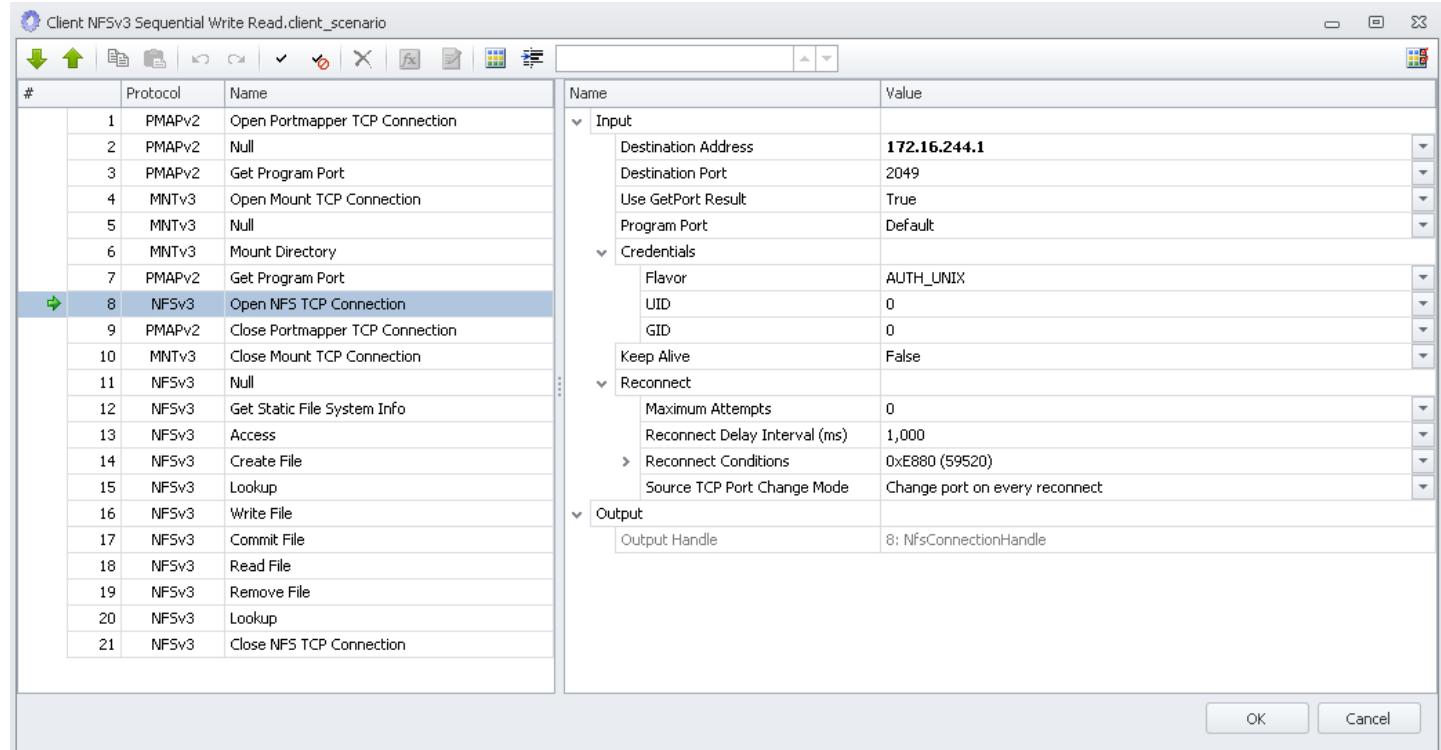
#	Protocol	Name
1	PMAPv2	Open Portmapper TCP Connection
2	PMAPv2	Null
3	PMAPv2	Get Program Port
4	MNTv3	Open Mount TCP Connection
5	MNTv3	Null
6	MNTv3	Mount Directory
7	PMAPv2	Get Program Port
8	NFSv3	Open NFS TCP Connection
9	PMAPv2	Close Portmapper TCP Connection
10	MNTv3	Close Mount TCP Connection
11	NFSv3	Null
12	NFSv3	Access
13	NFSv3	Create File
14	SWT	Begin Loop
15	NFSv3	Write File

Performance Impact

For tests that utilize a single connection between the Load DynamiX Appliance and the DUT (i.e. contain a single Scenario with a Load Specification of 1 Concurrent Scenario or 1 Concurrent Connection), the asynchronous I/O approach described above can provide a significant performance boost. Projects that utilize multiple connections can generally outperform a single connection that does

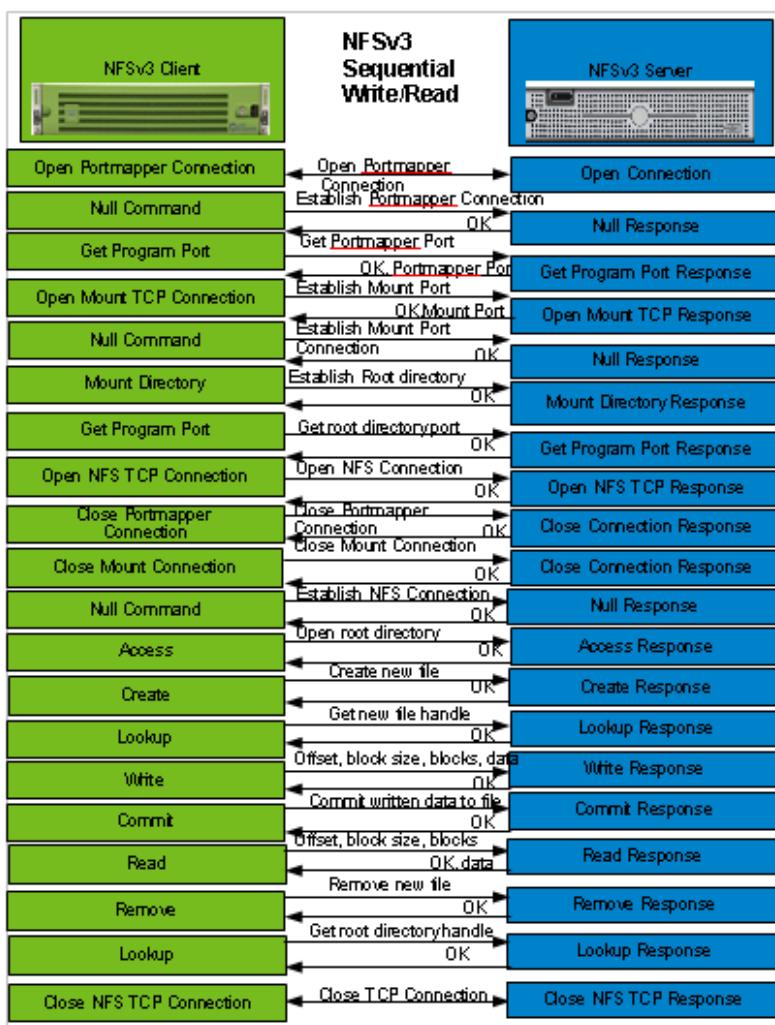
not use asynchronous I/O.

NFSv3 Sample Project



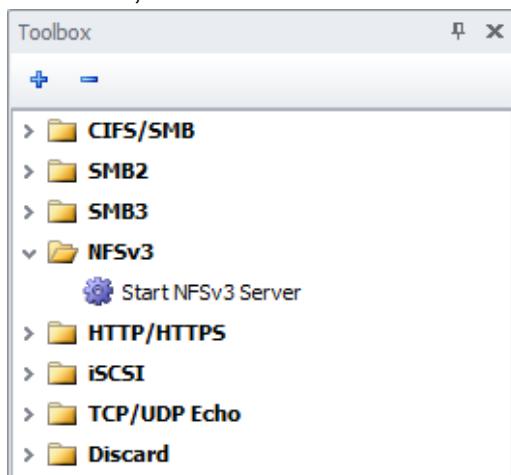
NFSv3 Project Flow

The following is the NFSv3 client/server interaction flow for a simple sequential write read Project.



NFSv3 Server

The Load DynamiX Appliance firmware supports an NFSv3 server emulation. The NFSv3 server is instantiated in a Server Scenario using the Start NFSv3 Server Action and providing, at a minimum, an IP address for the Server.



NFSv3 Statistics

See the [Advanced Concepts: Test Execution Rules](#) for a complete list of the statistics generated when NFSv3 Actions are used in a Scenario.

Reference: NFSv4, v4.1 Command List**Reference: Load DynamiX NFS version 4 command list**

ACTION
Close NFS TCP Connection
Open NFS TCP Connection
Access
Close File
Commit File
Create
Create RPSEC GSS Context
Create Link
Destroy RPSEC GSS Context
Get Attributes
Get Public File Handle
Get Root File Handle
Lock File
Lookup
Lookup Parent Directory
Null
Open File
Open File Confirm
Open File Downgrade
Read Directory
Read File
Read Symbolic Link
Release Lock Owner
Remove
Rename
Renew
Security Info
Set Attributes

Set Client ID
Set Client ID Confirm
Test for Lock
Unlock File
Write File

NFS v4.1 Command List

ACTION
Access
Close File
Close NFS TCP Connection
Commit File
Create
Create Link
Create RPSEC GSS Context
Create Session
Destroy ClientID
Destroy RPSEC GSS Context
Destroy Session
Exchange ID
Free StateID
Get Attributes
Get Device Info
Get Layout
Get Public File Handle
Get Root File Handle
Lock File
Lookup
Lookup Parent Directory
Open File
Open File Downgrade

Open NFS TCP Connection
Open pNFS TCP Connection
Read Directory
Read File
Read Symbolic Link
Reclaim Complete
Remove
Rename
Sequence
Set Attributes
Test For Lock
Unlock File
Write File

Links to NFSv4, NFSv4.1 protocol reference materials are provided in the [References and Terminology section](#).

NFSv4/NFSv4.1 Sample Projects

Client NFSv4 Commands.client_scenario*

#	Protocol	Name
1	NFSv4	Open NFS TCP Connection
2	NFSv4	Null
3	NFSv4	Get Root File Handle
4	NFSv4	Get Attributes
5	NFSv4	Lookup
6	NFSv4	Lookup
7	NFSv4	Access
8	SWT	Create Variable
9	SWT	Create Variable
10	SWT	Create Variable
11	SWT	Create Variable
12	SWT	Create Variable
13	NFSv4	Set Client ID
14	NFSv4	Set Client ID Confirm
15	NFSv4	Open File
16	NFSv4	Open File Confirm
17	NFSv4	Write File
18	NFSv4	Commit File
19	NFSv4	Read File
20	NFSv4	Close File
21	NFSv4	Get Attributes
22	NFSv4	Lookup
23	NFSv4	Create
24	NFSv4	Lookup
25	NFSv4	Access
26	NFSv4	Remove
27	NFSv4	Create
28	NFSv4	Lookup
29	NFSv4	Access
30	NFSv4	Read Directory
31	NFSv4	Read Directory
32	NFSv4	Lookup
33	NFSv4	Rename
34	NFSv4	Remove
35	NFSv4	Remove

Name

Input

Connection Handle	Default
Directory Handle	Default
Client ID Handle	Default
File Name	=@VARIABLE(12)
Open Confirm	Manual
Open Owner	Unique Per Open
Create On Open	True
Create Mode	UNCHECKED

Share reservation

Access	Both
Deny	None

File Attributes

Use Mode Attributes	True
Mode Attributes	0x000001B6 (438)

Output

Output File Handle	15: NfsFileHandle
Output State ID Handle	15: NfsStateIDHandle

Response Handlers

Completion Status	
Scenario Impact	

Client NFSv4.1 Threads Write Read.client_scenario

The screenshot shows a software interface for defining a client scenario. On the left, a list of operations is shown in a table:

#	Protocol	Name
1	NFSv4.1	Open NFS TCP Connection
2	NFSv4.1	Exchange ID
3	NFSv4.1	Create Session
4	NFSv4.1	Get Root File Handle
5	NFSv4.1	Lookup
6	NFSv4.1	Lookup
7	NFSv4.1	Access
8	NFSv4.1	Open File
9	#	---< Begin 5 Write Threads
10	SWT	Begin Loop
11	SWT	Begin Thread
12	SWT	Begin Async
13	NFSv4.1	Write File
14	NFSv4.1	Write File
15	SWT	End Async
16	SWT	End Thread
17	SWT	End Loop
18	SWT	Wait For All Threads
19	#	---> End the Write Threads
20	NFSv4.1	Comm File
21	#	---< Begin 5 Read Threads
22	SWT	Begin Loop
23	SWT	Begin Thread
24	SWT	Begin Async
25	NFSv4.1	Read File
26	NFSv4.1	Read File
27	SWT	End Async
28	SWT	End Thread
29	SWT	End Loop
30	SWT	Wait For All Threads
31	#	---> End the Read Threads
32	NFSv4.1	Close File
33	NFSv4.1	Remove
34	NFSv4.1	Destroy Session

On the right, configuration settings are displayed in a tree view:

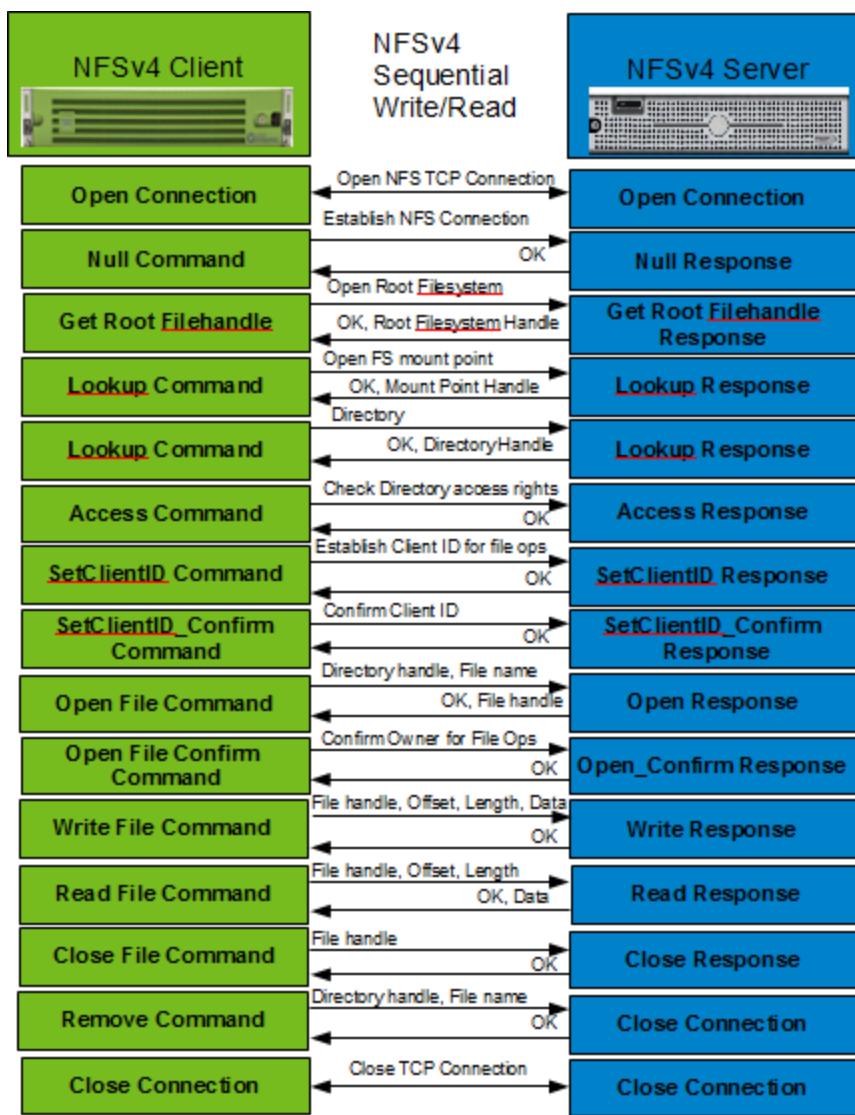
- Input**:
 - Connection Handle: Default
 - Session ID Handle: Default
- Output**:
 - Output Handle: 4: NfsFileHandle
- Response Handlers**:
 - Completion Status
 - Scenario Impact

Buttons at the bottom right: OK and Cancel.

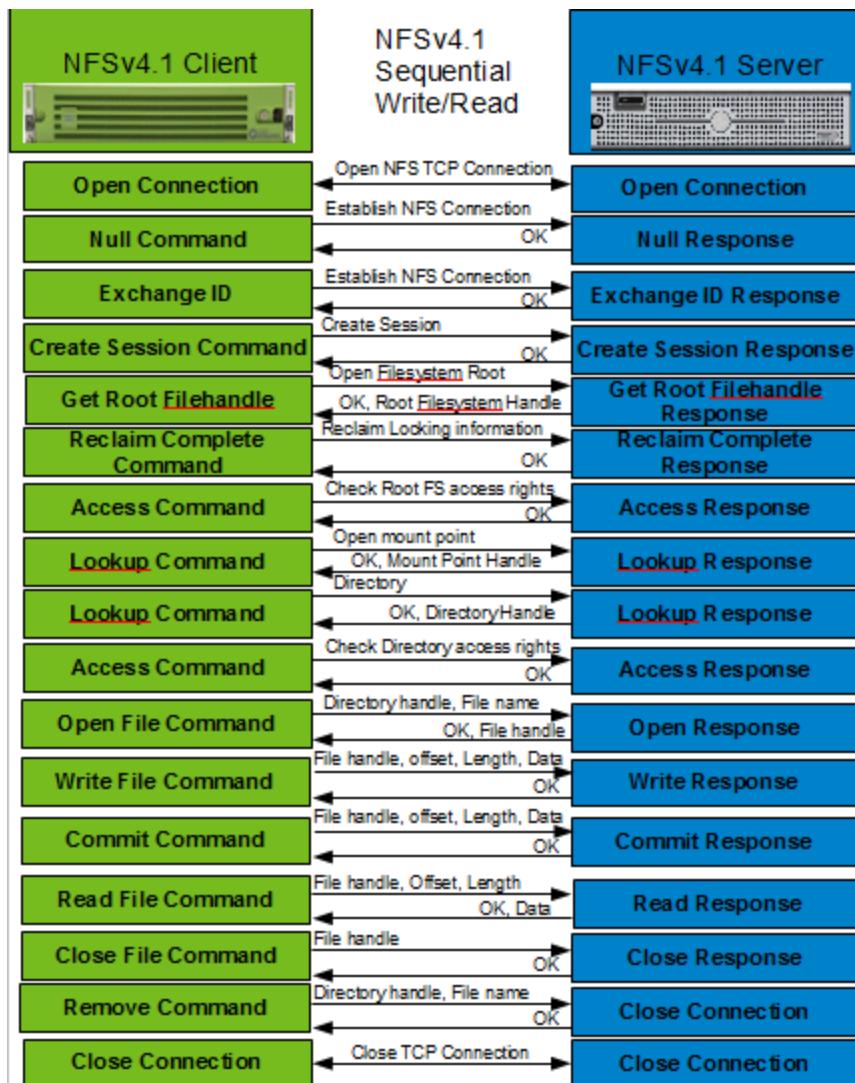
NFSv4/NFSv4.1 Project Flow

The following is the NFSv4/V4.1 client/server interaction flow for a simple sequential write read Project.

NFSv4



NFSv4.1



NFSv4/NFSv4.1

See the [Advanced Concepts: Test Execution Rules](#) for a complete list of the statistics generated when NFSv4/NFSv4.1 Actions are used in a Scenario.

Reference: Kerberos v5 Command List

Reference: Load DynamiX Kerberos v5 command list

ACTION
AS-REQ
Close Kerberos Connection
Open Kerberos Connection
TGS-REQ

A link to Kerberos protocol reference material is provided in the [References and Terminology section](#).

See the [Advanced Concepts: Test Execution Rules](#) for a complete list of the statistics generated when Kerberos Actions are used in a Scenario.

Reference: Load DynamiX HTTP/HTTPS Commands and Behaviors

Reference: Load DynamiX HTTP/HTTPS Commands and Behaviors

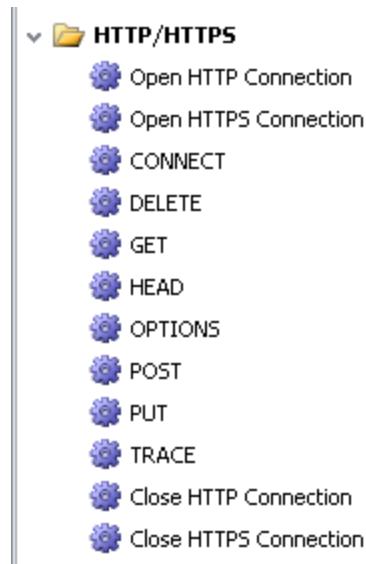
ACTION
Close HTTP connection
Close HTTPS connection
Open HTTP connection
Open HTTPS connection
CONNECT
DELETE
GET
HEAD
OPTIONS
POST
PUT
TRACE
SERVER ACTION
Start HTTP Server
Start HTTPS Server

A link to HTTP/HTTPS protocol reference material is provided in the [References and Terminology section](#).

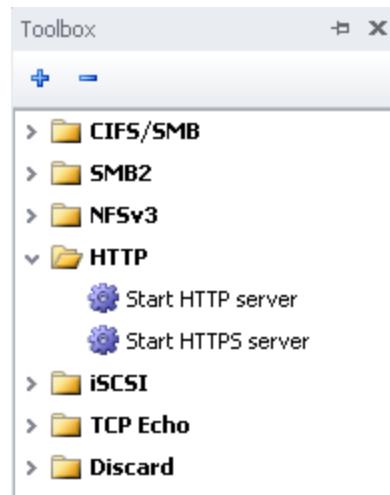
HTTP/HTTPS

The HTTP protocol serves as a means to transfer applet, text and image content to and from web server pages to browsers and as a framework to deliver more complex content such as video or animation. HTTP is a request-response protocol. HTTP clients submit content requests to HTTP servers which return a completion status and possibly the requested content. HTTP is an Application Layer protocol designed within the framework of the Internet Protocol Suite. HTTPS is the secured version of the HTTP protocol. A link to HTTP/HTTPS protocol reference material is provided in the [References and Terminology section](#).

The HTTP/HTTPS Client Actions that are supported by the Load DynamiX TDE and Appliance are:



The HTTP Server Actions that are supported by the Load DynamiX TDE and Appliance are:



HTTP Sample Project

The screenshot shows the configuration dialog for an 'HTTP Client.client_scenario'. The scenario table contains the following steps:

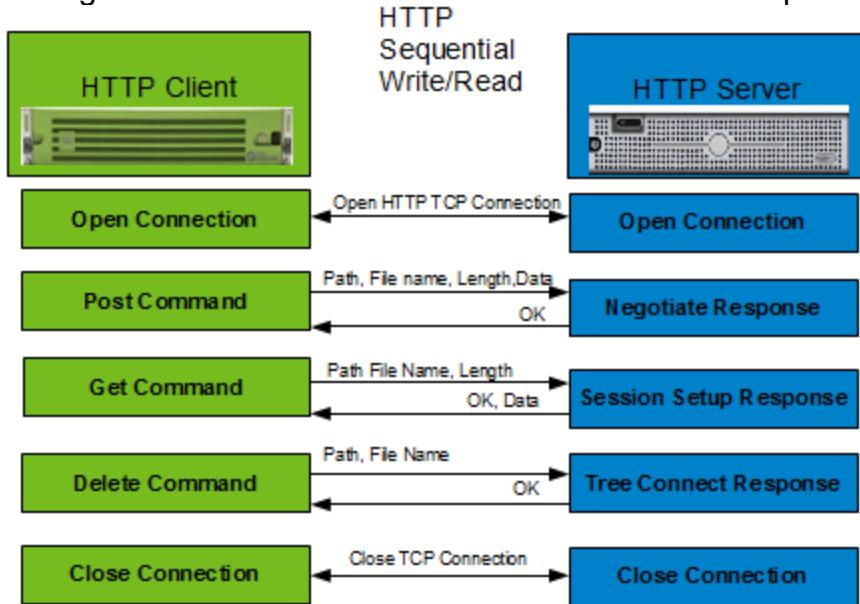
#	Protocol	Name
1	HTTP	Open HTTP Connection
2	#	<i>Initial GET to SWT server.</i>
3	HTTP	GET
4	#	<i>Values extracted from the server response to previous GET are being used in this step.</i>
5	HTTP	GET
6	HTTP	Close HTTP Connection

The right panel displays detailed configuration settings:

- Input:** Connection Handle (Default), Request version (1.1), Request URI (/example.json).
- Request Headers:** Transfer Encoding (None).
- Authentication:** Preemptive Authorization (None), Passive Authorization (Basic Enabled, Digest Enabled, NTLM: NTLM Disabled, Negotiate: NTLM Disabled, Negotiate: Ker... Disabled).
- Credentials:** Username and Password fields.
- Entity data:** Entity data field.
- Data Verification:** On Failed Verification (Continue).
- Scenario Impact:** On Failed Verification (Continue).
- Output:** Extract Headers and Extract Body options.

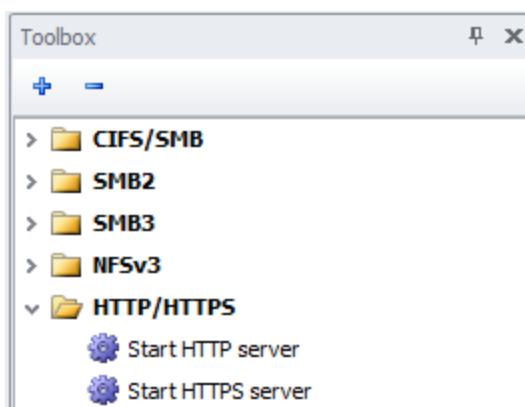
HTTP Project Flow

The following is the HTTP client/server interaction flow for a simple sequential write read Project.



HTTP/HTTPS Server

The Load DynamiX Appliance firmware supports HTTP and HTTPS server emulations. The HTTP server is instantiated in a Server Scenario using the Start HTTP Server Action and providing, at a minimum, an IP address for the Server. Likewise, the HTTPS server is instantiated in a Server Scenario using the Start HTTPS Server Action and providing, at a minimum, an IP address for the Server.

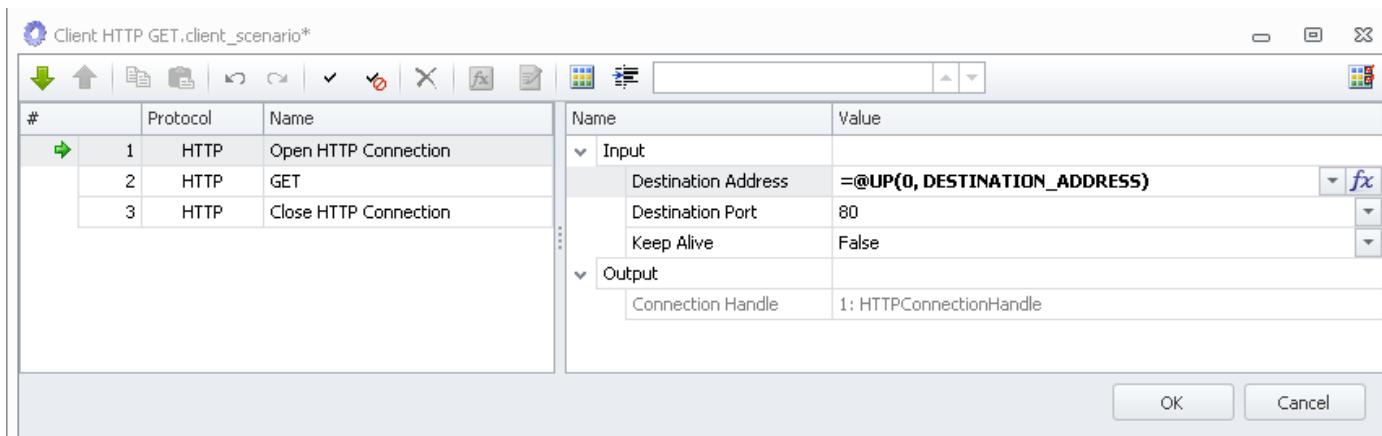


HTTP/HTTPS Statistics

See the [Advanced Concepts: Test Execution Rules](#) for a complete list of the statistics generated when HTTP/HTTPS Actions are used in a Scenario.

HTTP/HTTPS Open Connection Host Name

The **Open HTTP Connection** and **Open HTTPS Connection** Actions support Fully Qualified Domain Names (as defined by RFC 3886) in the Destination Address input field. The contents of the Host header is constructed, by default, using the contents of the Destination Address field of the **Open HTTP/HTTPS Connection** Action.



This behavior can be changed by using the "Override Host Header" setting in the HTTP Header Editor. The default behavior (flag not checked) is to construct the Host header from the contents of the **Open HTTP Connection** and **Open HTTPS Connection** Actions. There is no longer a Host header shown in the Request Headers list below. It is automatically added and contains the constructed Host from Open Connection Actions. The Content Editor will not allow a Header named "Host" to be added to the Request Headers list unless the "Override Host Header" is enabled. See below for examples:

Not Enabled = No Host header allowed

HTTP Content Editor*

Request Headers		Preset ▾
Name	Value	
User-Agent	SwiftTest (http://www.swifttest.com)	
Accept	text/html, text/plain, text/css, text/sgml, */*;q=0.01	
Accept-Encoding	identity	
Accept-Language	en	
<input checked="" type="checkbox"/> Host		
<input type="checkbox"/> Override Host Header		

HTTP headers

Enabled = Host header allowed

HTTP Content Editor*

Request Headers		Preset ▾
Name	Value	
User-Agent	SwiftTest (http://www.swifttest.com)	
Accept	text/html, text/plain, text/css, text/sgml, */*;q=0.01	
Accept-Encoding	identity	
Accept-Language	en	
<input checked="" type="checkbox"/> Host		
<input checked="" type="checkbox"/> Override Host Header		

HTTP headers

When "Override Host Header" is enabled, HTTP requests contains the Tester-defined Host header (the contents of the Host header created using the Content Editor) only.

Notes:

- When Override Host Header is enabled, the first HTTP request is issued with the substituted Host header. However, if the address is redirected, the next HTTP request will be sent with the address from the Location field.
- Amazon S3 Host header processing does not conform to the HTTP/HTTPS Host header processing rules described above. See [Reference: HTTP Storage Commands and Behaviors](#) for Amazon S3 Host processing rules.

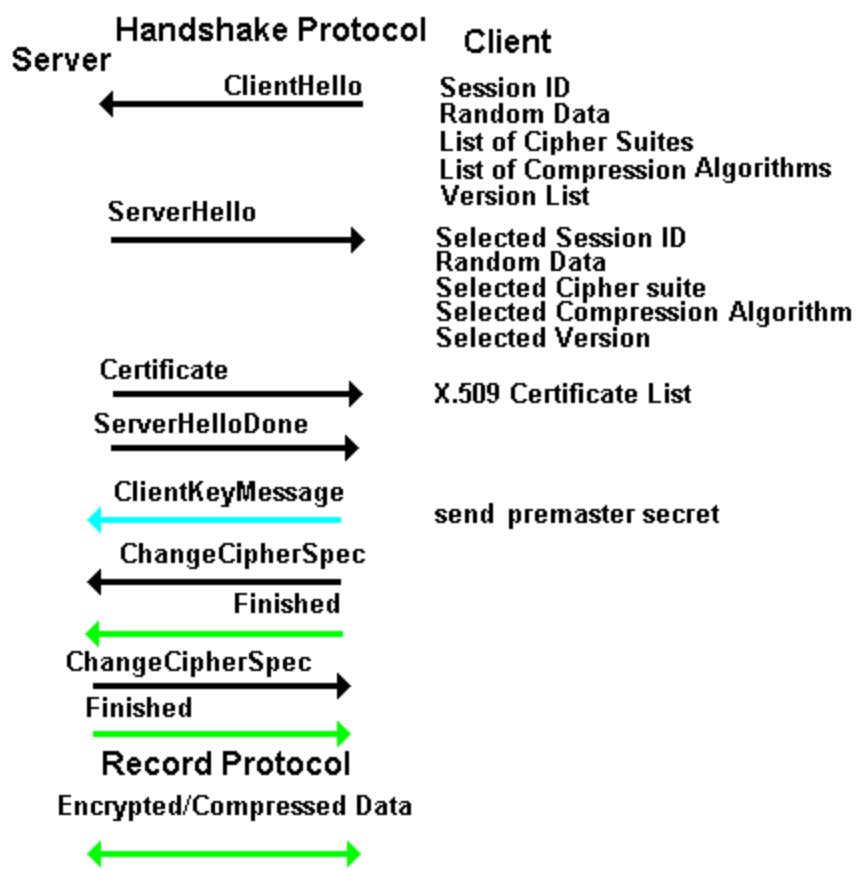
HTTPS

Certificates

Secure communications between HTTP Client and Server (the "HTTPS" protocol) requires the availability of SSL Certificates (also known as X509 certificates) present on the Server and (optionally) the Client. In the Load DynamiX TDE, certificates are stored in Projects in a resource file named the Certificate Content and referenced in the Open HTTPS Connection, Initiate SSL/TLS Handshake and Start HTTPS Server Actions as

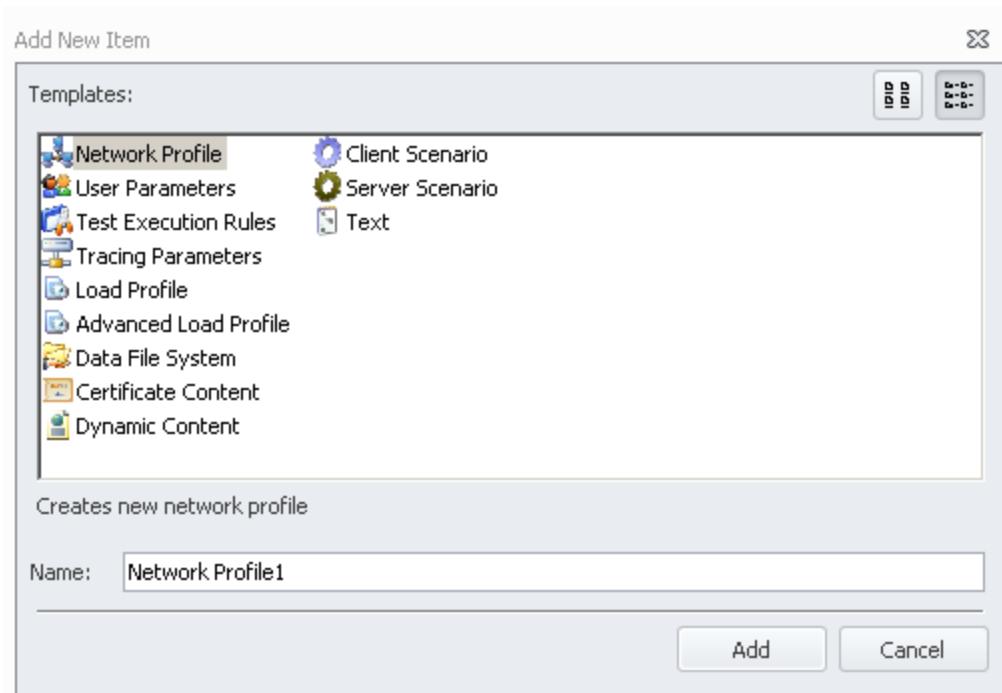
::CertificateContent(x), where (x) is a reference to an entry in the Certificate Content resource file

The typical handshake between an HTTPS Client and Server in which only the Server provides its certificates might look like this



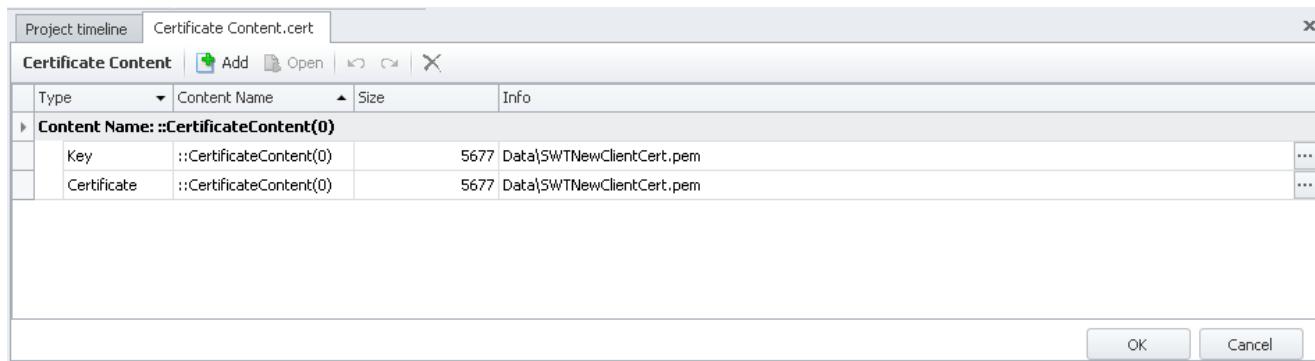
Certificate Content Resource file

To create a Certificate Content Resource, click the Add New Item button on the Project Explorer toolbar or click Add new Item from the Project drop down menu. Highlight the Certificate Content entry, provide a name such as My Test Certificates, and click Add.



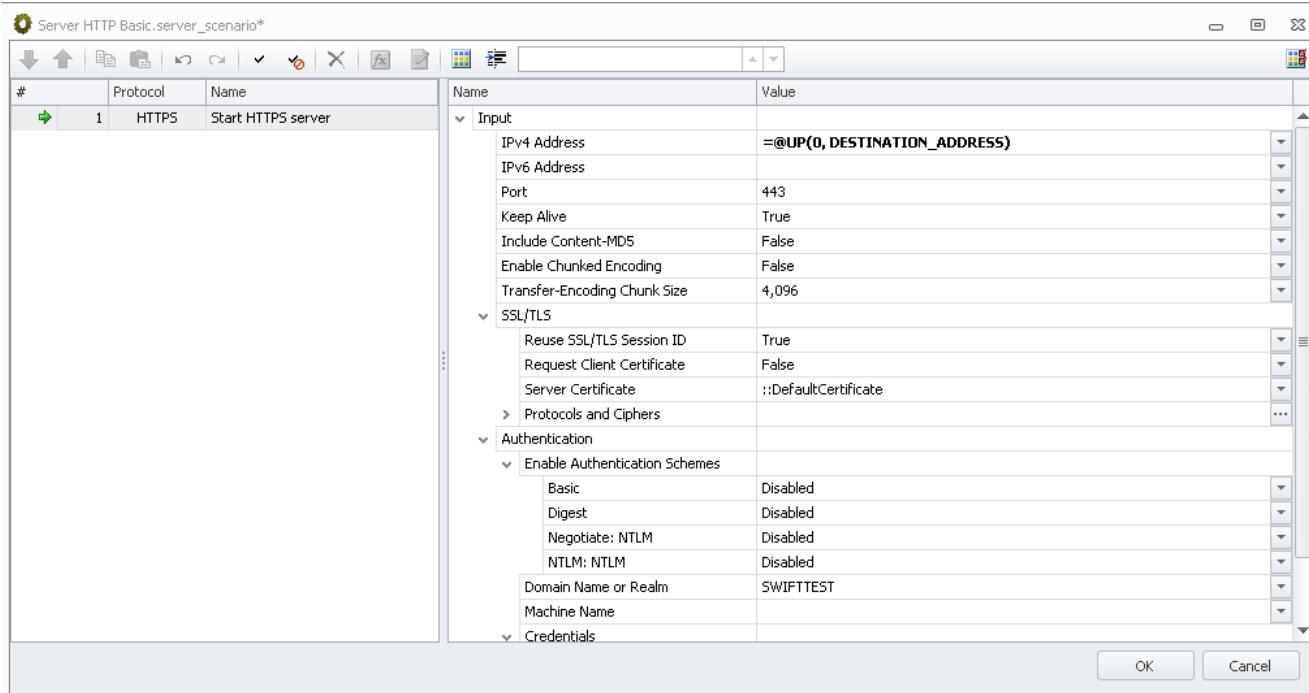
An empty Certificate Content editor window will be opened

Click the Add button to select the Certificate to add to the Certificate Content Resource. After Adding the certificates, the Certificate Content will look like



To use the certificates in a **Start HTTPS Server or Open HTTPS Connection Action**, drop the Certificate Content Resource onto the Server and/or Client Timeline (Logical Port, Network Profile or Scenario) and refer to them in the Actions as follows:

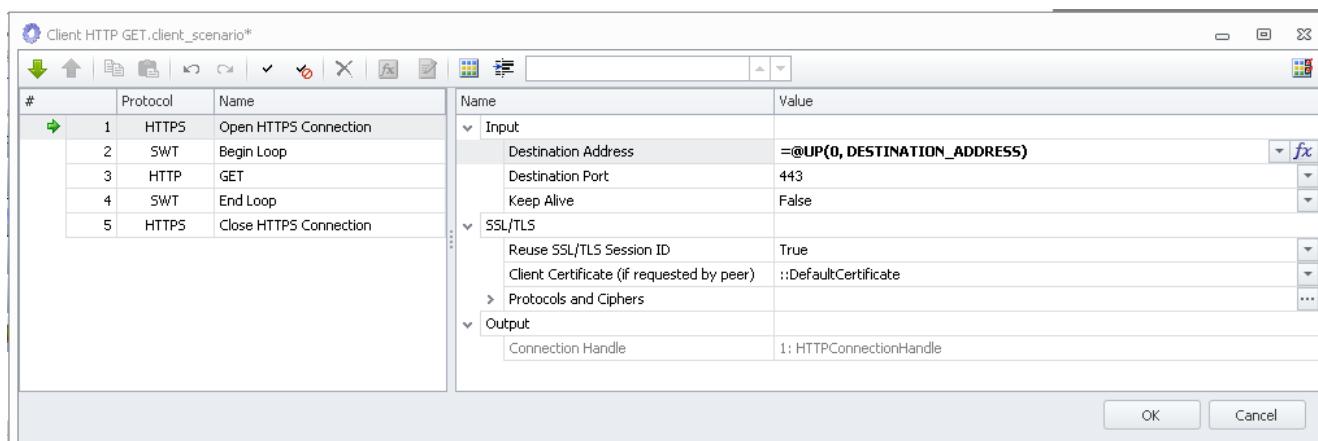
Start HTTPS Server Action:



Note: In the above Action, the Authenticate Client input field is set to True. This forces the Load DynamiX HTTPS server to request a certificate from the Client. If the Client were not provisioned with a certificate as shown below, the HTTPS session will fail. If the Authenticate Client input field is set to False (the default value) then the only certificate exchange is the Server sending its certificate to the Client.

The above process allows Testers to use certificates that have been created outside of the Load DynamiX software. If the Tester needs to use a Load DynamiX supplied certificate then use the ::DefaultCertificate selection in the Certificate input field. See below.

Open HTTPS Connection Action:



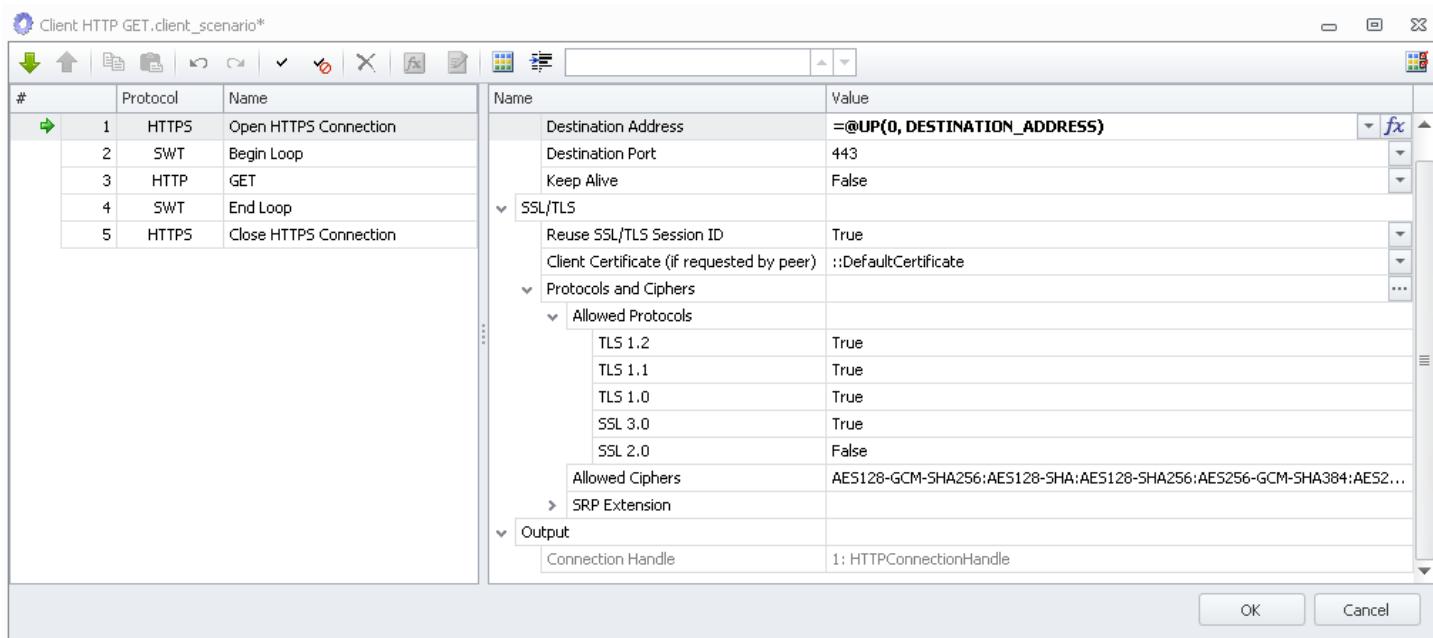
Protocols and Ciphers

Protocols

Load DynamiX HTTPS Client and Server support SSLv2, SSLv3 and TLSv1 (1.0, 1.1 and 1.2) protocols.

Ciphers (examples)

AES128-SHA256, AES256-SHA256, CAMELLIA128-SHA, etc.



See [Reference: HTTP Command List](#) for additional information regarding the supported HTTP and HTTPS Actions.

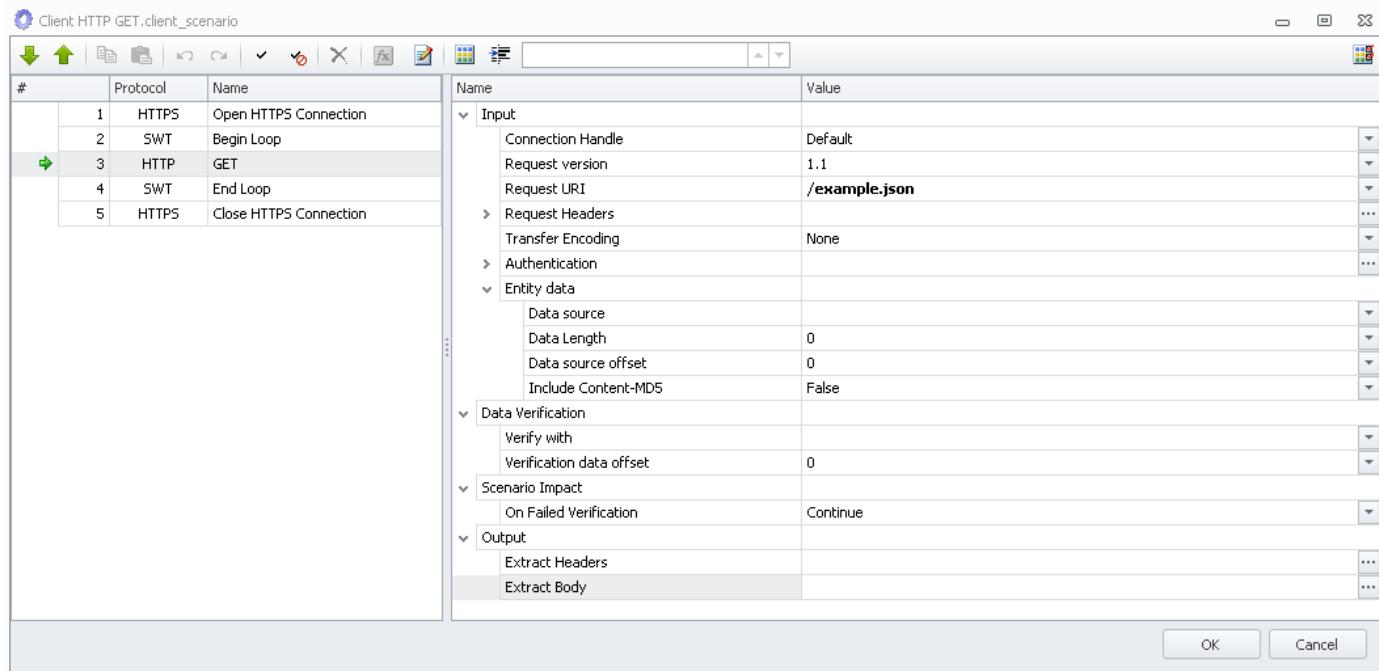
HTTP/HTTPS Header and Body Parsing, and Dynamic Content

HTTP Projects may require that various standard or customer HTTP Header or Body contents to be sent in or extracted from HTTP packets. The Load DynamiX HTTP Actions (Put, Get, Post, Connect, Head, Delete, Options and Trace) allow the Tester to specify:

- Header contents to be sent as part of an Action (**Request Headers**) or received in response to the Action (**Extract Headers**)
- Body contents to be sent as part of an Action (**Request Body**) or received in response to the Action (**Extract Body**)

Request Headers are specified in the HTTP Content Editor which can be opened by either clicking

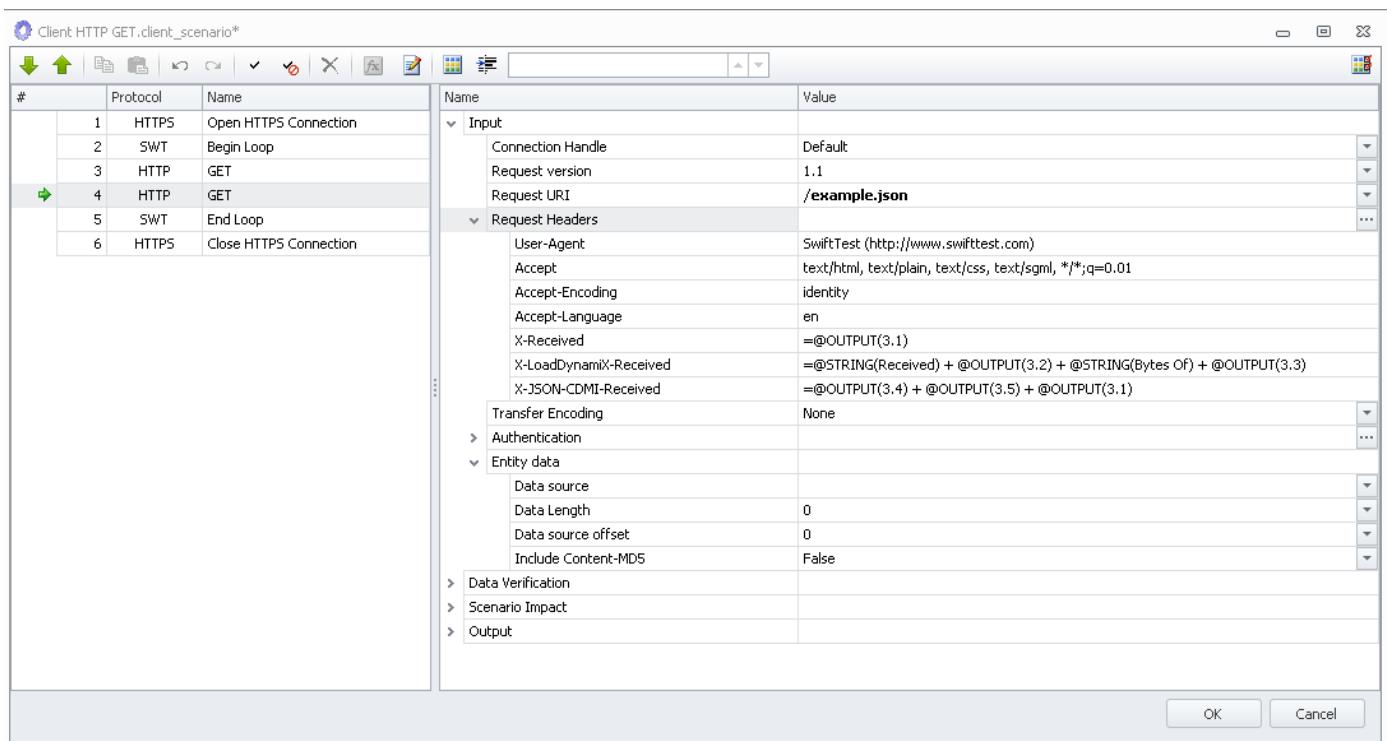
the HTTP Content Editor button on the tool bar or by clicking the "3 dots" button on the Request Headers field in the scenario editor .



Dynamic Content

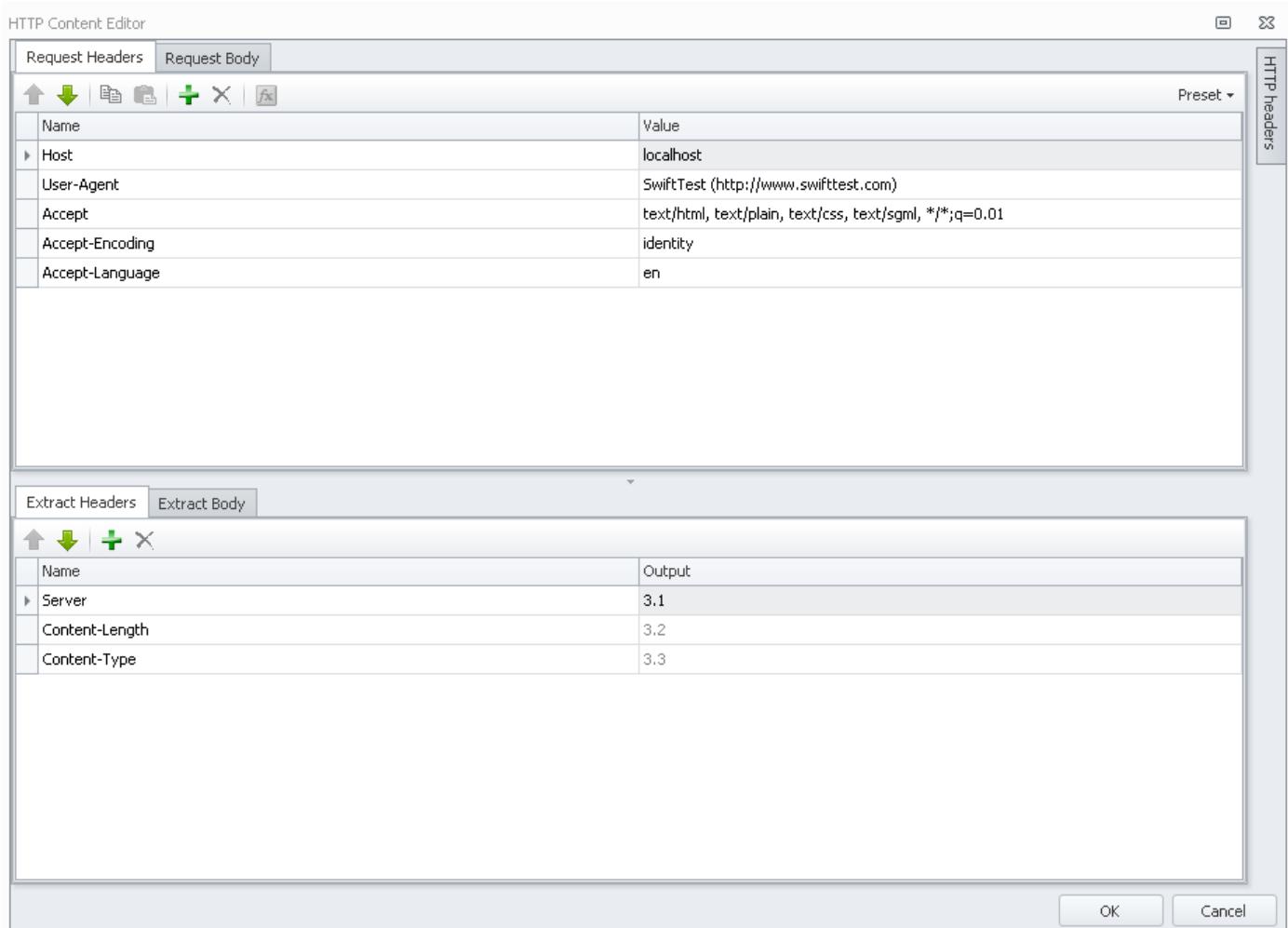
HTTP information (Headers or Body contents) extracted from HTTP packets is stored in a Resource called Dynamic Content. Dynamic Content is the equivalent of a Data File System in that it defines a "file" associated with an element of the Dynamic Content Resource: ::DynamicContent(0), ::DynamicContent(1), ::DynamicContent(N), etc. The Tester adds the DynamicContent Resource to a Project and then associates file with ::DynamicContent(i) entries and then used in HTTP operations. The contents of the ::DynamicContent(i) is based on information that is extracted from HTTP packets before it is used.

In the Scenario below, the second GET uses DynamicContent(0) to hold information extracted from the first GET and then delivers that information as the Entity Data of the second GET.



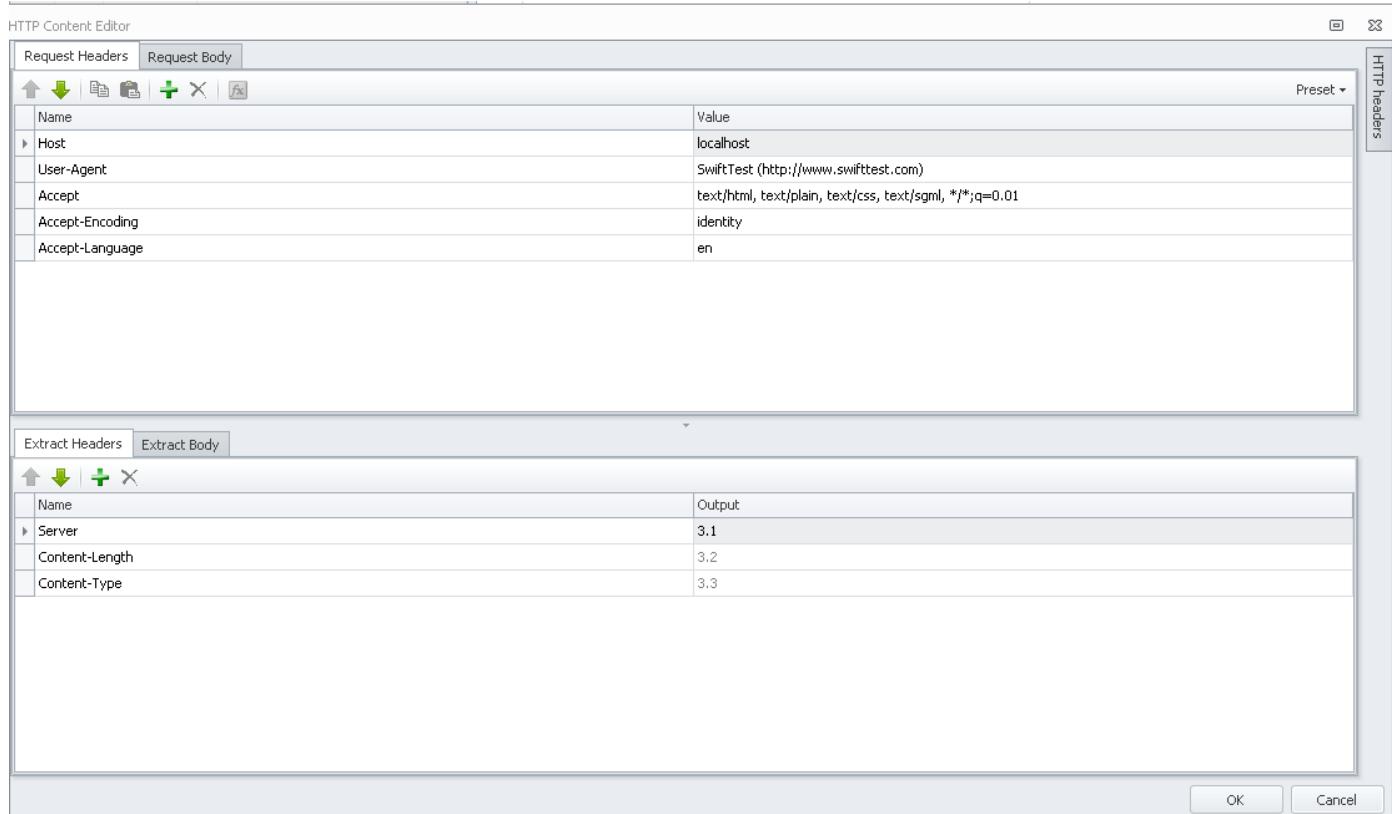
Request Headers

Headers added to the Request Headers list will be sent to the HTTP Server. Headers added to the Extract Headers field will be read from the HTTP Server. In the screenshot below, two new headers have been added to the Request Headers list for the PUT Action in the Scenario above. This was accomplished by opening the HTTP Content Editor using the PUT Action's button and using the button to add new entries. The Value field of the new headers may contain Function content as shown below.



Extract Headers

To define Headers to read (Extract) from an HTTP server, use the HTTP Content Editor to add entries to the Extract Header list. Click the HTTP Content Editor button for the desired Action and use the to add items to the Extract Headers list. The screenshot above shows three Headers added the Extract Headers list of the GET Action in line 3 of the Scenario above. Notice the Output field for each of these headers: 3.1, 3.2 and 3.3. These values refer to the first, second and third Extract Headers of the Action in line 3. To reference the value of these headers after the Action in line 3 has been executed, use the @OUTPUT() function.



Request and Extract Body

Similarly for HTTP Body contents to be extracted and used in future HTTP operations, the Content Editor is used to manipulate HTTP Body contents. In the screenshot below, the fourth through sixth elements of the OUTPUT() function for line 3 are defined in the Extract Body tab.

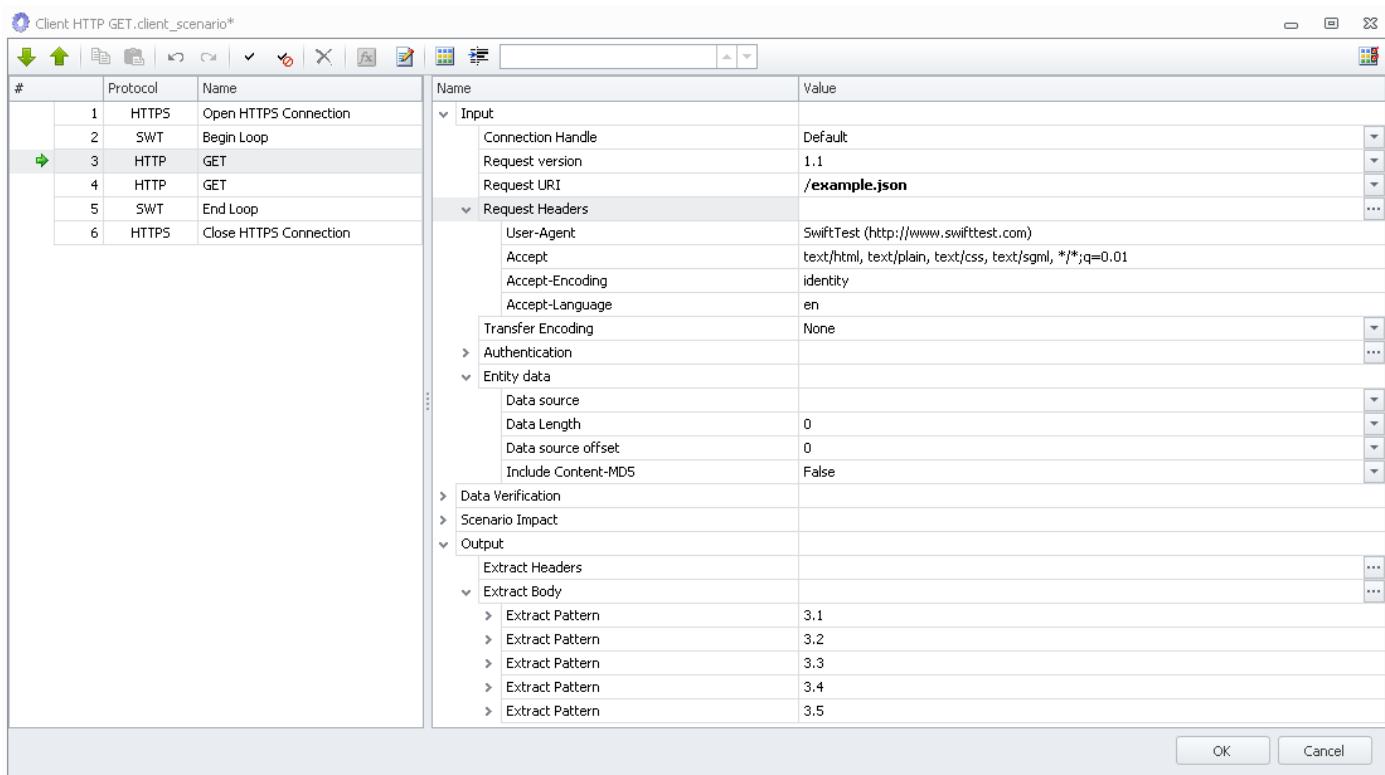
HTTP Content Editor

Request Headers		Request Body	Preset ▾
Name	Value		
Host	localhost		
User-Agent	SwiftTest (http://www.swifttest.com)		
Accept	text/html, text/plain, text/css, text/sgml, */*;q=0.01		
Accept-Encoding	identity		
Accept-Language	en		

Extract Headers		Extract Body
Name	Value	
Extract Pattern	3.4	
Content-Type	JSON	
Path	/SwiftTest/Protocols/Protocol/NFS[Action="Access"]/version	
Extract Pattern	3.5	
Content-Type	JSON	
Path	/SwiftTest/Protocols/Protocol/SMB[version=1]/Action[3]	
Extract Pattern	3.6	
Content-Type	JSON	
Path	/SwiftTest/Media/Ethernet[1]/gigabits	

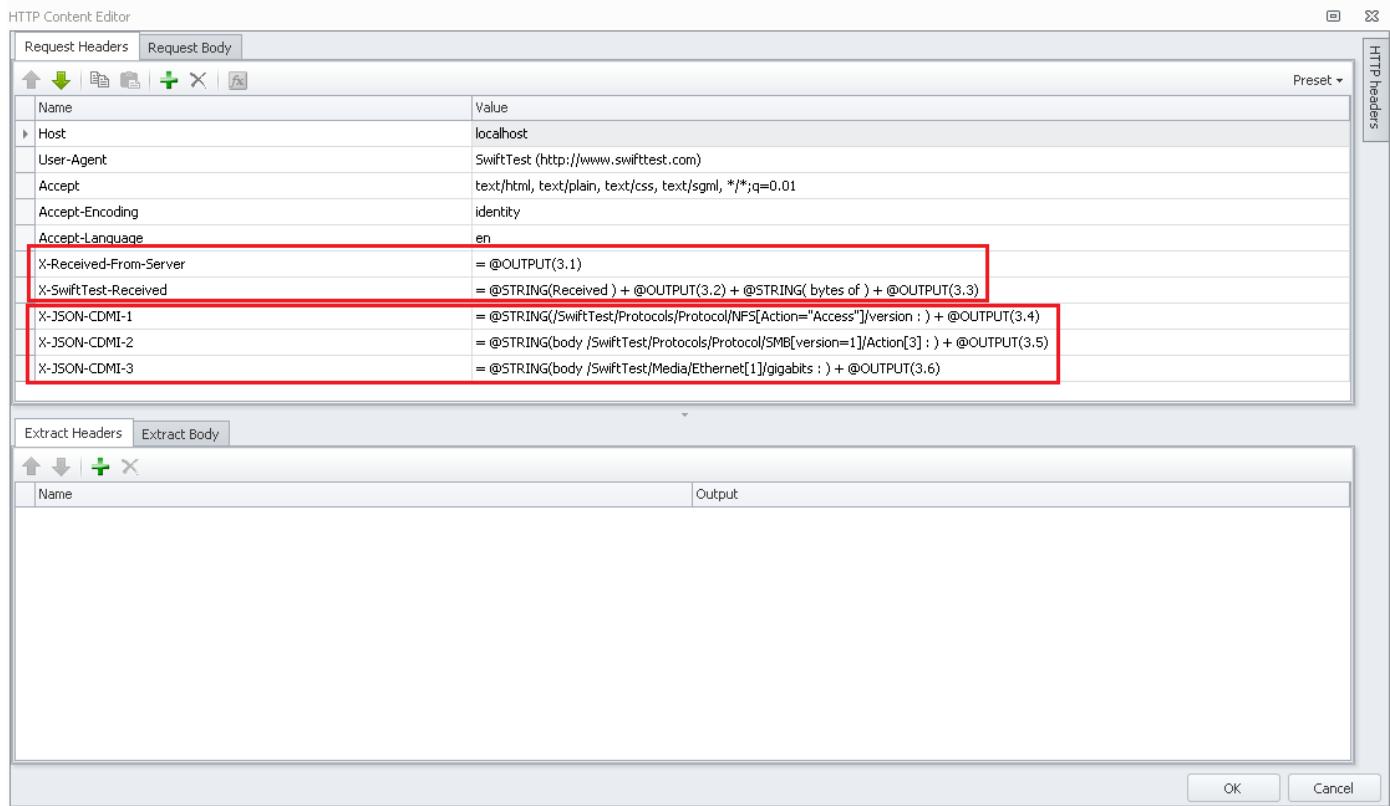
OK **Cancel**

Click OK on the HTTP Content Editor and the new headers can be seen in the Action's Request Header's field and in the Output field where the extracted information is recorded.

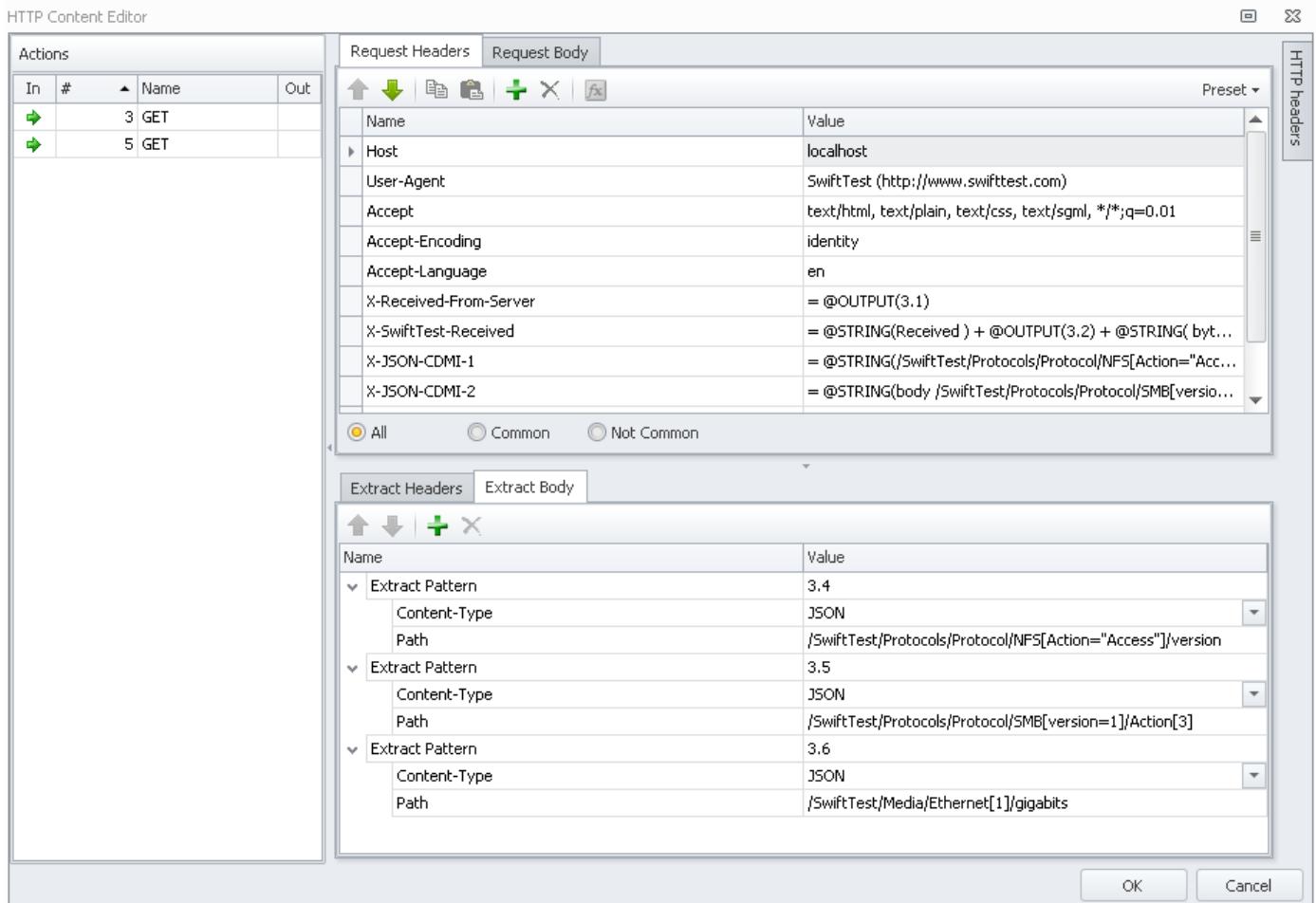


Using Extracted Header and Body Contents to Create Custom Headers or Body Content

If it were necessary to send the custom content back to the HTTP server in a subsequent Action in this Scenario, a header can be added to the Request Headers list that contains a reference to the value read from the HTTP server. The GET Action in Line 5 can reference the value returned by the Action in Line 3 by adding to the Request Headers list and referring to the output of Action 3 using the @OUTPUT() Function as seen below [OUTPUT(3.1), OUTPUT(3.2), and OUTPUT(3.3)]. Body contents extracted from the response to the GET in line 3 can also be used in Headers as demonstrated below in the references below to OUTPUT(3.4), OUTPUT(3.5) and OUTPUT(3.6).



To edit the Request Headers or Extract Headers lists of more than one Action at a time, highlight the Actions to be modified and click the button. The HTTP Content Editor open for two Actions (line 3 GET and line 5 GET) is shown below.



HTTP/HTTPS Redirect

The Load DynamiX HTTP and HTTPS Clients support handling of HTTP 3XX status codes returned from HTTP and HTTPS servers. If an HTTP or HTTPS Server responds to an HTTP or HTTPS request with a redirect response, the HTTP or HTTPS Client software will accept the Redirect response and act as appropriate to the kind of Redirect received.

The following PCAP segment illustrates the HTTPS Client handling a 307 Redirect response. In line 9, the HTTP Client (172.16.91.1) opens a connection to an HTTP Server 172.16.0.7 and then issues an HTTP GET request for a file named "/W2k8". The HTTP Server responds to that request with a Temporary Redirect (HTTP 307 message) response that informs the Client that the file /W2k8 is on a different server (172.16.1.48) with a different name (/redirect).

9 2.000675	172.16.91.1	172.16.0.7	TCP	62 exec > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256
10 2.001266	172.16.0.7	172.16.91.1	TCP	62 http > exec [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 WS=2
11 2.001355	172.16.91.1	172.16.0.7	HTTP	282 GET /w2k8 HTTP/1.1
12 2.001498	172.16.0.7	172.16.91.1	TCP	60 http > exec [ACK] Seq=1 Ack=229 Win=128480 Len=0
13 2.001741	172.16.0.7	172.16.91.1	HTTP	534 HTTP/1.1 307 Temporary Redirect (text/html)
14 2.001764	Intel_0a:b1:19	Broadcast	ARP	60 who has 172.16.1.48? tell 172.16.91.1

In line 33, the Client logs into the new Server (172.16.1.48) and in line 35 issues a GET request for the new file (/redirect).

33 5.002710	172.16.91.1	172.16.1.48	TCP	62 login > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256
34 5.004534	172.16.1.48	172.16.91.1	TCP	62 http > login [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256
35 5.004544	172.16.91.1	172.16.1.48	HTTP	280 GET /redirect HTTP/1.1

The Load DynamiX HTTP and HTTPS Clients support all HTTP 3xx messages. Load DynamiX HTTP and HTTPS Clients support FQDN as defined by RFC3986 (see PCAP example below)

8 1.000890	172.16.93.1	172.16.1.58	DNS	105 standard query A ipv4-58-1.apache.https.localdomain
9 1.000894	172.16.93.1	172.16.1.58	DNS	105 standard query AAAA ipv4-58-1.apache.https.localdomain
10 1.001442	172.16.1.58	172.16.93.1	DNS	179 Standard query response A 172.16.1.58
11 1.001450	172.16.1.58	172.16.93.1	DNS	146 Standard query response
12 1.001474	172.16.93.1	172.16.1.58	TCP	62 2001 > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256
13 1.001592	172.16.1.58	172.16.93.1	TCP	62 http > 2001 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 WS=64
14 1.001752	172.16.93.1	172.16.1.58	HTTP	255 GET /ipv4-58-2 HTTP/1.1
15 1.002138	172.16.1.58	172.16.93.1	TCP	60 http > 2001 [ACK] Seq=1 Ack=202 win=30272 Len=0
16 1.002155	172.16.1.58	172.16.93.1	HTTP	598 HTTP/1.1 301 Moved Permanently (text/html)

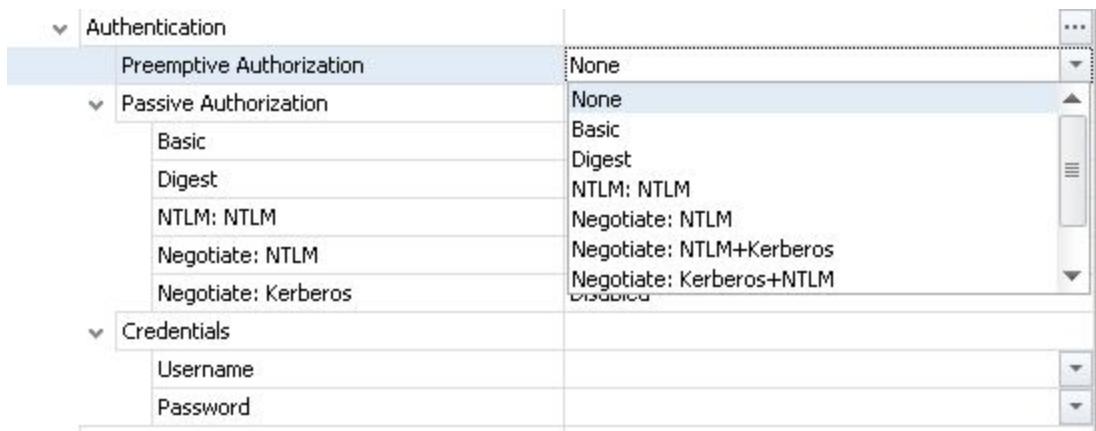
The Load DynamiX HTTP and HTTPS servers do not support redirection.

HTTP/HTTPS Authentication

All of the HTTP protocol Actions (those other than **Open HTTP Connection** and **Close HTTP Connection**), support a variety of Authentication inputs. In the screenshots below are seen the choices that exist for all of the HTTP protocol Actions:



The drop down menu choices for Preemptive Authorization are similar to Passive Authorization but also includes Amazon S3 authentication (which does not support Passive Authentication):



Authentication Modes:

Users may choose whether to automatically provide authentication information (Preemptive Authorization) or to provide it upon request from the HTTP server (Passive Authorization). The two input fields allow the user to tell the Load DynamiX software which to use. The Preemptive Authorization can be set using the DropDownList menu or the Authorization Parameters dialog box (see below). The Passive Authorization parameters can only be set using the Authorization Parameters dialog box either by clicking on the ... button on the Authentication field or double clicking any of the Passive Authorization entries.

HTTP Authentication Parameters



Preemptive Authorization:	Passive Authorization:				
<input type="radio"/> None <input type="radio"/> Basic <input type="radio"/> Digest <input checked="" type="radio"/> NTLM: NTLM <input type="radio"/> Negotiate: NTLM <input type="radio"/> Negotiate: NTLM+Kerberos <input type="radio"/> Negotiate: Kerberos+NTLM <input type="radio"/> Negotiate: Kerberos <input type="radio"/> Amazon S3	Enabled: NTLM: NTLM Disabled: Basic Digest Negotiate: NTLM Negotiate: Kerberos				
Credentials:	NTLM Information:				
Username: <input type="text" value="User001"/> Password: <input type="text" value="Pass001"/> Kerberos Ticket Handle: Kerberos TKT: <input type="text" value="Default"/>	Domain Name: <input type="text" value="SWIFTTEST"/> Machine Name: <input type="text"/> NTLM Flags: <table border="1"> <thead> <tr> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>NTLM Flags</td> <td>0x00080007 (524295)</td> </tr> </tbody> </table>	Name	Value	NTLM Flags	0x00080007 (524295)
Name	Value				
NTLM Flags	0x00080007 (524295)				
<input type="button" value="OK"/> <input type="button" value="Cancel"/>					

To Enable an Authorization method, highlight the method in the Disabled list and click the **Enable** button. To Disable an Authentication method, highlight the method in the Enabled list and click the **Disable** button. To change priority in the Enabled list, highlight the method and click Priority Up or Priority Down buttons.

Use Cases:

Passive: Client does not know what kind of Authentication processes the HTTP server supports so it waits for the HTTP server to tell it what kind of authentication to provide.

Preemptive: Client knows in advance what kind of Authentication processes the HTTP server supports so it sends the supported type in advance of being asked (also reduces some network traffic by removing the need for the HTTP server to inform the Client of what kind of Authentication it supports).

Preemptive Authorization (send Authentication info to the Server without being requested to do so):

- None, Disable Preemptive Authentication (no Preemptive Authorization will be attempted)
- Basic, Use Basic HTTP Authentication (Username and Password are lightly obscured and sent to the Server)

- Digest, Apply a hash function to the Password before sending to the Server
- NTLM: NTLM, Use the enhanced (challenge/response) NTLM protocol to specify authentication information
- Negotiate: NTLM, Use the Negotiate protocol (also known as SPNEGO) to deliver NTLM security information
- Negotiate: NTLM+Kerberos, Use the Negotiate protocol to deliver both NTLM and Kerberos security information, NTLM first as preference
- Negotiate: Kerberos+NTLM, Use the Negotiate protocol to deliver both Kerberos and NTLM security information, Kerberos first as preference
- Negotiate: Kerberos, Use the Negotiate protocol to deliver Kerberos security information
- Amazon S3: Use Amazon Web Services style authentication

Passive Authorization (send Authentication info to the Server if/when requested):

- If all Passive Authentication modes are Disabled, no Passive Authentication will be supported
- If Passive Authentication is being used, more than one Authorization mode response from Client to Server is possible and is determined by which Authorization modes are Enabled and the order that the enabled modes are listed in the Passive Authorization window.
- Passive Authorization supports the following Authorization modes
 - Basic, Use Basic HTTP Authentication (Username and Password are lightly obscured and sent to the Server)
 - Digest, Apply a hash function to the Password before sending to the Server
 - NTLM: NTLM, Use the enhanced (challenge/response) NTLM protocol to specify authentication information
 - Negotiate: NTLM, Use the Negotiate protocol (also known as SPNEGO) to deliver NTLM security information
 - Negotiate: Kerberos, Use the Negotiate protocol to deliver Kerberos security information

In the Passive Authorization example below

Passive Authorization	
NTLM: NTLM	Enabled
Negotiate: Kerberos	Enabled
Basic	Disabled
Digest	Disabled
Negotiate: NTLM	Disabled

If the HTTP Server offers NTLM + Kerberos + Digest: Client uses enhanced NTLM because it is Enabled and at the top of the list.

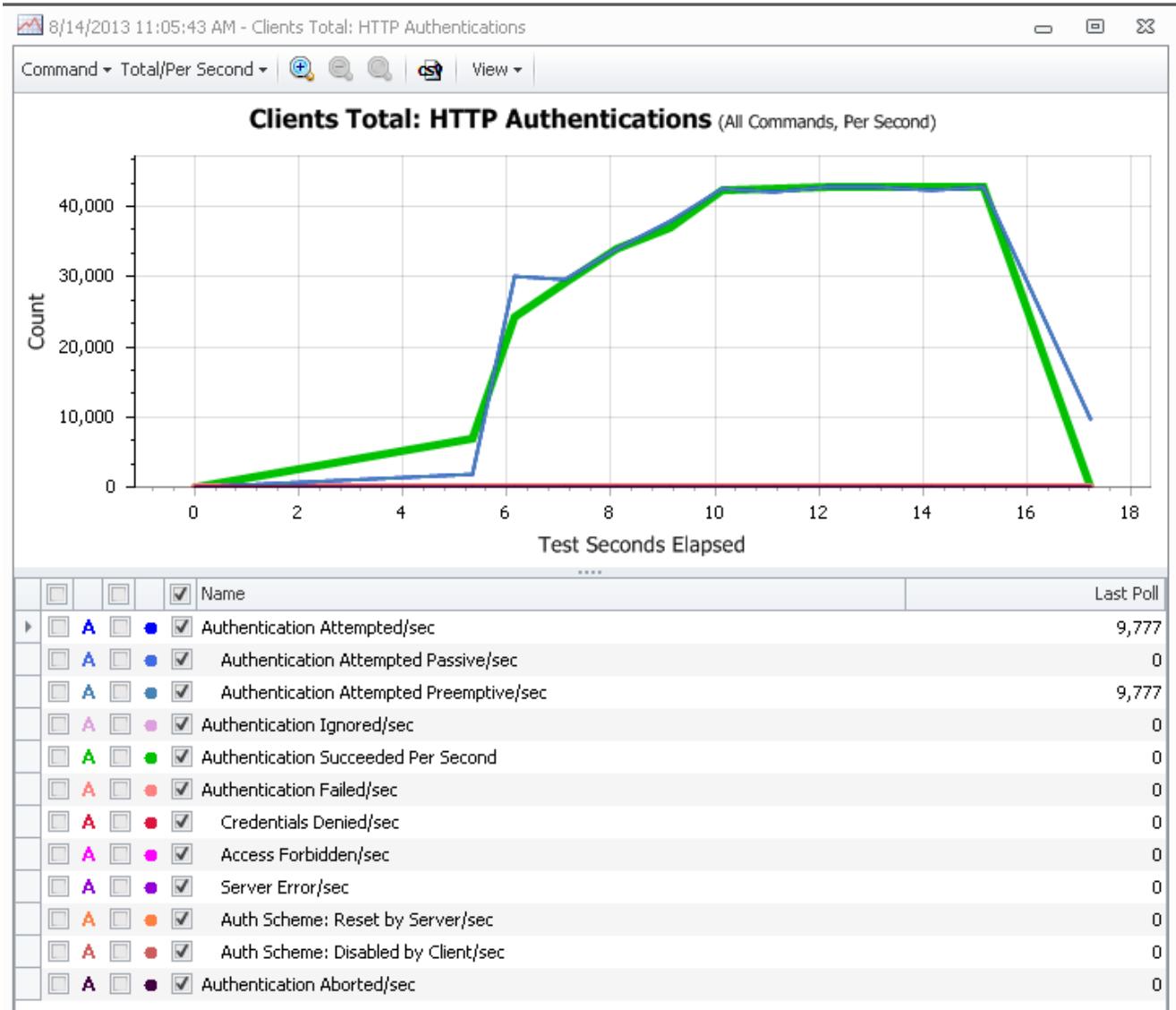
If the HTTP Server offers Kerberos + Digest: Client uses Kerberos using the Negotiate protocol.

If the HTTP Server offers Digest: Client fails authentication

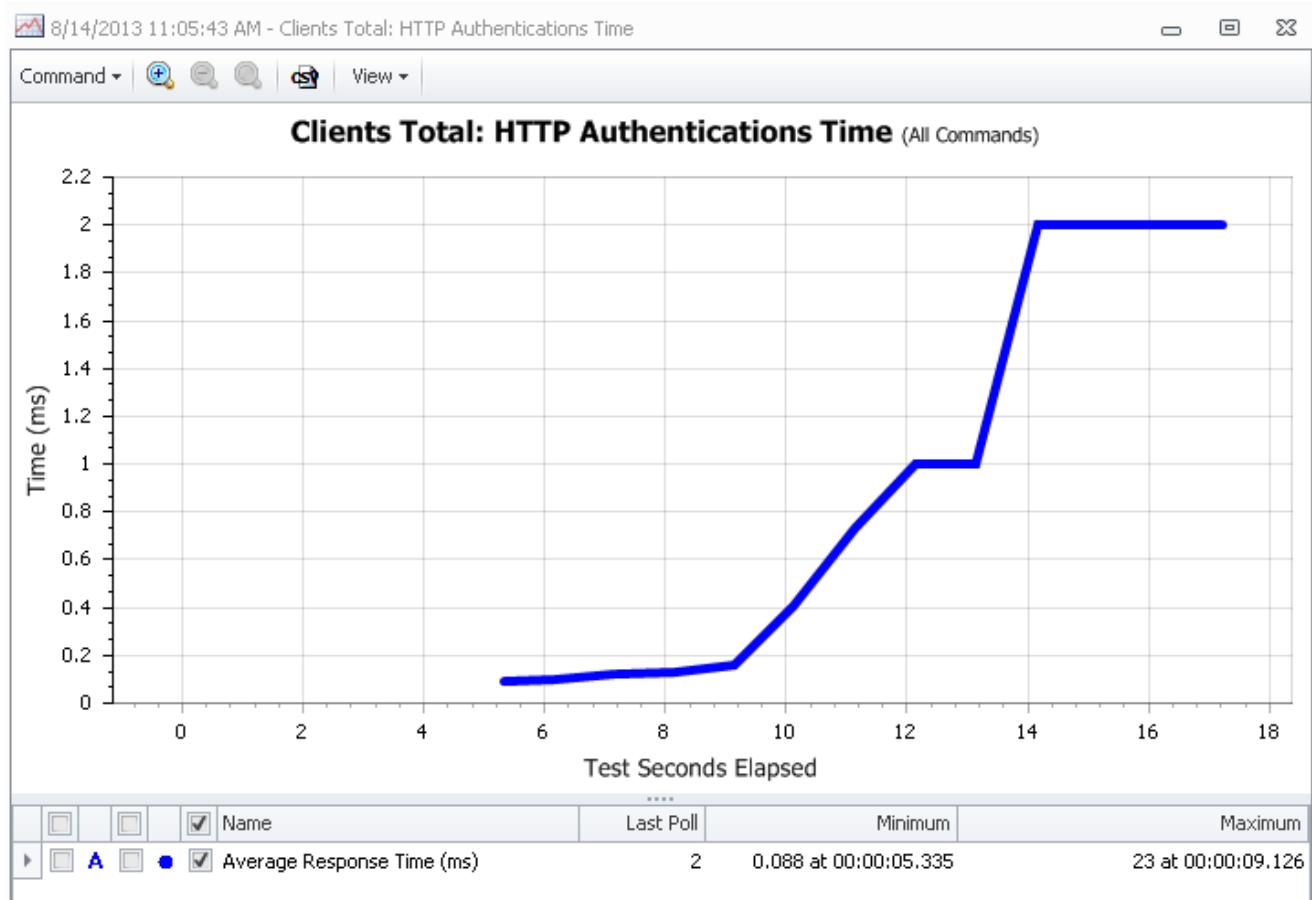
Authentication Statistics

When HTTP Authentication is enabled in HTTP Actions, statistics graphs showing Attempted/Successful/Failed Authentication attempts and the time that the Authentication attempts take.

HTTP Authentications



HTTP Authentications Time



HTTP/HTTPS Content and Transfer Encoding

HTTP Client and Server interactions will have enhanced flexibility by the use of Content Encoding and/or Transfer Encoding.

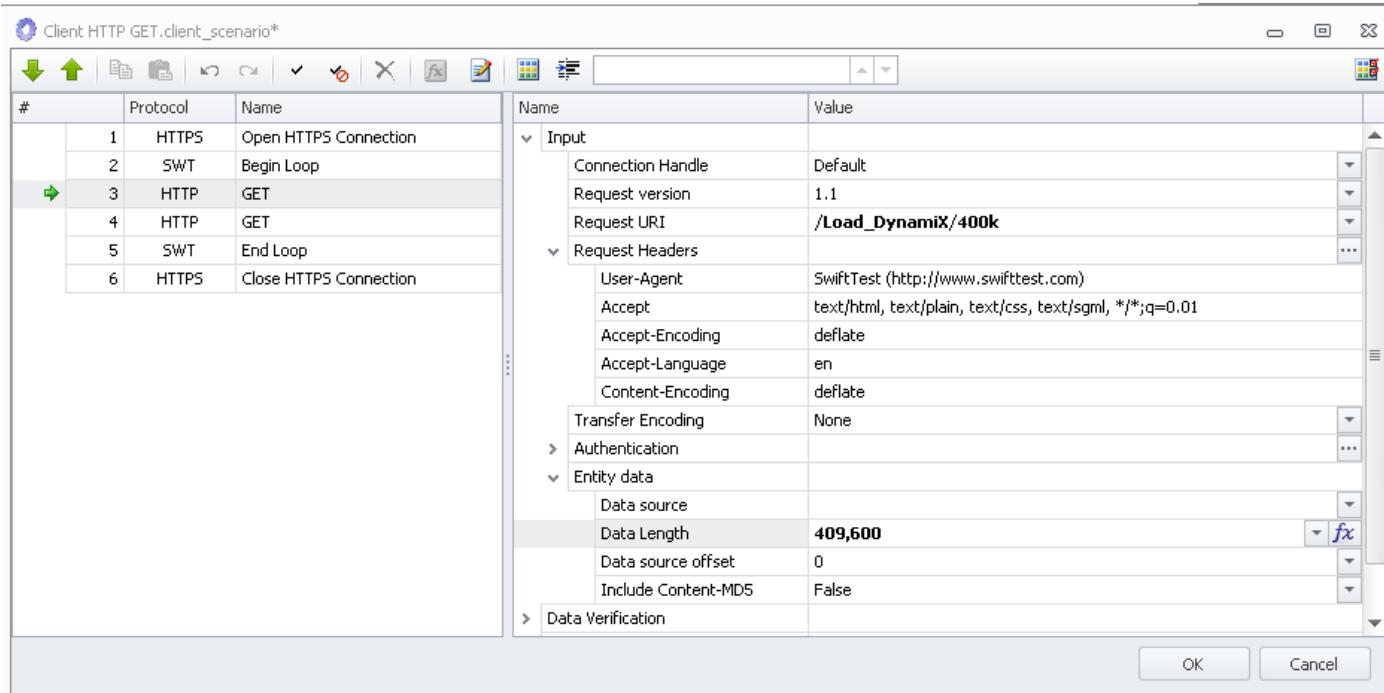
Content Encoding

HTTP Client and Server interactions can be enhanced by compressing the data that is being sent from Client to Server or vice versa. Data compression is specified in the HTTP Actions that are content-capable (**GET, POST, PUT, DELETE, HEAD, OPTIONS**) by adding Accept-Encoding and Content-Encoding headers. The value specified by these headers is either:

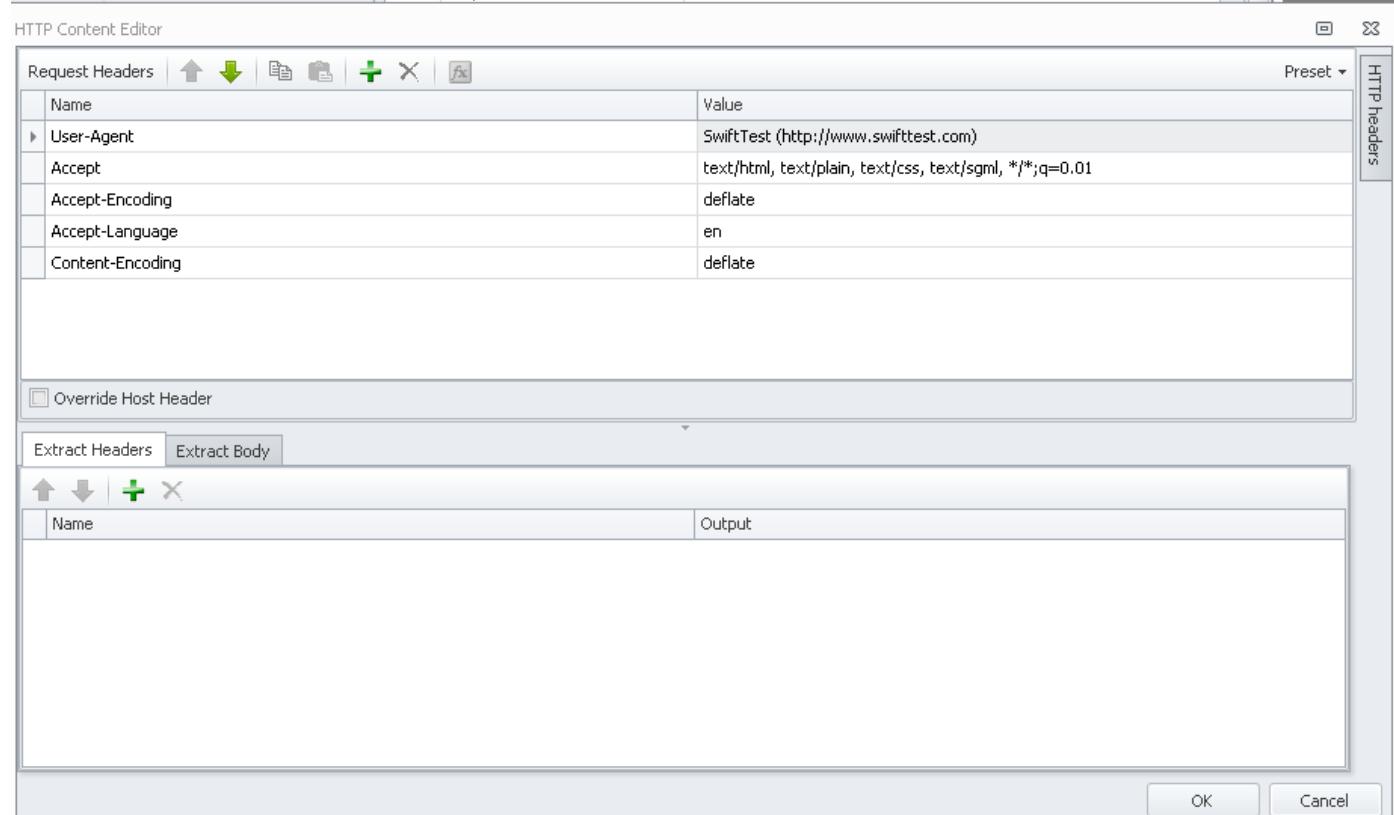
- gzip - use the gzip algorithm to compress/decompress the content
- deflate - use the deflate (zlib) algorithm to compress/decompress the content
- identity - don't do any compression/decompression of the content

Note: For performance reasons, LDX HTTP Client and Server limit the size of the data to be compressed for transmission to 1MB per request/reply. If the LDX HTTP Client or Server is requested to transmit more than 1MB of data in a single request/reply, it transmits the data using "identity" encoding only, within allowances and restrictions of section 14.3 of RFC 2616 (before June 2014) and section 5.3.4 of RFC 7231 (since June 2014).

In the screenshot below, the PUT Action has added Accept-Encoding and Content-Encoding headers specifying that the Deflate algorithm be used to compress/decompress the content.



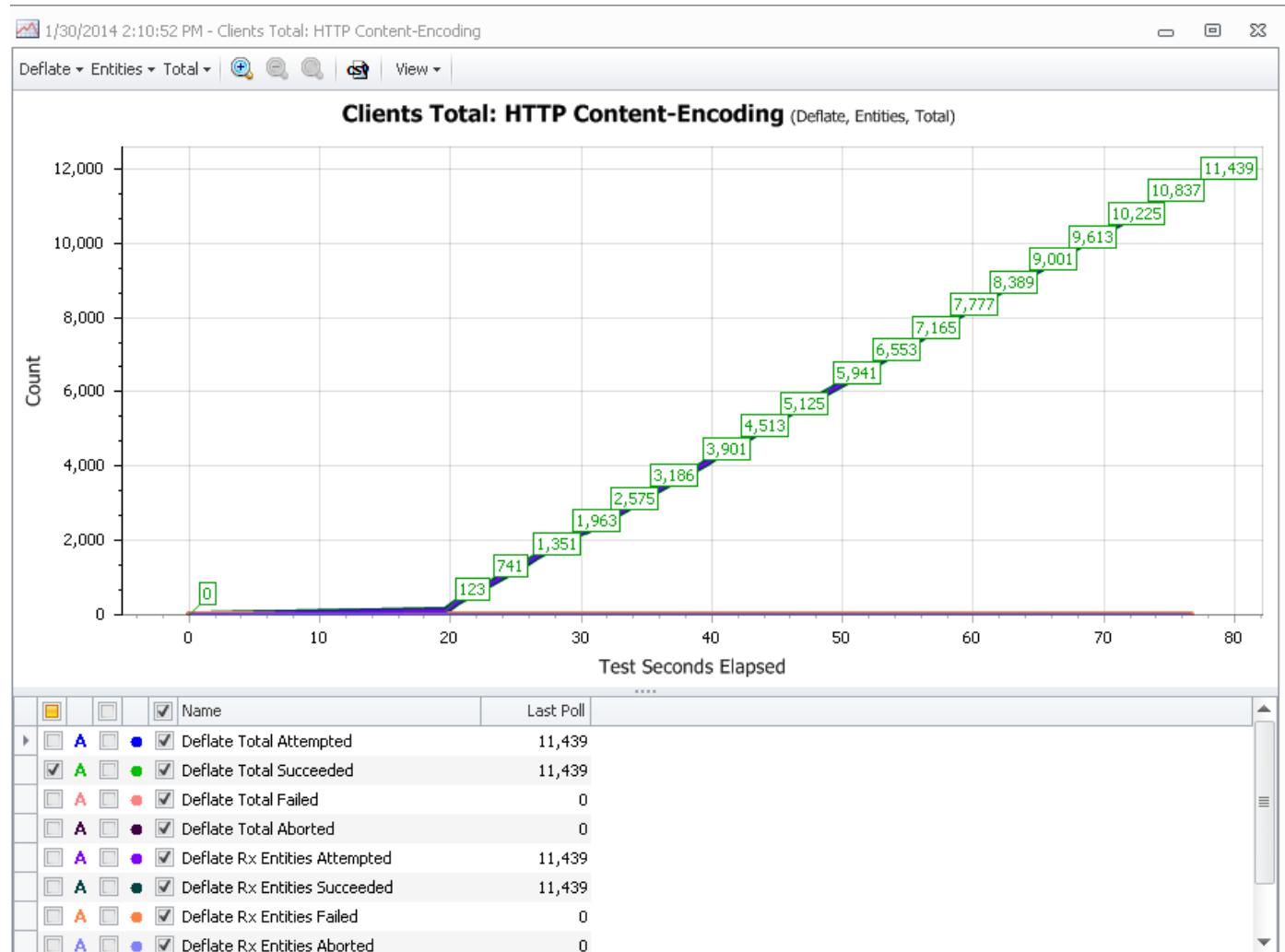
The Accept-Encoding header informs the receiver what encoding the sender is able to Accept (deflate encoding in this case). The Content-Encoding header informs the receiver what encoding the sender is using to send content that is encoded (deflate encoding in this case). The Request Headers are created using the HTTP Content Editor which is opened by clicking on the Request Headers line.



Content Encoding Graphs and Log File Content

HTTP Content Encoding graphs and Log file content will be generated regardless of whether a

specific encoding (ex: gzip) is specified or not due to the fact that Content-Encoding:identity is presumed if no other encoding is specified.

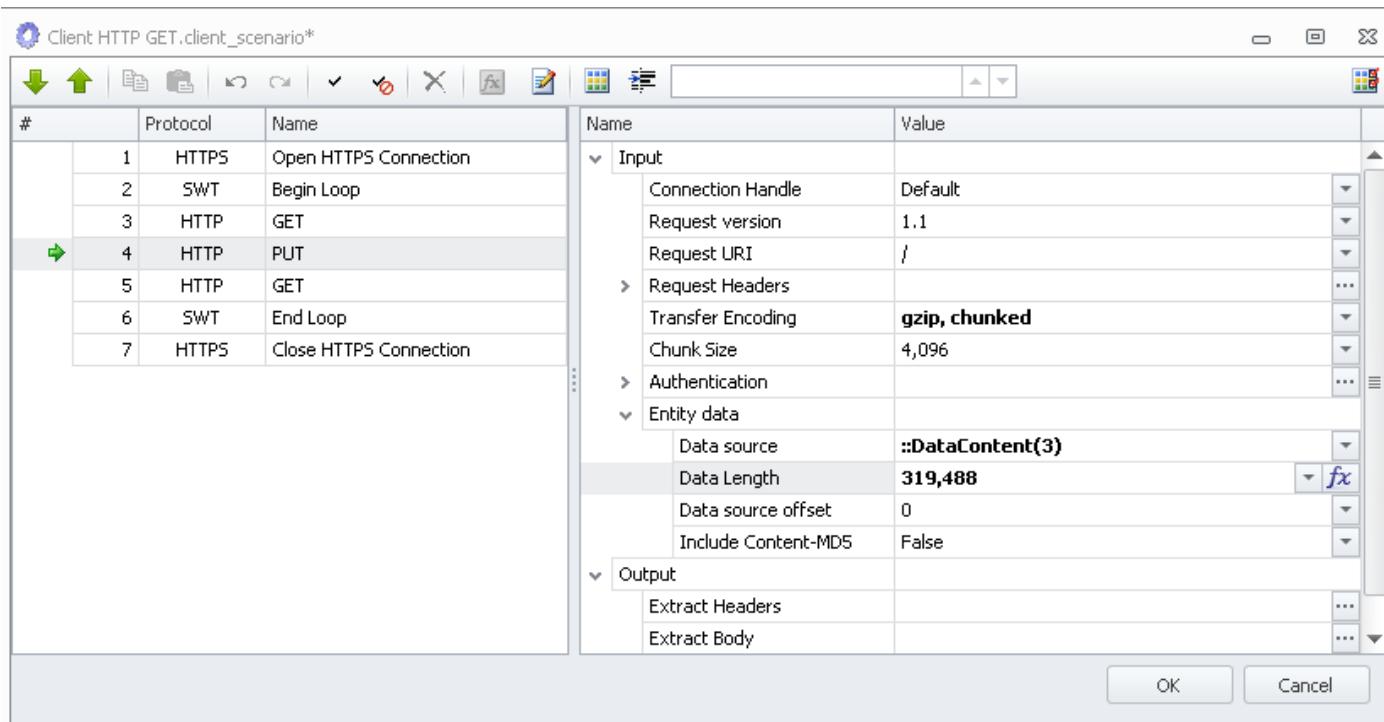


Debug	Info	Warning	Error	Exception	Status	All
Line	Type	Date / Time	Text			
104	Info	1/30/2014 2:04:54 PM	=====			
105	Info	1/30/2014 2:04:54 PM	HTTP deflate Content-Encoding Statistics	Attempted	Succeeded	Failed
106	Info	1/30/2014 2:04:54 PM	=====			Aborted
107	Info	1/30/2014 2:04:54 PM	Total:	11439	11439	0
108	Info	1/30/2014 2:04:54 PM	Received entities with Content-Encoding:	11439	11439	0
109	Info	1/30/2014 2:04:54 PM	-----			
110	Info	1/30/2014 2:04:54 PM	Total Bytes Encoded:	21333735		
111	Info	1/30/2014 2:04:54 PM	- Received Bytes Encoded:	21333735		
112	Info	1/30/2014 2:04:54 PM	Total Bytes Decoded:	4685414400		
113	Info	1/30/2014 2:04:54 PM	- Received Bytes Decoded:	4685414400		
114	Info	1/30/2014 2:04:54 PM	=====			
115	Info	1/30/2014 2:04:54 PM				
116	Info	1/30/2014 2:04:54 PM	=====			
117	Info	1/30/2014 2:04:54 PM	HTTP chunked Transfer-Encoding Statistic	Attempted	Succeeded	Failed
118	Info	1/30/2014 2:04:54 PM	=====			Aborted
119	Info	1/30/2014 2:04:54 PM	Total:	11439	11439	0
120	Info	1/30/2014 2:04:54 PM	Sent messages with Transfer-Encoding:	11439	11439	0
121	Info	1/30/2014 2:04:54 PM	-----			
122	Info	1/30/2014 2:04:54 PM	Total HTTP Entity Bytes:	1429875		
123	Info	1/30/2014 2:04:54 PM	- Sent HTTP Entity Bytes:	1429875		
124	Info	1/30/2014 2:04:54 PM	Total HTTP Message Bytes:	1555704		
125	Info	1/30/2014 2:04:54 PM	- Sent HTTP Message Bytes:	1555704		
126	Info	1/30/2014 2:04:54 PM	=====			
127	Info	1/30/2014 2:04:54 PM				
128	Info	1/30/2014 2:04:54 PM	=====			
129	Info	1/30/2014 2:04:54 PM	HTTP gzip Transfer-Encoding Statistics	Attempted	Succeeded	Failed
130	Info	1/30/2014 2:04:54 PM	=====			Aborted
131	Info	1/30/2014 2:04:54 PM	Total:	11439	11439	0
132	Info	1/30/2014 2:04:54 PM	Sent messages with Transfer-Encoding:	11439	11439	0
133	Info	1/30/2014 2:04:54 PM	-----			
134	Info	1/30/2014 2:04:54 PM	Total HTTP Entity Bytes:	21448125		
135	Info	1/30/2014 2:04:54 PM	- Sent HTTP Entity Bytes:	21448125		

Transfer Encoding

Typical HTTP sending or receiving of information requires that the length of the information that is to be sent or received is known in advance and specified in the Content-Length Header. In the case of dynamically generated or streamed information, the length of what is being sent or received may not be known at send time. To address that possibility, Chunked Transfer Encoding is supported. Chunked Transfer Encoding is enabled in the Load DynamiX HTTP client content-capable Actions (**GET, POST, PUT, DELETE, HEAD, OPTIONS**) by setting the Transfer-Encoding Chunked input to **True** and specifying a Chunk Size greater than 0. The maximum Chunk Size that the Load DynamiX Appliance supports is 100KB (102400 bytes). If a value greater than 100KB is in the Chunk Size field, the Appliance Firmware will truncate the value to 100KB.

When Chunked input is set to **True**, a Transfer-Encoding Header (Transfer-Encoding:chunked) is added to the Request Headers when the content-capable Action is executed.



Independent of the Content-Encoding settings, the Transfer-Encoding header can also specify that the Chunked Content is compressed using either gzip or deflate. In the screenshot above, the Chunked Content is compressed using gzip. In this case, Transfer-Encoding Header (Transfer-Encoding: gzip,chunked) is added to the Request Headers when the content-capable Action is executed.

When Transfer Encoding is used in a Load DynamiX Client, HTTP Transfer Encoding graphs and Log file information is generated.

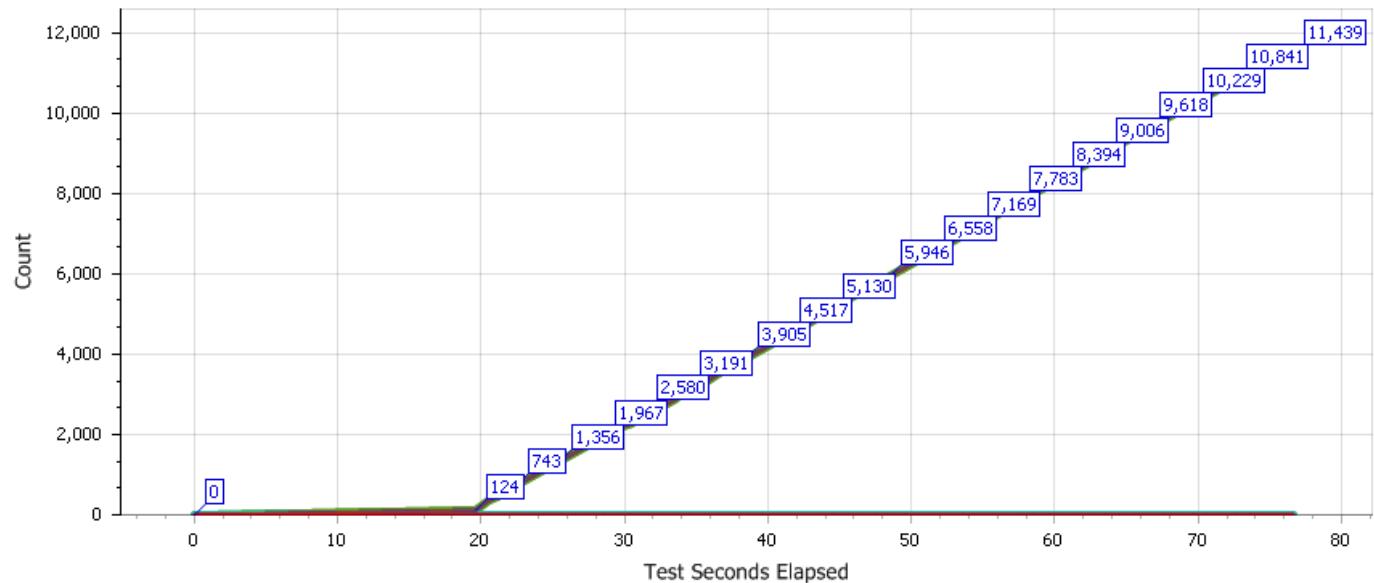
1/30/2014 2:10:52 PM - Clients Total: HTTP Transfer-Encoding



Chunked ▾ Messages ▾ Total ▾



View ▾

Clients Total: HTTP Transfer-Encoding (Chunked, Messages, Total)

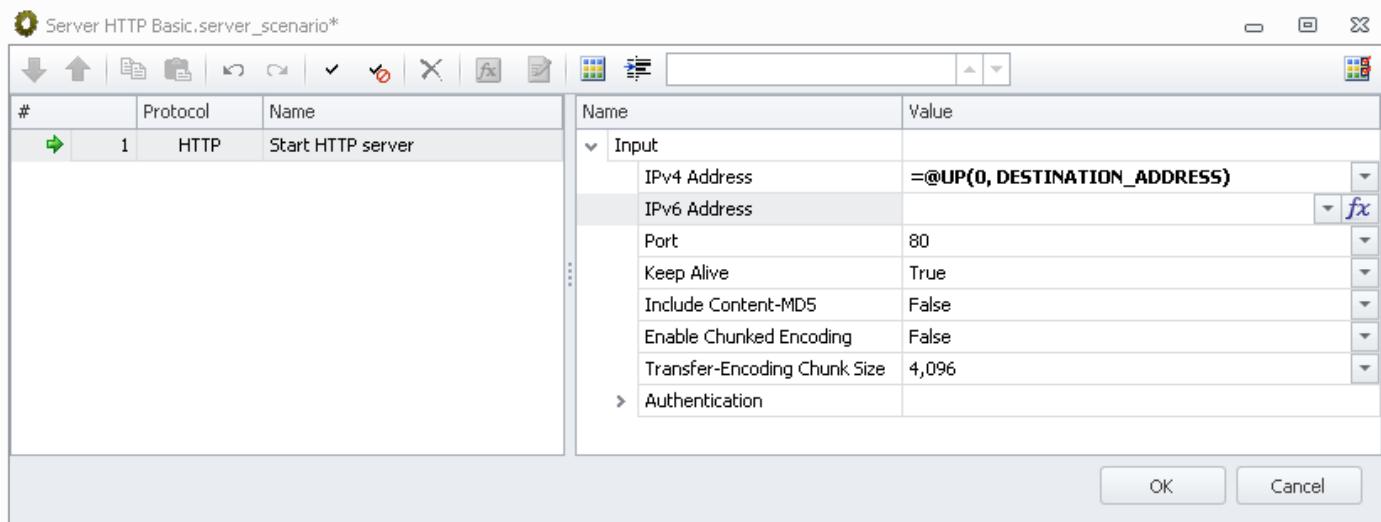
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Name	Last Poll
▶	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Chunked Total Messages Attempted	11,439
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Chunked Total Messages Succeeded	11,439
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Chunked Total Messages Failed	0
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Chunked Total Messages Aborted	0
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Chunked Tx Messages Attempted	11,439
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Chunked Tx Messages Succeeded	11,439
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Chunked Tx Messages Failed	0
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Chunked Tx Messages Aborted	0

Debug Info Warning Error Exception Status All

Line	Type	Date / Time	Text				
				Attempted	Succeeded	Failed	Aborted
104	Info	1/30/2014 2:04:54 PM	=====				
105	Info	1/30/2014 2:04:54 PM	HTTP deflate Content-Encoding Statistics	Attempted	Succeeded	Failed	Aborted
106	Info	1/30/2014 2:04:54 PM	=====				
107	Info	1/30/2014 2:04:54 PM	Total:	11439	11439	0	0
108	Info	1/30/2014 2:04:54 PM	Received entities with Content-Encoding:	11439	11439	0	0
109	Info	1/30/2014 2:04:54 PM	=====				
110	Info	1/30/2014 2:04:54 PM	Total Bytes Encoded:	21333735			
111	Info	1/30/2014 2:04:54 PM	- Received Bytes Encoded:	21333735			
112	Info	1/30/2014 2:04:54 PM	Total Bytes Decoded:	4685414400			
113	Info	1/30/2014 2:04:54 PM	- Received Bytes Decoded:	4685414400			
114	Info	1/30/2014 2:04:54 PM	=====				
115	Info	1/30/2014 2:04:54 PM	=====				
116	Info	1/30/2014 2:04:54 PM	=====				
117	Info	1/30/2014 2:04:54 PM	HTTP chunked Transfer-Encoding Statistic	Attempted	Succeeded	Failed	Aborted
118	Info	1/30/2014 2:04:54 PM	=====				
119	Info	1/30/2014 2:04:54 PM	Total:	11439	11439	0	0
120	Info	1/30/2014 2:04:54 PM	Sent messages with Transfer-Encoding:	11439	11439	0	0
121	Info	1/30/2014 2:04:54 PM	=====				
122	Info	1/30/2014 2:04:54 PM	Total HTTP Entity Bytes:	1429875			
123	Info	1/30/2014 2:04:54 PM	- Sent HTTP Entity Bytes:	1429875			
124	Info	1/30/2014 2:04:54 PM	Total HTTP Message Bytes:	1555704			
125	Info	1/30/2014 2:04:54 PM	- Sent HTTP Message Bytes:	1555704			
126	Info	1/30/2014 2:04:54 PM	=====				
127	Info	1/30/2014 2:04:54 PM	=====				
128	Info	1/30/2014 2:04:54 PM	=====				
129	Info	1/30/2014 2:04:54 PM	HTTP gzip Transfer-Encoding Statistics	Attempted	Succeeded	Failed	Aborted
130	Info	1/30/2014 2:04:54 PM	=====				
131	Info	1/30/2014 2:04:54 PM	Total:	11439	11439	0	0
132	Info	1/30/2014 2:04:54 PM	Sent messages with Transfer-Encoding:	11439	11439	0	0
133	Info	1/30/2014 2:04:54 PM	=====				
134	Info	1/30/2014 2:04:54 PM	Total HTTP Entity Bytes:	21448125			
135	Info	1/30/2014 2:04:54 PM	- Sent HTTP Entity Bytes:	21448125			

HTTP Server Content and Transfer Encoding

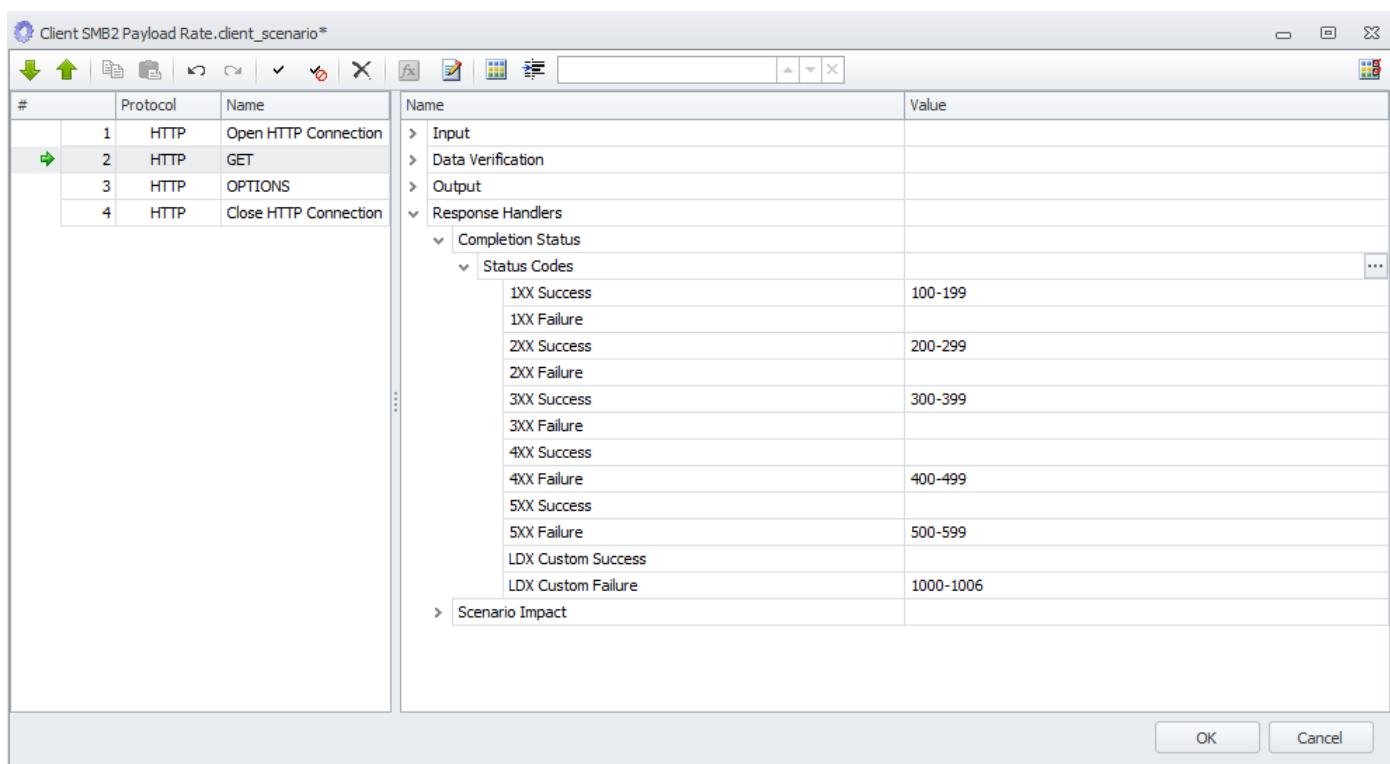
The Load DynamiX HTTP Server is capable of supporting gzip and deflate compression/decompression and Transfer Encoding Chunking. The only Transfer Encoding inputs that the HTTP server takes are an Enable Chunked Encoding input (**True or False**) and a Transfer-Encoding Chunk Size input (integer). Chunked Transfer Encoding is used by the server when Enable Chunked Encoding is set to **True** (for all responses), or when the HTTP client requests the server to enable it via “TE” header (on per-request basis)



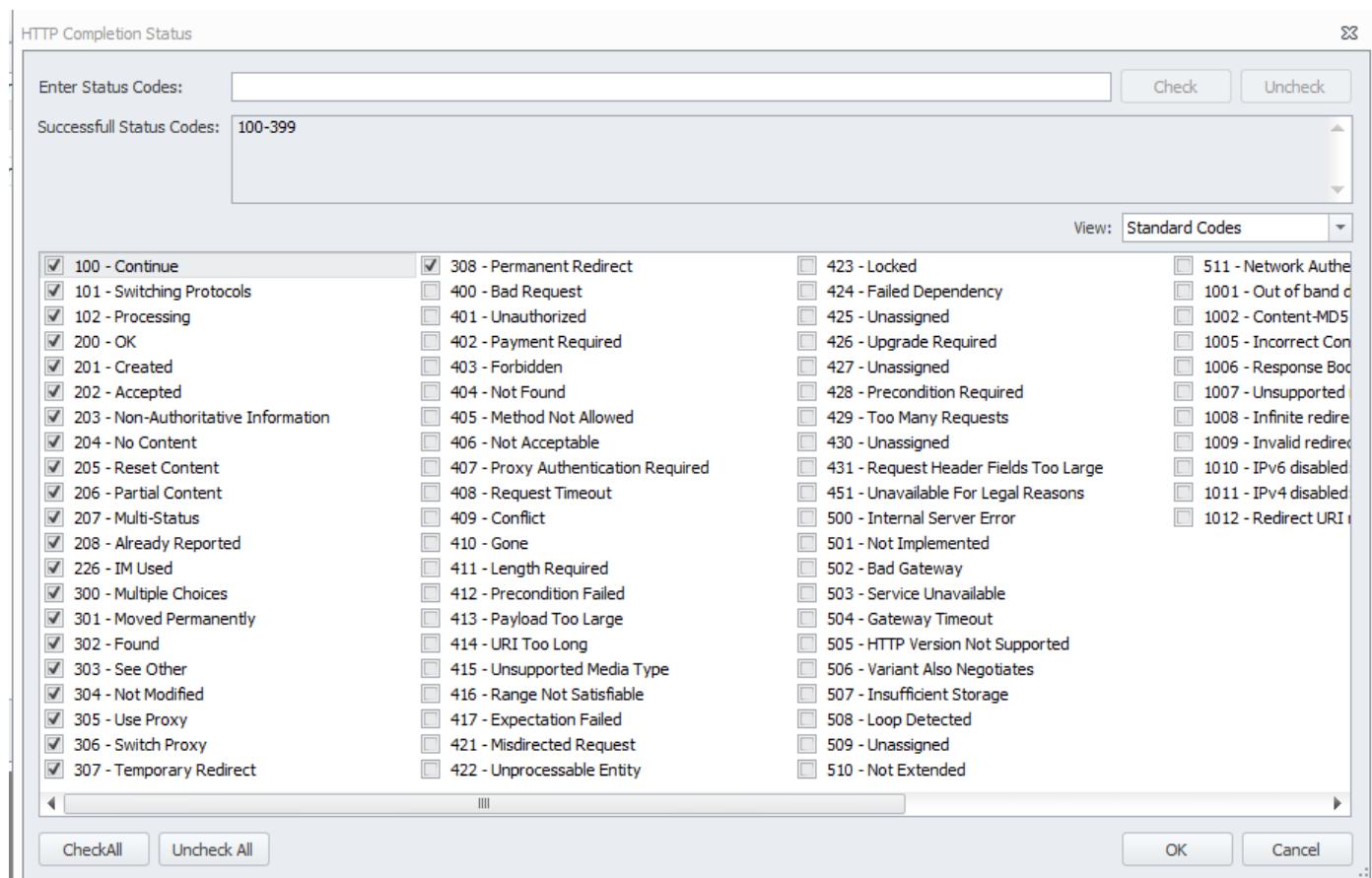
Like the Client, the Server also generates graphs for Content Encoding and Transfer Encoding when they are present and Log file entries for both.

HTTP (HTTP, HTTPS, HTTP Storage) Response Handling

The HTTP and HTTP-based protocol Actions (HTTP, HTTPS, OpenStack Swift, OpenStack Cinder, Keystone, CDMI and Amazon S3) define the HTTP status codes Informational (100-199), Success (200-299), Redirection (300-399) as a Success indication and the Client Error (400-499), Server Error(500-599) and LDX Custom (1000-1006) status codes as a Failure indication by default.

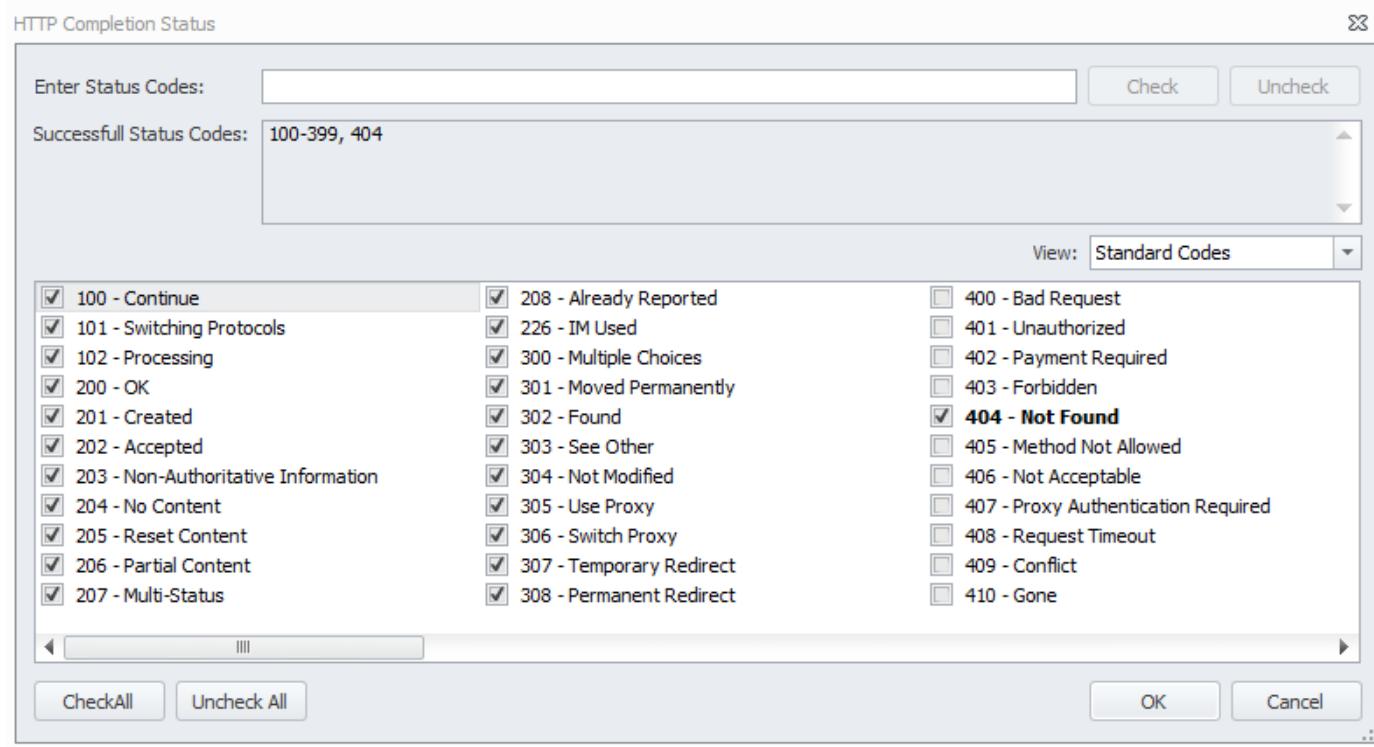


The Tester can, using the HTTP Status Code editor, change the interpretation of any HTTP status code from Success to Failure (or Failure to Success) on a per Action basis. Click the on the Status Codes line to open the HTTP Status Code Editor.

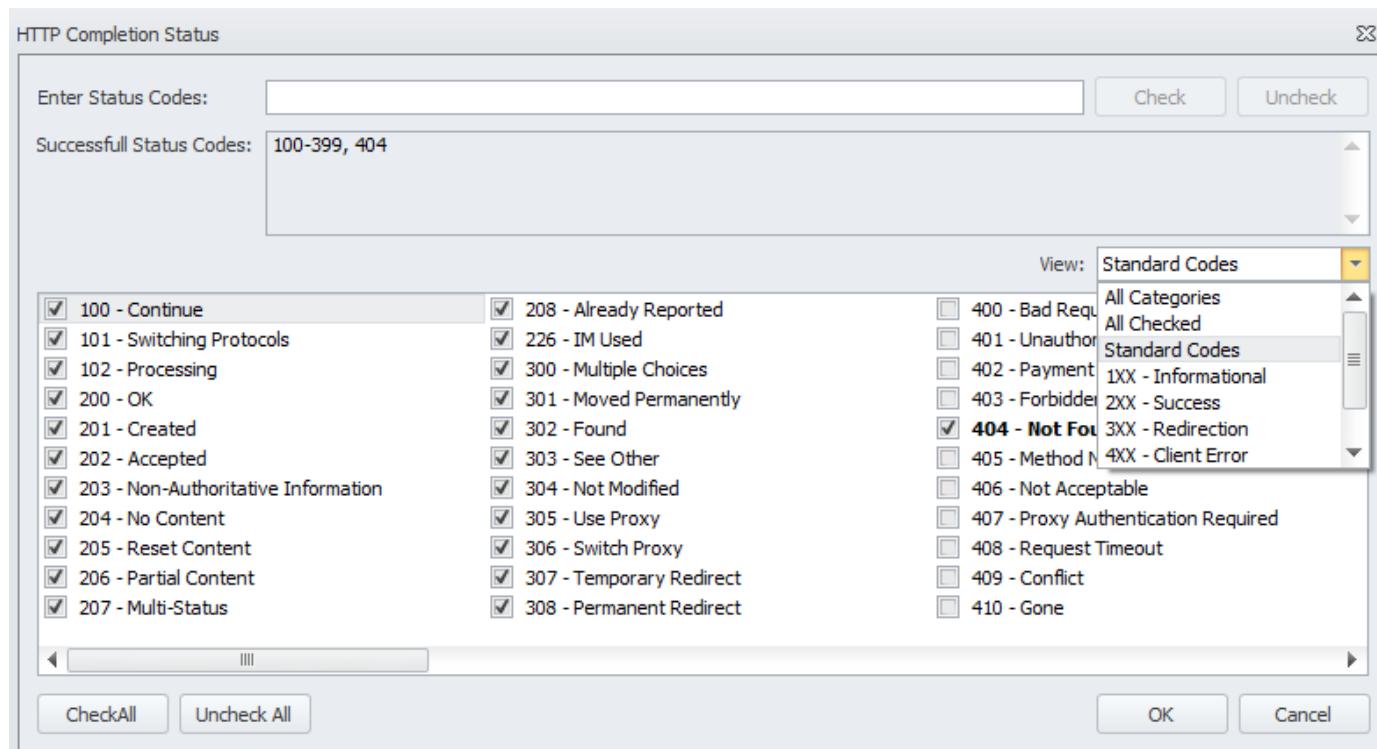


The status codes with the check mark to the left are the codes that are currently selected as a Success indication for a given Action. To add a status code (or codes) to the list of success

indicators, simply click the empty box to the left of the return code . Likewise, to indicate that a status code is now a Failure indication, clear the check box by clicking on the check mark. After all changes are made, click OK to complete the process.



The Tester can change which return codes are displayed by changing the "View". To change the View, select the View desired by clicking the in the View input field.



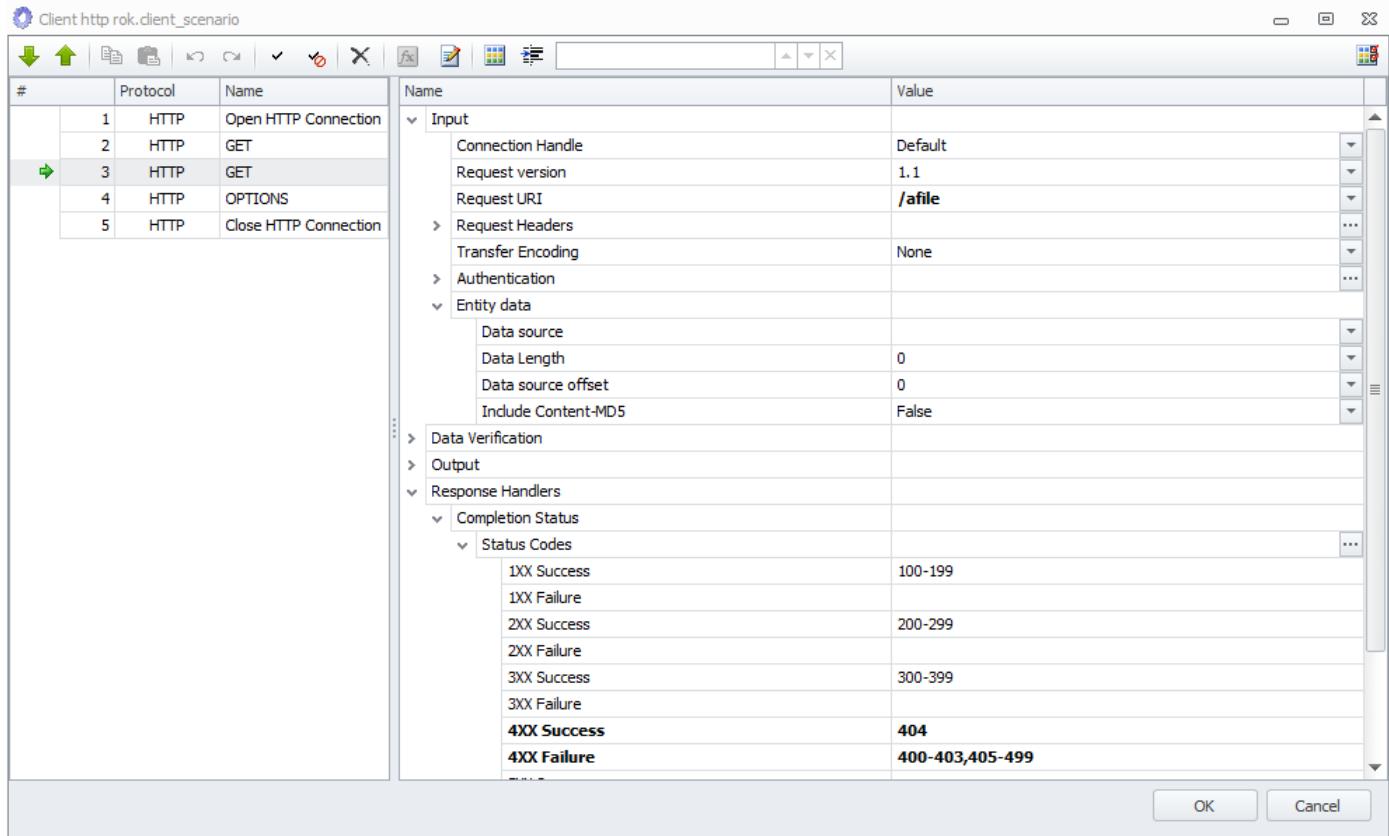
The Tester can choose

View Name	Meaning
All Categories	All return code values from 100 to 1012
All Checked	Just those return codes that have been Checked
Standard Codes	The default set of return codes
1XX - Informational	Return codes 100 - 199 all checked
2XX - Success	Return codes 200 - 299 all checked
3XX - Redirection	Return codes 300 - 399 all checked
4XX - Client Error	Return codes 400 - 499 NONE checked
5XX - Server Error	Return codes 500 - 599 NONE checked
LDX Custom	Return codes 1000 - 1012 NONE checked

The Tester can use the Enter Status Code

Example

In the following HTTP, the second **GET** Action attempts to get a file named "/afile" that does not exist. The result of the operation is a 404 status code which is desired so the 404 status code is designated for this one Action as a Success.



When this scenario is executed, it produces the following PCAP result which shows the **GET** of "/afile" returning a 404 and the Scenario continuing to execute because 404 had been added to the Success indicators..

Protocol	Length	Info
ARP	60	Who has 172.16.244.1? Tell 172.16.240.1
ARP	60	172.16.244.1 is at 00:15:17:cc:f8:2c
TCP	62	2000→80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256
TCP	62	80→2000 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256
HTTP	249	GET / HTTP/1.1
HTTP	311	HTTP/1.1 200 Ok (application/binary)
HTTP	254	GET /afile HTTP/1.1
HTTP	312	HTTP/1.1 404 Not Found (text/html)
TCP	60	2000→80 [FIN, ACK] Seq=396 Ack=516 Win=16776960 Len=0
TCP	60	80→2000 [FIN, ACK] Seq=516 Ack=397 Win=16776960 Len=0
TCP	60	2000→80 [ACK] Seq=397 Ack=517 Win=16776960 Len=0
TCP	62	2001→80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256
TCP	62	80→2001 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256
HTTP	253	OPTIONS / HTTP/1.1
HTTP	156	HTTP/1.1 200 Ok
TCP	60	2001→80 [FIN, ACK] Seq=200 Ack=103 Win=16776960 Len=0
TCP	60	80→2001 [FIN, ACK] Seq=103 Ack=201 Win=16776960 Len=0
TCP	60	2001→80 [ACK] Seq=201 Ack=104 Win=16776960 Len=0

The above example shows the HTTP protocol in action but the concepts apply to all HTTP-based Protocol Actions (HTTP, HTTPS, OpenStack Swift, OpenStack Cinder, Keystone, CDMI and Amazon S3).

HTTP/HTTPS Pipelining (Asynchronous I/O)

The HTTP/HTTPS and HTTP Storage (CDMI, OpenStack Cinder, Swift and Amazon S3) protocols support the Asynchronous execution of Actions. See [Advanced Concepts: Threads and Async Operations](#) for more details. The Load DynamiX HTTP and HTTPS Servers accept pipelined HTTP/HTTPS Actions.

HTTP/HTTPS Threads

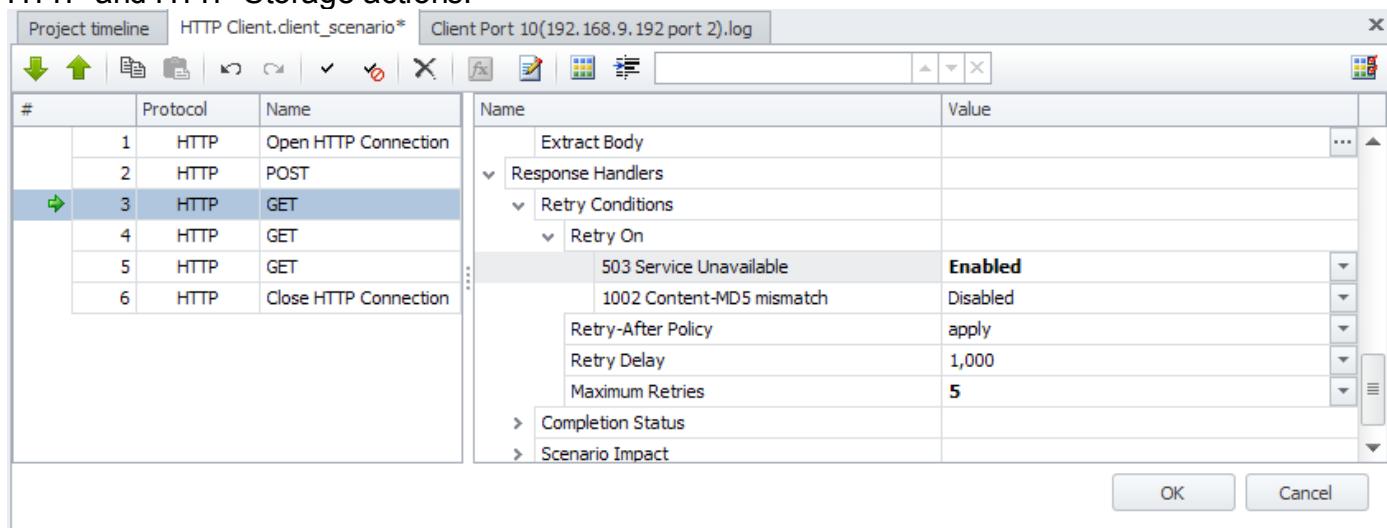
The HTTP/HTTPS and HTTP Storage (CDMI, OpenStack Cinder, Swift and Amazon S3) protocols support Threads. See [Advanced Concepts: Threads and Async Operations](#) for more details.

HTTP Retries

HTTP and HTTP Storage command retries are captured as part of the standard set of HTTP/HTTP Storage statistics. The Load DynamiX HTTP Firmware counts and accounts for HTTP and HTTP Storage command retries in the statistics and counters that are captured during the execution HTTP/S and HTTP Storage Projects. LDX HTTP and HTTP Storage directives support retries on 503 HTTP stats code and "Content-MD5 mismatch" LDX defined custom code. Additionally the following six OpenStack Cinder actions support retries on "Response Body Condition" status:

- Volume Show
- Volume List
- Snapshot Show
- Snapshot List
- Backup Show
- Backup List

Retries are disabled by default and are enabled and controlled by the group of parameters of each HTTP and HTTP Storage actions:



The group contains the following parameters:

Parameter	Values	Default value	Notes
Retry On	group which allows to enable code (for example 503) on which retries initiated		
Retry-After Policy	"apply" or "ignore"	apply	visible only when "Retry On 503" is enabled
Retry Delay	1 ... 86,400,000 msec (1 msec ... 1 day)	1000	
Maximum Retries	unsigned integer	0	retries are disabled by default

To enable retries customer has to set "Enabled" for the corresponding code of the "Retry On" group and "Maximum Retries" > 0. In this case when 503 status code is received or LDX defined "Content-MD5 mismatch" is met and this code is adjusted as failed by Completion Status, client doesn't move to the next action, but waits for the "retry alert" and resends the request body. "Retry alert" is determined as "in 'Retry Delay' msec" for "Content-MD5 mismatch" and the following way for 503:

Retry-After Policy	Retry-After response header	"Retry alert"
apply	is absent or has invalid value	in "Retry Delay" msec
apply	apply is present and has valid value	the time from Retry-After response header
ignore	doesn't make sense	in "Retry Delay" msec

When retries are already made "Maximum Retries" times action fails with "Retry Limit Reached" LDX defined code.

When "Maximum Retries" is set to zero the retries are disabled and each 503 and/or "Content-MD5 mismatch" response is treated as "Retry Limit Reached".

Reference: iSCSI Commands and Behaviors

Reference: Load DynamiX iSCSI Commands and Behaviors

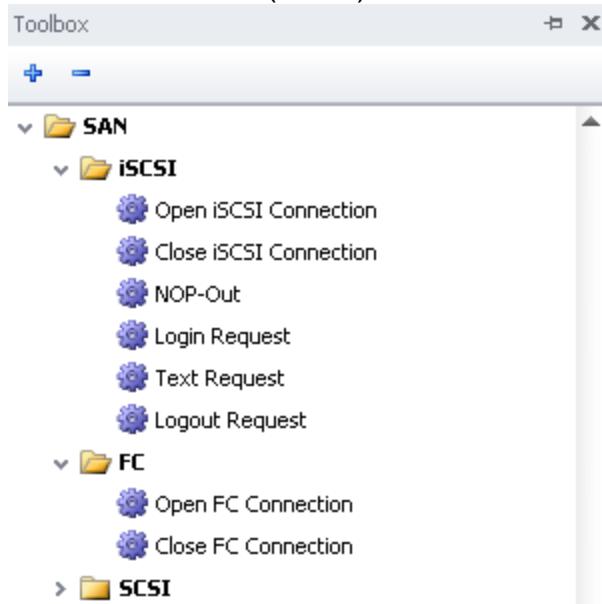
ACTION
Close iSCSI connection
Open iSCSI connection
Login Request
Logout Request
NOP-Out
Text Request

A link to iSCSI protocol reference material is provided in the [References and Terminology section](#).

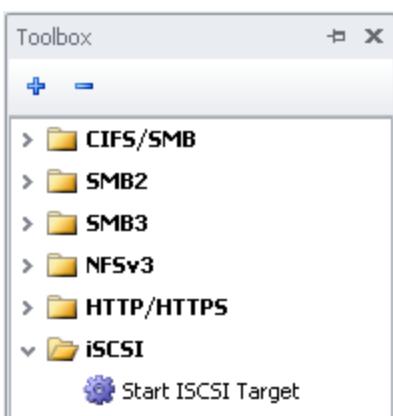
iSCSI

The SCSI protocol was designed to allow systems to communicate with I/O devices, especially storage devices. The SCSI protocol is a request/response application protocol that supports a standard architecture model and basic command set as well as standardized command sets for different types of I/O devices (disks, tapes, media changers, etc). The iSCSI protocol allows the standard SCSI commands to be delivered via TCP to devices that are directly connected to an IP network. The iSCSI/SCSI protocol operates at a different level than the other storage related protocols (SMB and NFS) that the Load DynamiX product supports in that it is a device-oriented protocol and SMB/NFS are filesystems-oriented protocols. iSCSI Clients are called Initiators and iSCSI Servers are called Targets.

The iSCSI Initiator (Client) Actions that are supported by the Load DynamiX TDE and Appliance are:



The iSCSI Target (Server) Actions supported are:



The iSCSI target is instantiated in a Server Scenario using the **Start iSCSI Target** Action and providing, at a minimum, an IP address for the Server.

An example iSCSI Client (Initiator) Scenario that will login and then read LUNs in a loop would look like:

The screenshot shows the 'Scenario Editor' window for a client scenario named 'Client iSCSI LUN Read.client_scenario'. The left pane displays a sequence of steps:

#	Protocol	Name
1	iSCSI	Open iSCSI Connection
2	iSCSI	Login Request
3	SCSI	Report LUNs
4	SCSI	Read Capacity(10)
5	SCSI	Read Capacity(10)
6	SWT	Begin Loop
7	SCSI	LUN Read
8	SWT	End Loop
9	iSCSI	Logout Request

The right pane shows configuration settings for the 'Open iSCSI Connection' step, specifically for the 'Input' section:

Name	Value
Destination Address	172.16.244.1
Destination Port	3260
Keep Alive	False

Below the 'Input' section, there are sections for 'Reconnect' and 'Reconnect Conditions'.

The iSCSI Discovery Sample test that is delivered with the Load DynamiX TDE is an example of how the Discovery capability of the iSCSI protocol can be used to detect all of the logical units (LUNs) supported by a given iSCSI Target device.

The screenshot shows a software interface for defining iSCSI scenarios. On the left, a table lists four actions:

#	Protocol	Name
1	iSCSI	Open iSCSI Connection
2	iSCSI	Login Request
3	iSCSI	Text Request
4	iSCSI	Logout Request

On the right, detailed configuration for the "Login Request" action is shown in a tree-view table:

Name	Value
Input	
Connection Handle	Default
ISIDT	1
ISIDA	0
ISIDB	=@RANDOM(0,999999,1)
ISIDC	=@RANDOM(0,999999,1)
ISIDD	=@RANDOM(0,999999,1)
TSIH	New
Response Timeout	30,000
Text Keys	
Initiator Name	=@STRING(iqn.2009.09.com.SWFTT:snc) + @RANDOM(0,999999,1)
Session Type	Discovery
Target	
Target Name	iqn.1992-08.com.swift:sn.1234560
AuthMethod	
CHAP	False
None	True
Enable Redirect	True

The **iSCSI Login Request** Action contains an input field named Session Type that has two possible settings: Normal or Discovery. If the scenario being developed is doing Discovery (e.g. using the **Text Request** Action to collect target IQN information), then the Session Type must be set to Discovery, otherwise the **Text Request** Action will fail. If the Scenario is doing Read or Write I/O then the Login Action Session Type must be set to Normal.

iSCSI/SCSI Features: TCP Reconnect, Redirect, Data Verification, Asynchronous Read/Write, VAAI (VMware vSphere Storage API Array Integration), 2-Way Chap Authentication

SCSI Asynchronous Read and Write Operations

SCSI Asynchronous I/O supports the Number of Outstanding Requests feature of the Load DynamiX NFSv3 Protocol and the Async and Threads blocks of the Load DynamiX CIFS-SMB and SMB2 Protocols.

Number of Outstanding Requests

SCSI LUN Read and **LUN Write** operations can be segmented into multiple operations that are issued asynchronously. To do this use the Number of Outstanding Requests feature of the iSCSI LUN Read and LUN Write Actions. Below are two screen shots showing the **LUN Read** and **LUN Write** Actions with the Number of Outstanding Requests field. The allowed range of values for the Number of Outstanding Requests field is 1 to 128. Values < 1 or > 128 will result in an error that will prevent saving the Project.

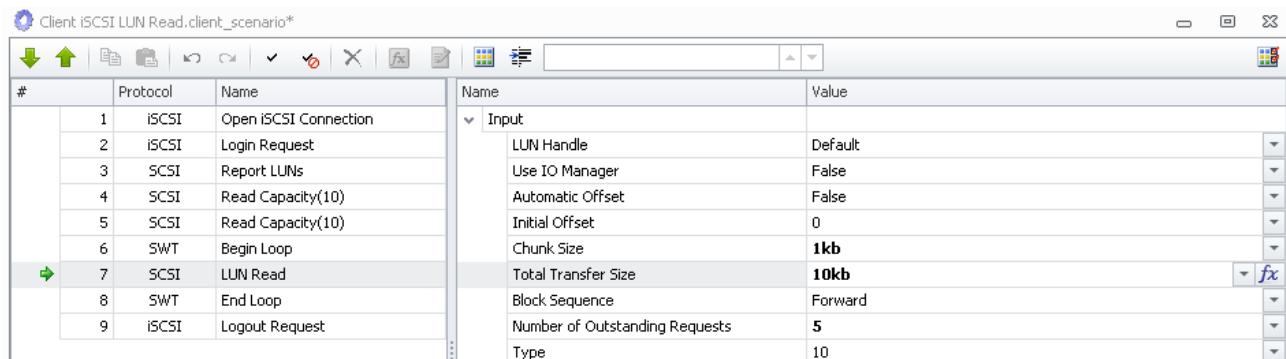
The default value of Number of Outstanding Requests is 1 which results in same behavior as before Number of Outstanding Requests was introduced. The generic behavior of a LUN Read and LUN Write Actions with Number of Outstanding Requests > 1 is that the operation (LUN Read or LUN Write) is broken into up to N independent operations of size Number of Bytes per Request where

N == (Number of Bytes/Number of Bytes per Request) or Number of Outstanding Requests,

whichever is smaller and these N operations are all issued at the same time. As these requests complete, new operations are issued until the total number represented by (Number

of Bytes/Number of Bytes per Request) has been completed. Any errors encountered during the processing of these requests will cause the entire Action to fail.

In the **LUN Read** Action below, Chunk Size == 1kb(1024), Total Transfer Size == 10kb(10240) so the result of the division is 10. Number of Outstanding Requests is 5 so N == 5 (the smaller of 5 and 10) and initially 5 requests will be sent in parallel to the target SCSI server and as any one of these 5 requests completes, the remaining requests will be issued.



iSCSI LUN Read and **LUN Write** Actions can also be executed in Thread Blocks or as Async Operations:

Thread Blocks

The Actions in Load DynamiX Scenarios execute sequentially until completion which generally means that the Action in Row N executes and completes before the Action in Row N+1 is executed. If it is necessary, collections of Actions can be executed in parallel in Sub-Scenarios. The Begin Thread...End Thread Actions allow the Tester to specify which Actions are executed as a Sub-Scenario.

Conceptually, a Thread is a separate process that executes independently of the rest of the Scenario. See [Advanced Concepts: Threads and Async Operations section](#) for details.

Async Operations

If the Tester needs to have individual Actions executed in Parallel, then Asynchronous Operations can be used. To identify a set of Actions that are to be executed in parallel, use the Begin Async...End Async Actions around the CIFS-SMB or SMB2 Actions that are to be executed in Parallel. Begin Async has only one input value - Wait For Completion with a default value of True which means that the Async collection of Actions will wait until all of the Actions have completed before the next Action is executed. If Wait For Completion is set to False then the Actions in the collection will be launched and Scenario execution will continue. In the Scenario below, the first Thread does all of the writing and the second Thread does all of the reading. Events are used to synchronize execution between the two threads and between the second Thread and the Tree Disconnect (we do not want to disconnect from the Share before the Reads are done). See [Advanced Concepts: Threads and Async Operations section](#) for details.

The difference between Number of Outstanding Operations and Thread Blocks and Async Operations is as follows:

NOR as described above allows the user to set the limits on how many concurrent operations are executed.

In Async Operations every Action inside the Begin Async and End Async pair gets executed concurrently.

In Thread Blocks, the Actions are executed sequentially but the entire block of Actions is executed in parallel with the rest of the Actions in the Scenario.

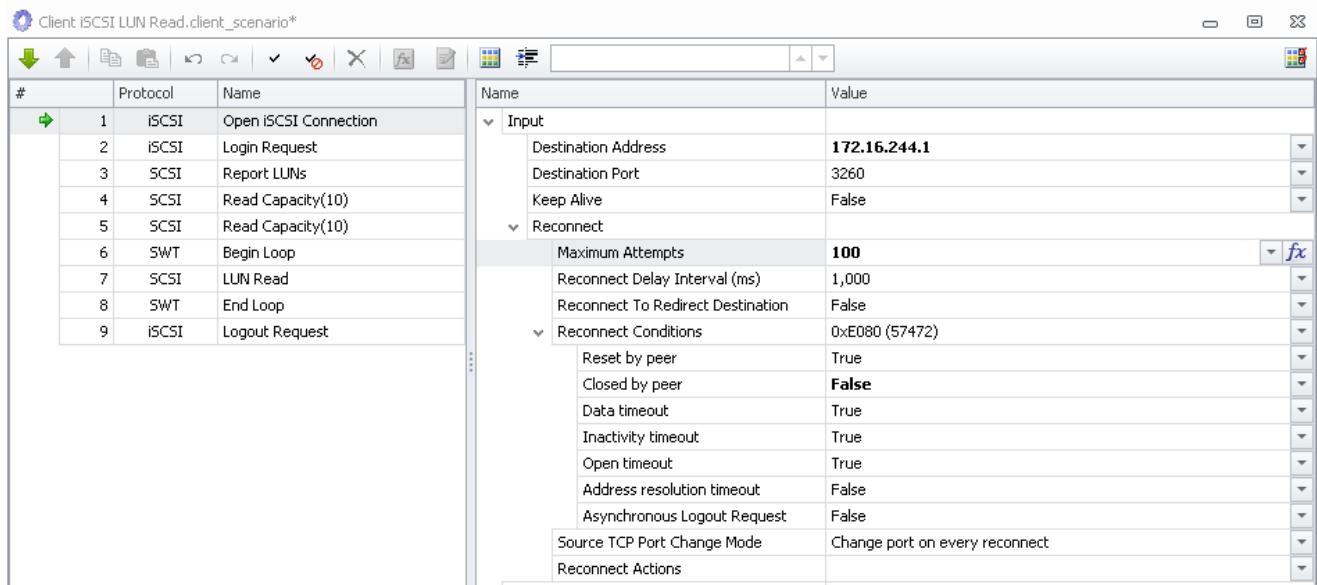
Redirect

iSCSI TCP Redirect behaviors

- In prior Load DynamiX software revisions, if an iSCSI TCP connection to target A was redirected to an iSCSI TCP connection to target B, and the connection to target A was terminated by target A after the connection to target B entered TCP handshake but before it exited TCP handshake, the system may crash. This revision fixes this issue.
- In prior Load DynamiX software revisions, if an iSCSI TCP connection to target A was redirected to an iSCSI TCP connection to target B, and opening the connection to target B failed (due to address resolution, or connection-open timeout), the connection to target A was not checked to see if it was closed by target A in context of iSCSI TCP Redirect. This revision fixes the issue by verifying if the connection to target A has been closed by target A itself while connection to target B was still in TCP handshake, and, if not, by closing it upon exiting the handshake for connection to target B. The previous revisions did not include the case of TCP handshake failing for the connection to target B.

TCP Reconnect

The Load DynamiX iSCSI Scenarios support iSCSI Reconnect. When an iSCSI server disconnects from a running Scenario for any of seven reasons, the Scenario can attempt to reconnect to the server using a series of Actions that are specified in the iSCSI Open TCP Connection Action.



The user is allowed to specify

Set Reconnect parameters:

- Maximum Attempts - how many times to try to reconnect per disconnect
- Reconnect Delay Interval (in milliseconds) - how long to delay before the first and in-between attempted reconnects.
- Reconnect to Redirect Destination: If the iSCSI open connection request was redirected to a different IP Address, if **True** use the IP Address that the connection was redirected to. If **False** (default behavior), use the IP Address in the iSCSI Open

TCP Connection Action.

- Reconnect Conditions: A list of Conditions that should (True) or should not (False) cause Reconnect processing.
- Source TCP Port Change Mode: Change port on every reconnect = use a new Port # with every reconnect, Change port on first reconnect = use a new Port # just on the first reconnect, Do not change port on reconnect = do not use a new Port # on reconnect.
- Reconnect Actions

Set the conditions under which Reconnect processing is to be executed:

- Reset by peer (TCP RST sent by the iSCSI Target) - True or False
- Closed by peer (TCP FIN was sent by the iSCSI Target) - True or False
- Data timeout (no ACK or data returned during a Read/Write operation) - True or False
- Inactivity timeout (inactivity timeout period specified in the Network Profile expired) - True or False
- Open timeout (ARP found server MAC address but Open of ISCI port fails) - True or False
- Address resolution timeout (ARP did not find MAC address of the server IP address) - True or False
- Asynchronous Logout Request (Logout request sent by the iSCSI Target to the iSCSI Client) - True or False

Set the Actions to be executed when a Reconnect occurs::

In the example above, the Actions List that has been selected to be executed on reconnect are:

<input type="checkbox"/> Action Name
<input checked="" type="checkbox"/> 2 Login Request
<input checked="" type="checkbox"/> 3 Report Luns
<input checked="" type="checkbox"/> 4 Read Capacity(10)
<input type="checkbox"/> 5 Read Capacity(10)
<input type="checkbox"/> 9 Logout Request

Set Reconnect parameters:

- Maximum Attempts - how many times to try to. reconnect per disconnect
- Reconnect Delay Interval (in milliseconds) - how long to delay before the first and in-between attempted reconnects.
- Port Change Mode: Change port on every reconnect (use a different Client port number on each attempt); Change port on first reconnect (only change the Client port number on the first reconnect attempt); Do not change port on reconnect (use Client port that was used on the original open connection)

Set the conditions under which Reconnect processing is to be executed:

- Reset by peer (TCP RST sent by the iSCSI Target) - True or False
- Closed by peer (TCP FIN was sent by the iSCSI Target) - True or False
- Data timeout (no ACK or data returned during a Read/Write operation) - True or False
- Inactivity timeout (inactivity timeout period specified in the Network Profile expired) - True or False
- Open timeout (ARP found server MAC address but Open of ISCI port fails) - True or False
- Address resolution timeout (ARP did not find MAC address of the server IP address) - True or False

Reconnect statistics

Log File:

Below are the TCP Reconnect statistics from Client log file:

Debug 5/2/2011 12:10:18 PM TCP reconnect statistics: Port 0
Debug 5/2/2011 12:10:18 PM

Debug 5/2/2011 12:10:18 PM Connection-level summary
Debug 5/2/2011 12:10:18 PM
Debug 5/2/2011 12:10:18 PM reconnects attempted: 1
Debug 5/2/2011 12:10:18 PM reconnects succeeded: 1
Debug 5/2/2011 12:10:18 PM reconnects failed: 0
Debug 5/2/2011 12:10:18 PM reconnects aborted: 0
Debug 5/2/2011 12:10:18 PM average reconnect success duration:
Debug 5/2/2011 12:10:18 PM -- excluding reconnect delay: 75027132
(microsec)
Debug 5/2/2011 12:10:18 PM -- including reconnect delay: 81029262 (microsec)
Debug 5/2/2011 12:10:18 PM minimum reconnect success duration:
Debug 5/2/2011 12:10:18 PM -- excluding reconnect delay: 75027132
(microsec)
Debug 5/2/2011 12:10:18 PM -- including reconnect delay: 81029262 (microsec)
Debug 5/2/2011 12:10:18 PM maximum reconnect success duration:
Debug 5/2/2011 12:10:18 PM -- excluding reconnect delay: 75027132
(microsec)
Debug 5/2/2011 12:10:18 PM -- including reconnect delay: 81029262 (microsec)
Debug 5/2/2011 12:10:18 PM

Debug 5/2/2011 12:10:18 PM Reconnect attempts details
Debug 5/2/2011 12:10:18 PM
Debug 5/2/2011 12:10:18 PM reconnects attempted: 6
Debug 5/2/2011 12:10:18 PM reconnects succeeded: 1
Debug 5/2/2011 12:10:18 PM reconnects failed: 5
Debug 5/2/2011 12:10:18 PM reconnects aborted: 0
Debug 5/2/2011 12:10:18 PM average reconnect success duration:
Debug 5/2/2011 12:10:18 PM -- excluding reconnect delay: 302 (microsec)
Debug 5/2/2011 12:10:18 PM -- including reconnect delay: 1000657 (microsec)
Debug 5/2/2011 12:10:18 PM minimum reconnect success duration:
Debug 5/2/2011 12:10:18 PM -- excluding reconnect delay: 302 (microsec)
Debug 5/2/2011 12:10:18 PM -- including reconnect delay: 1000657 (microsec)
Debug 5/2/2011 12:10:18 PM maximum reconnect success duration:
Debug 5/2/2011 12:10:18 PM -- excluding reconnect delay: 302 (microsec)
Debug 5/2/2011 12:10:18 PM -- including reconnect delay: 1000657 (microsec)
Debug 5/2/2011 12:10:18 PM average reconnect failure duration:
Debug 5/2/2011 12:10:18 PM -- excluding reconnect delay: 15005365
(microsec)
Debug 5/2/2011 12:10:18 PM -- including reconnect delay: 16005720 (microsec)
Debug 5/2/2011 12:10:18 PM minimum reconnect failure duration:
Debug 5/2/2011 12:10:18 PM -- excluding reconnect delay: 15005361
(microsec)
Debug 5/2/2011 12:10:18 PM -- including reconnect delay: 16005716 (microsec)
Debug 5/2/2011 12:10:18 PM maximum reconnect failure duration:
Debug 5/2/2011 12:10:18 PM -- excluding reconnect delay: 15005375
(microsec)

Debug 5/2/2011 12:10:18 PM -- including reconnect delay: 16005730 (microsec)

Reconnect attempts due to inactivity timeout

reconnects attempted: 1
reconnects succeeded: 0
reconnects failed: 1
reconnects aborted: 0

average reconnect failure duration:
-- excluding reconnect delay: 15005375

-- including reconnect delay: 16005730 (microsec)

minimum reconnect failure duration:
-- excluding reconnect delay: 15005375

-- including reconnect delay: 16005730 (microsec)

maximum reconnect failure duration:
-- excluding reconnect delay: 15005375

-- including reconnect delay: 16005730 (microsec)

Reconnect attempts due to open timeout

reconnects attempted: 5
reconnects succeeded: 1
reconnects failed: 4
reconnects aborted: 0

average reconnect success duration:
-- excluding reconnect delay: 302 (microsec)
-- including reconnect delay: 1000657 (microsec)

minimum reconnect success duration:
-- excluding reconnect delay: 302 (microsec)
-- including reconnect delay: 1000657 (microsec)

maximum reconnect success duration:
-- excluding reconnect delay: 302 (microsec)
-- including reconnect delay: 1000657 (microsec)

average reconnect failure duration:
-- excluding reconnect delay: 15005363

-- including reconnect delay: 16005718 (microsec)

minimum reconnect failure duration:
-- excluding reconnect delay: 15005361

-- including reconnect delay: 16005716 (microsec)

maximum reconnect failure duration:
-- excluding reconnect delay: 15005368

-- including reconnect delay: 16005723 (microsec)

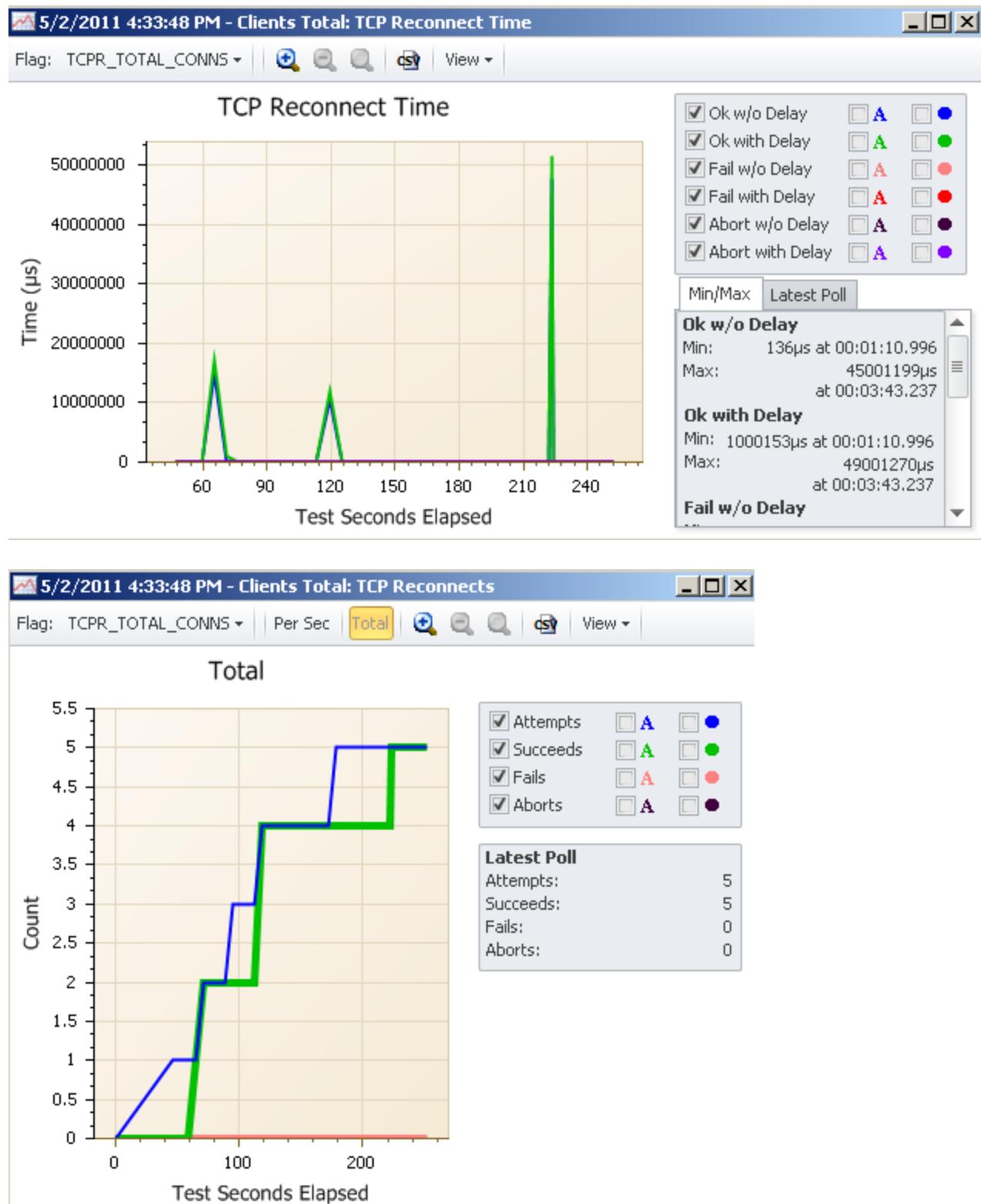
AUTOMATION:

TCP Reconnect statistics are available from Automated Projects. See the updated Automation Stats.XML file for the TCP Reconnect statistics that are supported. The location of

the Automation_Stats.XML file for your installation is defined in the Product Installation chapter.

GUI:

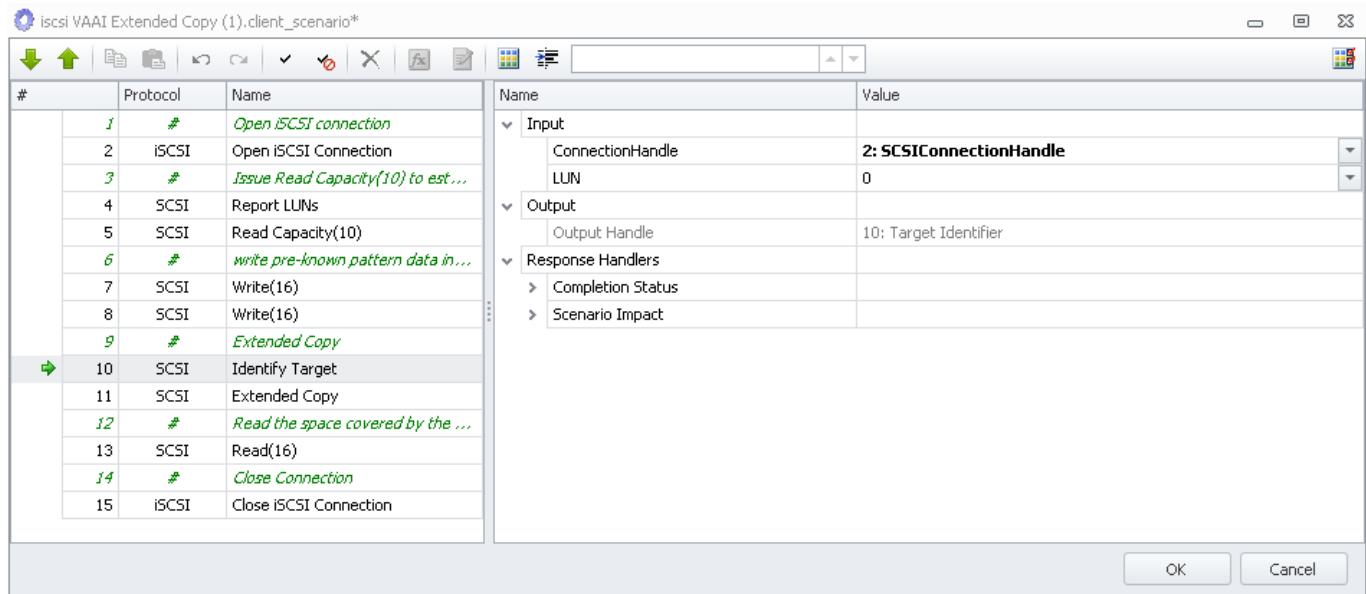
The TDE also provides a couple of real time graphs “TCP Reconnect Time” and “TCP Reconnects”



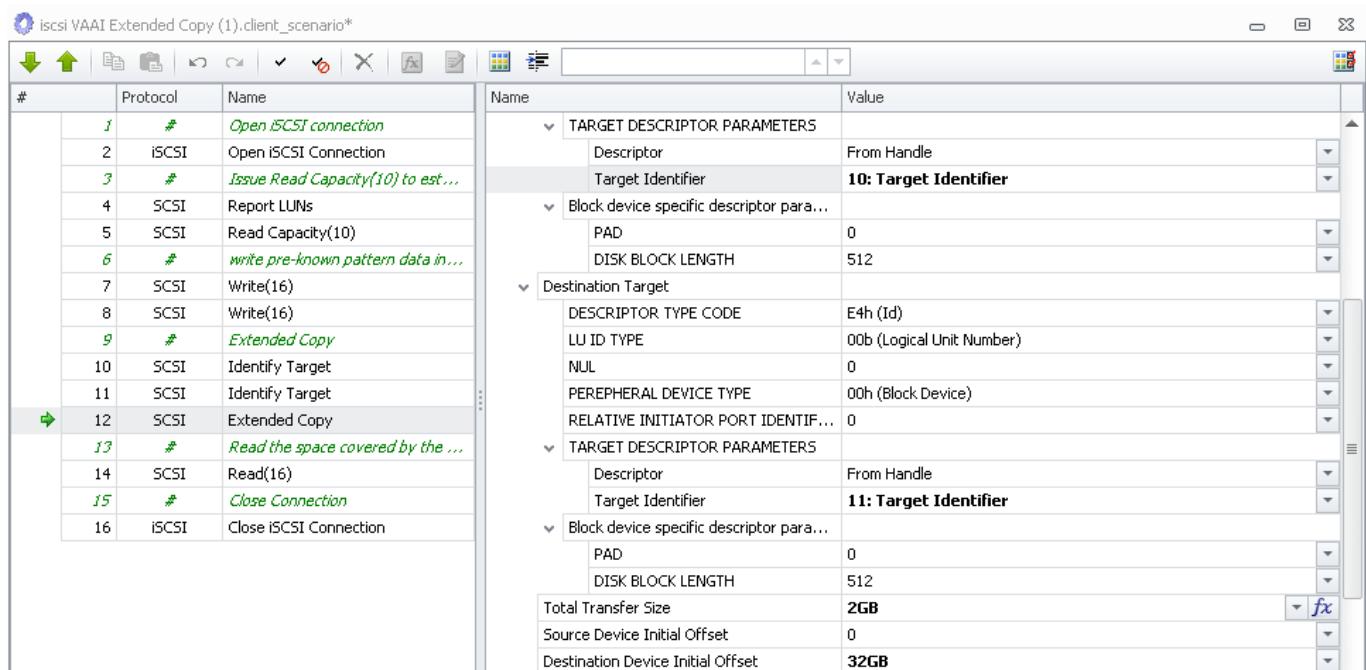
VAAI (VMware vSphere Storage API Array Integration)

Load DynamiX supports a set SCSI commands that are used by VMware's VSphere virtual machine management software to accelerate the creation, duplication and removal of virtual images. These commands are:

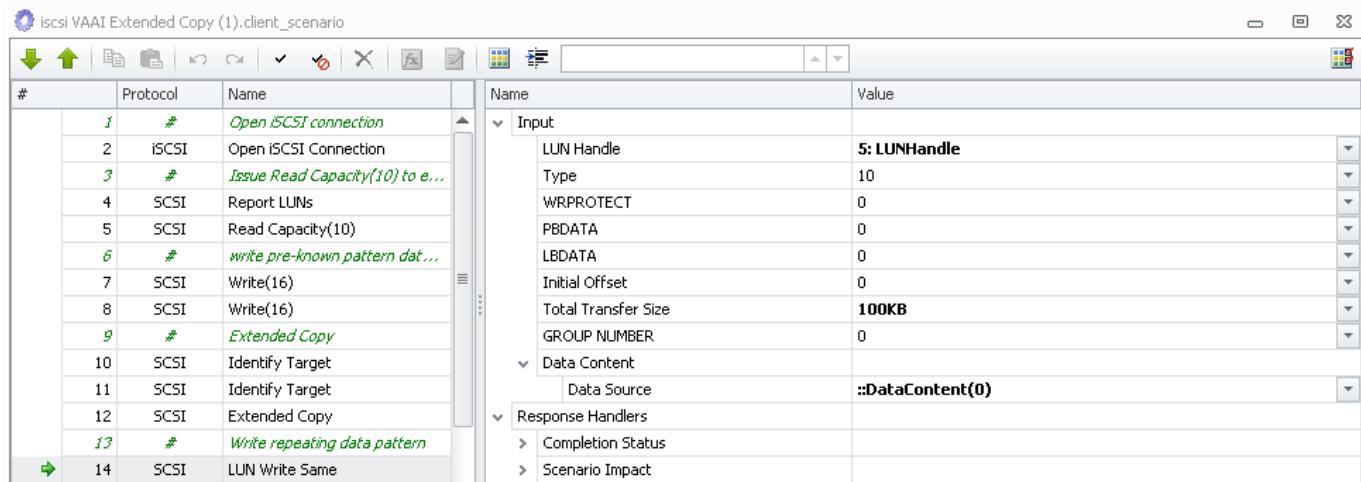
Identify Target (added to the Load DynamiX SCSI toolbox to make the use of Extended Copy easier) - provides a handle for a specific Device/LUN combination that can be used as input to the Source or Destination of an **Extended Copy** Action. Connection Handle comes from the **Open iSCSI Connection** Action and LUN identifies which logical unit on this device is to be acted upon.



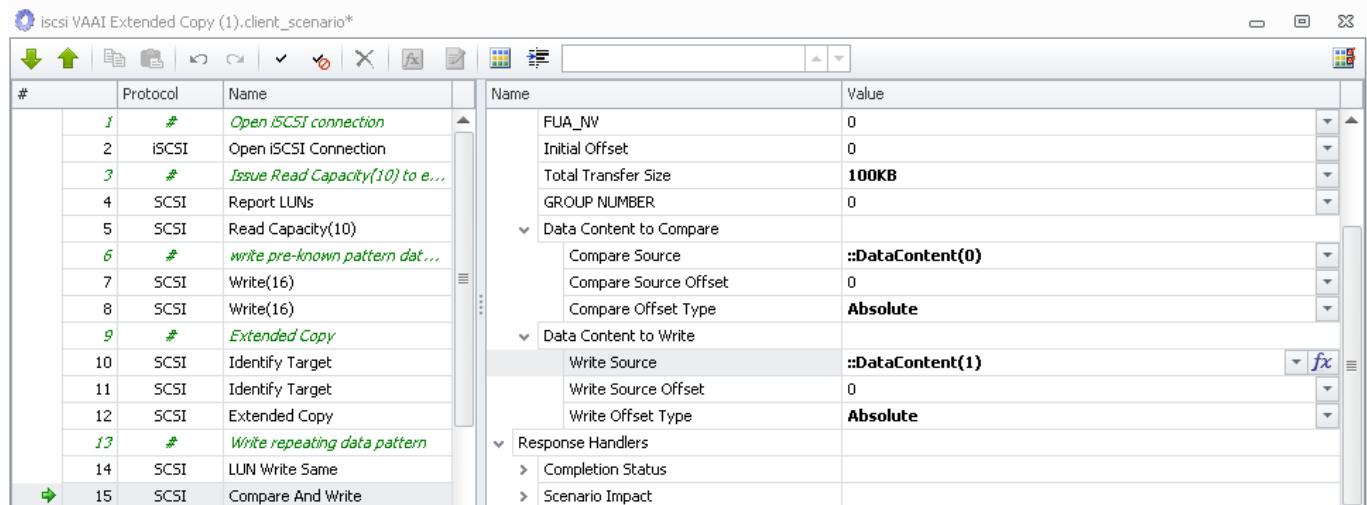
Extended Copy - within a single array, copy N blocks from a source LBA to a destination LBA. The output of Identify Target Action can be used as input to the Source or Destination Target. The Inputs Total Transfer Size, Source Device Initial Offset, Destination Device Initial Offset determine the number of bytes to copy and the source and destination byte offsets.



LUN Write Same (added to the Load Dynamix SCSI toolbox as a superset of the Write Same(10) and Write Same(16) commands) - used to write the same data pattern starting at a specific Initial Offset for some many bytes. LUN Handle is the output of the Read Capacity (10) Action. **Initial Offset** is the logical byte to begin the write at and **Total Transfer Size** is the count of bytes to write. The Data Content input provides the file containing the pattern to be written. The Type input field specifies if a Write Same (10) or Write Same (16) command format is to be used.



Compare and Write - an atomic allocate and write operation. Compare and Write compares a set of disc blocks to a known pattern (a lock) and, if they match, writes a set of blocks to the disc. LUN Handle is the output of the Read Capacity (10) Action. **Initial Offset** is the logical block to begin the write at and **Total Transfer Size** is the count of blocks to write. The Data Content input provides the file containing the data to be verified (Data Content to Compare) and the data to be written if the verify succeeds (Data Content to Write). Compare Offset Type allows the Tester to specify that Compare or Write Offsets are Relative (default) or Absolute.



Unmap - in a "Thin Provisioning" environment, return a set of disc blocks to the free list. LUN Handle is the output of the Read Capacity (10) Action. Initial Offset is the logical block address of the set of blocks to return to the free list and Total Unmap Size is the byte count.

Receive Copy Results - receive the results of an Extended Copy operation. LUN Handle is the output of the Read Capacity (10) Action.

1-WAY/2-WAY CHAP AUTHENTICATION

Load DynamiX supports 2-WAY CHAP Authentication as part of the iSCSI Login Action. Setting AuthMethod>CHAP == **True** enables a one-way CHAP authentication process requiring the Login Action to provide a CHAP Username and Password. (chptrg_u, chptrg_p) Setting 2-WAY CHAP == **True** enables 2-WAY CHAP Authentication and requires the Action to provide a second set of Username and Password inputs (chpint_u, chpint_p). During 2-WAY CHAP Authentication, the Initiator uses the CHAP Authentication Credentials in the first CHAP exchange and then uses the Initiator CHAP Authentication Credentials in the second CHAP exchange.

The Load DynamiX iSCSI Server Action does not support the 2-WAY CHAP Authentication protocol.

iscsi VAAI Extended Copy (1).client_scenario*

#	Protocol	Name
1	iSCSI	Open iSCSI connection
2	iSCSI	Open iSCSI Connection
3	iSCSI	Login Request
4	#	Issue Read Capacity(10) to e...
5	SCSI	Report LUNs
6	SCSI	Read Capacity(10)
7	#	write pre-known pattern dat...
8	SCSI	Write(16)
9	SCSI	Write(16)
10	#	Extended Copy
11	SCSI	Identify Target
12	SCSI	Identify Target
13	SCSI	Extended Copy
14	#	Write repeating data pattern
15	SCSI	LUN Write Same

Name **Value**

- Text Keys**
 - Initiator Name =@UP(0, A)
 - Session Type Normal
- Target**
 - Target Name =@UP(0, B)
- AuthMethod**
 - CHAP True
 - None False
- CHAP Authentication Credentials**
 - Username =@UP(0, CHPTRG_U)
 - Password =@UP(0, CHPTRG_P)
 - 2-WAY CHAP True
- Initiator CHAP Authentication Cr...**
 - Initiator Username =@UP(0, CHPINT_U)
 - Initiator Password =@UP(0, CHPINT_P)

iSCSI Sample Projects

Client iSCSI All Commands.client_scenario*

#	Protocol	Name
1	iSCSI	Open iSCSI Connection
2	iSCSI	Login Request
3	SCSI	Report LUNs
4	SCSI	Test Unit Ready
5	SCSI	Read Capacity(10)
6	SCSI	Read Capacity(16)
7	iSCSI	NOP-Out
8	SCSI	Inquiry
9	SCSI	Mode Sense(6)
10	SCSI	Mode Sense(6)
11	SCSI	Mode Sense(6)
12	SCSI	Mode Sense(6)
13	SCSI	Mode Sense(10)
14	SCSI	Mode Sense(10)
15	SCSI	Mode Sense(10)
16	SCSI	Mode Sense(10)
17	SCSI	Request Sense
18	SCSI	Mode Select(6)
19	SCSI	Mode Select(10)
20	SCSI	Reserve(6)
21	SCSI	Release(6)
22	SCSI	Start/Stop Unit
23	SCSI	Verify(10)
24	SCSI	LUN Read
25	SCSI	Read(6)
26	SCSI	Read(10)
27	SCSI	Read(16)
28	SCSI	LUN Write
29	SCSI	Write(6)
30	SCSI	Write(10)
31	SCSI	Write(16)
32	iSCSI	Logout Request
33	iSCSI	Close iSCSI Connection

Name **Value**

- Input**
 - Destination Address =@UP(0,A)
 - Destination Port 3260
 - Keep Alive False
- Reconnect**
 - Maximum Attempts 0
 - Reconnect Delay Interval (ms) 1,000
 - Reconnect To Redirect Destination False
- Reconnect Conditions**
 - Source TCP Port Change Mode 0xE880 (59520)
 - Reconnect Actions Change port on every reconnect
- Output**
 - Output Handle 1: SCSIConnectionHandle

OK Cancel

Client iSCSI LUN Read Threads in a Loop.client_scenario

#	Protocol	Name
1	iSCSI	Open iSCSI Connection
2	iSCSI	Login Request
3	SCSI	Report LUNs
4	SCSI	Read Capacity(10)
5	SCSI	Read Capacity(10)
6	SWT	Begin Loop
7	SWT	Begin Loop
8	SWT	Begin Thread
9	SCSI	LUN Read
10	SWT	End Thread
11	SWT	Begin Thread
12	SCSI	LUN Read
13	SWT	End Thread
14	SWT	End Loop
15	SWT	End Loop
16	SWT	Wait For All Threads
17	iSCSI	Logout Request
18	iSCSI	Close iSCSI Connection

Input

Connection Handle	Default
ISIDT	1
ISIDA	0
ISIDB	= @RANDOM(1,10000,1)
ISIDC	= @RANDOM(1,10000,1)
ISIDD	= @RANDOM(1,10000,1)
TSIH	New
Response Timeout	30,000

Text Keys

Initiator Name	= @STRING(iqn.2009.09.com.SWFTT:snc) + @RANDOM(0,999999,1)
Session Type	Normal

Target

Target Name	iqn.1986-03.com.sun:02:99d9ab85-c310-4fe5-8c6d-979447aae809
-------------	---

AuthMethod

CHAP	False
None	True

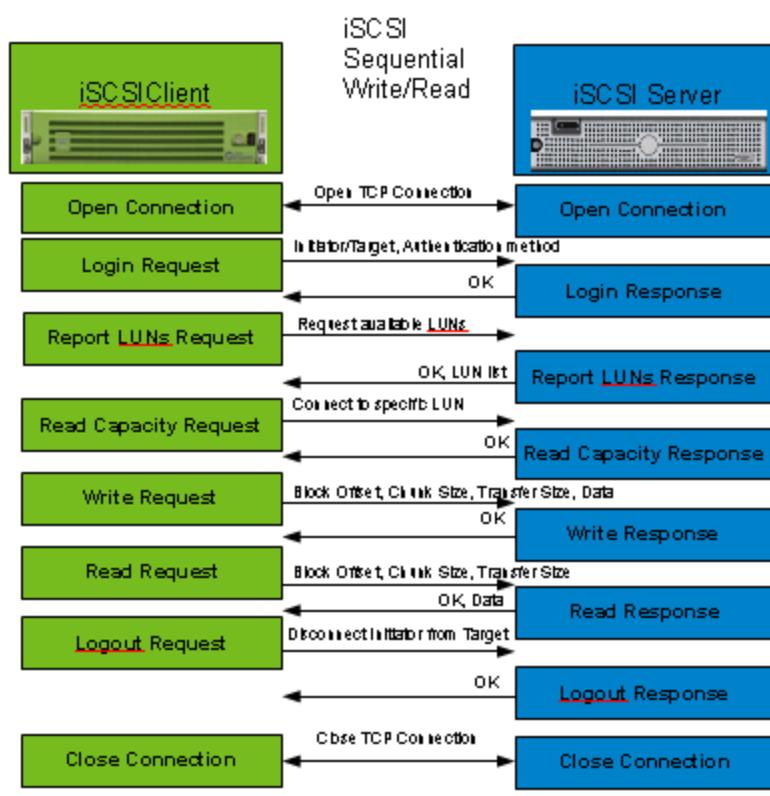
Enable Redirect

Completion Status	
Scenario Impact	

OK Cancel

iSCSI Project Flow

The following is the iSCSI client/server interaction flow for a simple sequential write/read Project.



iSCSI Statistics

See the [Advanced Concepts: Test Execution Rules](#) for a complete list of the statistics generated when iSCSI Actions are used in a Scenario.

SCSI Action Retries

See [Reference: FC/iSCSI/SCSI Commands and Behaviors](#) for details.

SCSI Per LUN Statistics

See [Reference: FC/iSCSI/SCSI Commands and Behaviors](#) for details.

iSCSI Caveats

- Multiple iSCSI Open TCP Connection Actions per Scenario are not supported.
- The internal SCSI Server (Start SCSI Server Action) does not save data written to it.
- To handle iSCSI authentication error responses when using 1-WAY and 2-WAY CHAP Authentication, use Completion Status = **Error_Other** for 2-WAY CHAP Authentication or Completion Status = **Authentication_Failure** for 1-WAY CHAP Authentication.

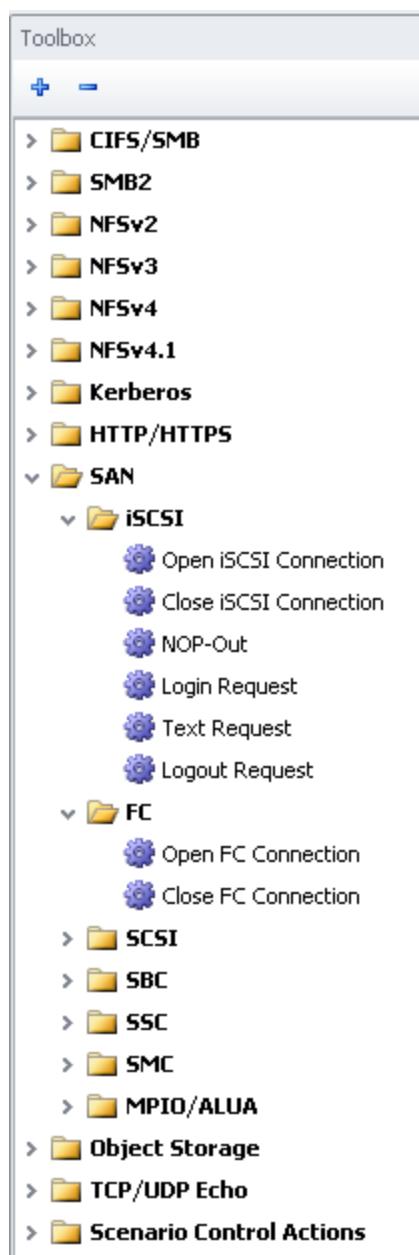
Reference: FC/iSCSI/SCSI Commands and Behaviors

Reference: Fibre Channel/iSCSI/SCSI Commands and Behaviors

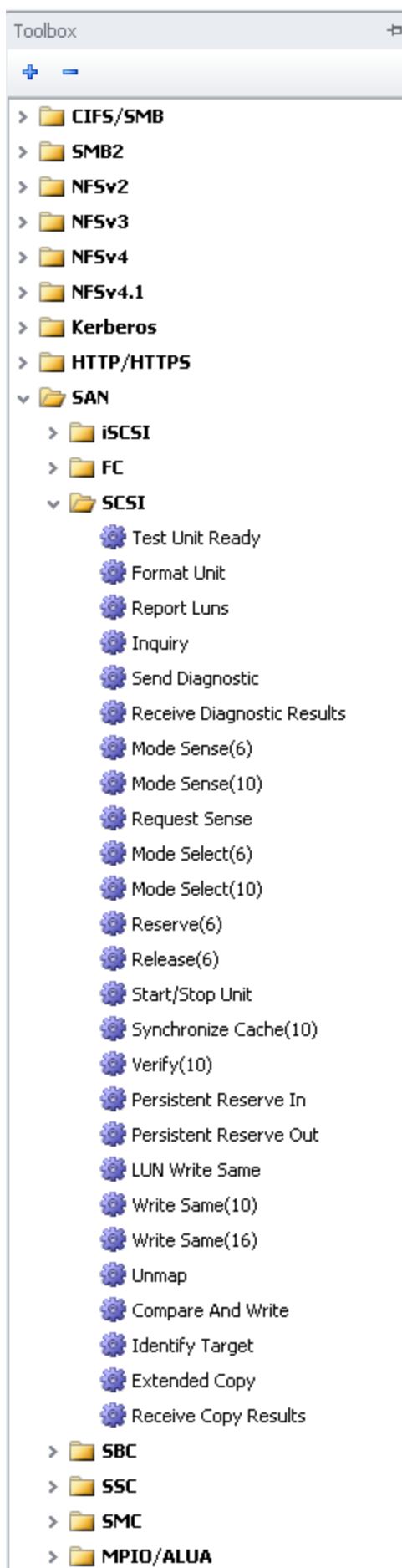
The Load DynamiX TDE and Automation support two Block Protocol transports, Fibre Channel (FC) and iSCSI (TCP/IP). These transmission technologies can be used to send and receive SCSI Block Protocol commands. The Toolbox entries (Actions) for these two transmission technologies are similar with Open and Close Connection Actions to open and close the connection to a Fibre Channel, Fibre Channel over Ethernet or iSCSI capable device. iSCSI in addition provides four additional Actions: **NOP Out, Login Request, Logout Request and Text Request**.

The SCSI protocol Toolbox contains the SCSI Actions for read and write operations as well as device status and device management commands.

The SAN (FC and iSCSI folders) toolbox contains the Client (Initiator) Actions for establishing iSCSI and Fibre Channel connections:

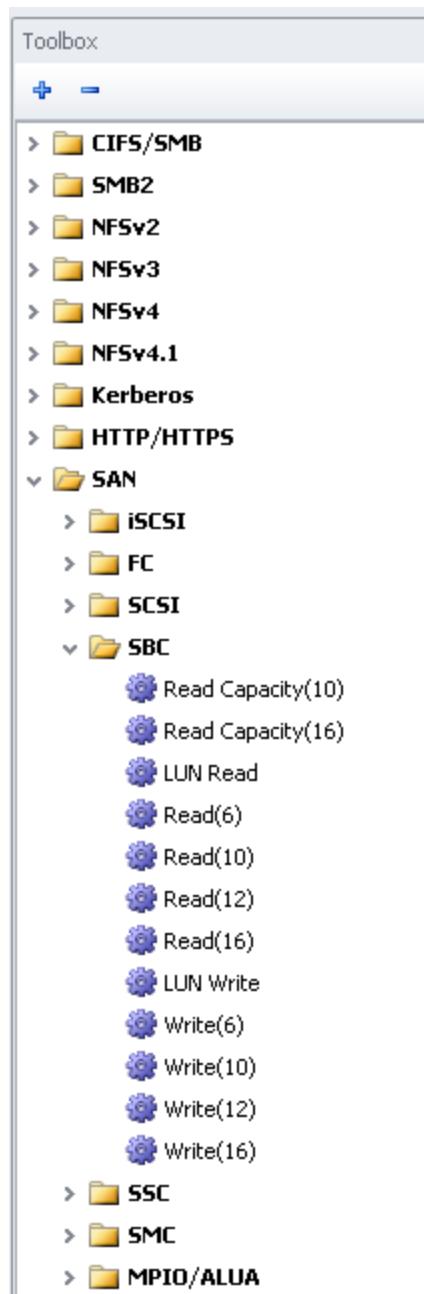


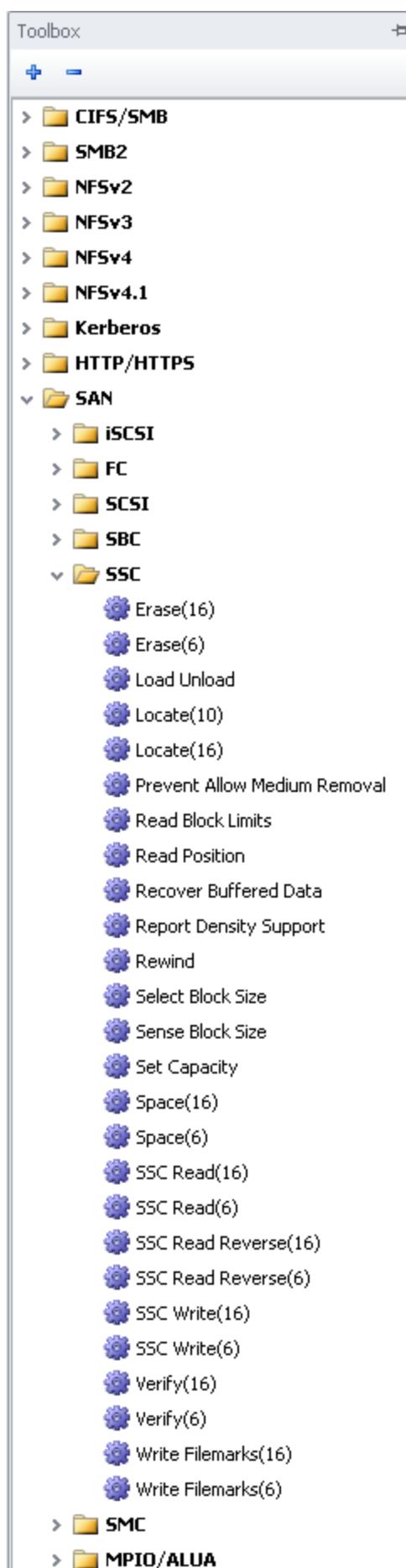
and a broad collection of SCSI-protocol-specific commands (SCSI folder):

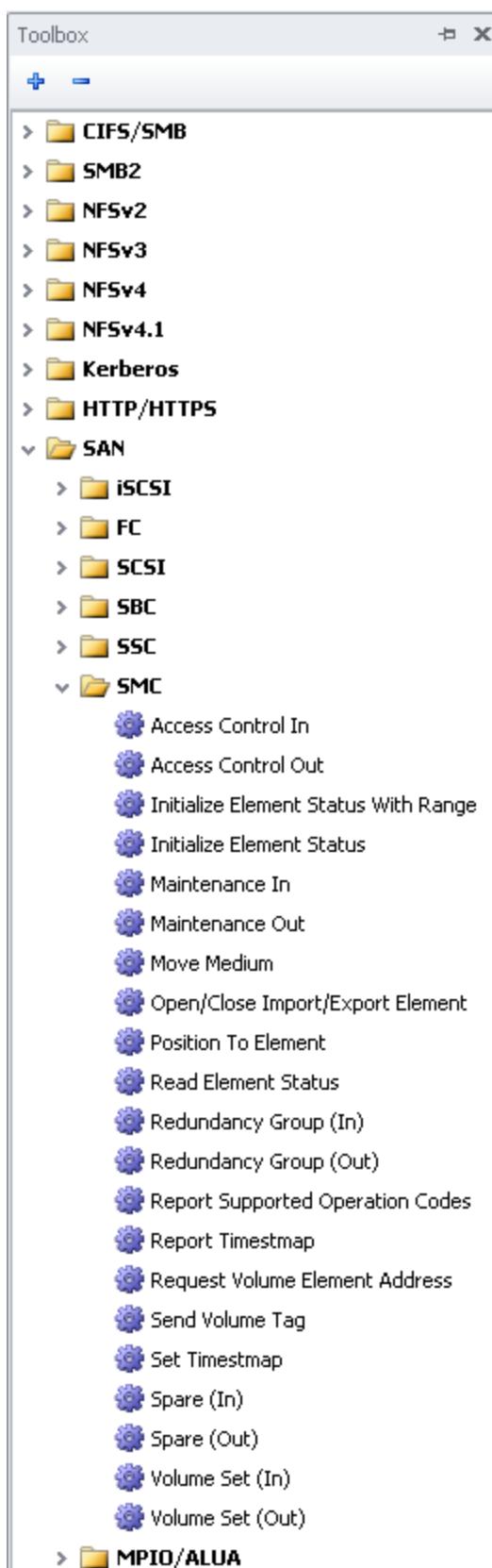


There are four other folders of SCSI commands that contain device-specific SCSI Actions, one for

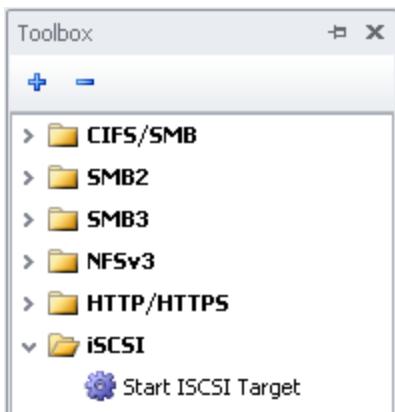
Direct Access (SBC) devices, one for Streaming (SSC) devices, one for SCSI Medium Changer (SMC) commands and a folder containing the MPIO/ALUA related Actions (see below for a MPIO/ALUA discussion)







In addition, there is a **Start iSCSI Target** (Server) Action. The **Start iSCSI Target** Action (shown in the Server Toolbox below) supports only the basic Direct Access Device (SBC) SCSI Actions and the iSCSI Actions. It does not support SSC Actions or VAAI Actions. The iSCSI Target is designed to support testing of IP infrastructures, such as WAN Optimization devices, routers and switches, using the Load DynamiX iSCSI Initiators. It is not designed to support testing of 3rd party iSCSI Initiators.



SCSI SSC (VTL) commands

The SCSI SSC Toolbox folder contains the SCSI commands that are to be used on SCSI Streaming devices such as Tape Drives. These Streaming Devices are also known as Sequential Access Devices and differ from the SBC devices (Direct Access Devices) in that they are intended to be accessed sequentially not randomly. This does not mean these devices cannot be accessed randomly but the Action sequence to do that is much more complicated than that of a Direct Access Device. Sequential Access Device concepts:

Partition: A detectable region of the Sequential Access Device.

Current Position : the position on the Sequential Access Device of the read/write head. It is at this Current Position that the next read or write command, or positioning operations like Space, will take place.

Current Partition: The Partition in which the Current Position is in.

Logical Object: a set of data written to the Sequential Device that has a detectable beginning and end. It could a file, a directory or an entire disc image.

Most of the SCSI Actions in the SSC Toolbox are designed to manage the Sequential Access Device rather than read or write to it:

Read Block Limits: The equivalent of Read Capacity for Direct Access Devices. Establishes the LUN of interest on the Sequential Access Device and provides a Handle (TapeHandle) for that LUN.

Sense Block Size: Used to read the Block Size of a SCSI Sequential Device.

Select Block Size: Use to set the Block Size of a SCSI Sequential Device by setting a positive integer value in the Block Size field.

All other SSC commands require a TapeHandle and operate on the LUN that is specified in the Read Block Limits command. All Actions below apply to the LUN of the TapeHandle input they require.

- **Erase(6):** Erase the Sequential Access Device starting at the specified Logical Object of the specified Partition.
- **Erase(16):** Erase the Sequential Access Device starting at the specified Logical Object of the specified Partition.
- **Locate(10):** Locate the specified Logical Object on the specified Partition of a Sequential Access Device.
- **Locate (16):** Locate the specified Logical Object on the specified Partition of a Sequential Access Device.
- **Rewind:** Move the Current Position to the beginning of the Current Partition
- **Space(6):** Move the Current Position to a new Current Position either a Count of Logical Blocks or a Count of Filemarks. (See the Operation of the SCSI Space command section

below)

- **Space(16)**: Move the Current Position to a new Current Position either a Count of Logical Blocks or a Count of Filemarks. (See the Operation of the SCSI Space command section below)

Note: In general, Space (-1 filemark) places the Current Position at the end of the previous file (or Logical Object).

- **Load Unload**: Used to Load or Unload ("mount" or "unmount") a LUN of a Sequential Access Device. This command can make targets unavailable, it should be used judiciously.
- **Read Position**: Read the Current Position of a LUN of a Sequential Access Device.
- **Report Density Support**: Returns various kinds of information regarding a LUN of a Sequential Access Device (media width, tracks, capacity, etc).
- **Set Capacity**: Sets the Capacity of the LUN of a Sequential Access Device to the specified size.
- **Prevent Allow Media Removal**: Enable or Disable the removal of Media
- **Verify(6)**: Verify one or more blocks starting at the current location.
- **Verify(16)**: Verify one or more blocks starting at the current location.
- **Recover Buffered Data**: Recover data written to a Logical Unit's object buffer but not successfully written to the medium.

There are only a few commands used to read or write a Sequential Access Device:

- **SSC Read Reverse(6)**: Read Total Transfer Size bytes from the current position of the Sequential Access Device, Chunk Size bytes at a time in Reverse direction on the medium.
- **SSC Read Reverse(16)**: Read Total Transfer Size bytes from the current position of the Sequential Access Device, Chunk Size bytes at a time in Reverse direction on the medium.
- **SSC Read(6)**: Read Total Transfer Size bytes from the current position of the Sequential Access Device, Chunk Size bytes at a time.
- **SSC Read(16)**: Read Total Transfer Size bytes from the current position of the Sequential Access Device, Chunk Size bytes at a time.
- **SSC Write(6)**: Write Total Transfer Size bytes starting at the current position of the Sequential Access Device, Chunk Size bytes at a time.
- **SSC Write(16)**: Write Total Transfer Size bytes starting at the current position of the Sequential Access Device, Chunk Size bytes at a time.

Note: **SSC Read (6/16)** and **SSC Write (6/16)** Action Chunk Size limits vary depending on the target SCSI device. Observed maximum is 128MB. The response to the Read Block Limits command contains the maximum value for SSC Command Chunk Size. This value can be seen in the **Read Block Limits** Action response in a PCAP file. Using a Chunk Size greater than what Read Block Limits specifies will result in a "Transmission Failed" error.

SSC Read (6/16) and **SSC Write (6/16)** Actions can be variable sized (FIXED input field == 0) or they can be fixed size (FIXED input field == 1). When FIXED == 1, Chunk Size is specified in terms of the number of Blocks that are specified in either what was read from the SCSI Sequential Device by the Sense Block Size Action or what is written to the SCSI Sequential Device by the Select Block Size Action.

Write Filemarks(6): Write a special indicator to the Sequential Access Device at the Current Position.

Write Filemarks(16): Write a special indicator to the Sequential Access Device at the Current Position. It is recommended that any Project writing to a Sequential Access Device using the SSC SCSI commands, write filemarks at the beginning of the Project.

SCSI Medium Changer commands

The SCSI Medium Changer (SMC) commands are used to control SCSI devices (typically Streaming devices) that support removable media (ex: tape jukeboxes).

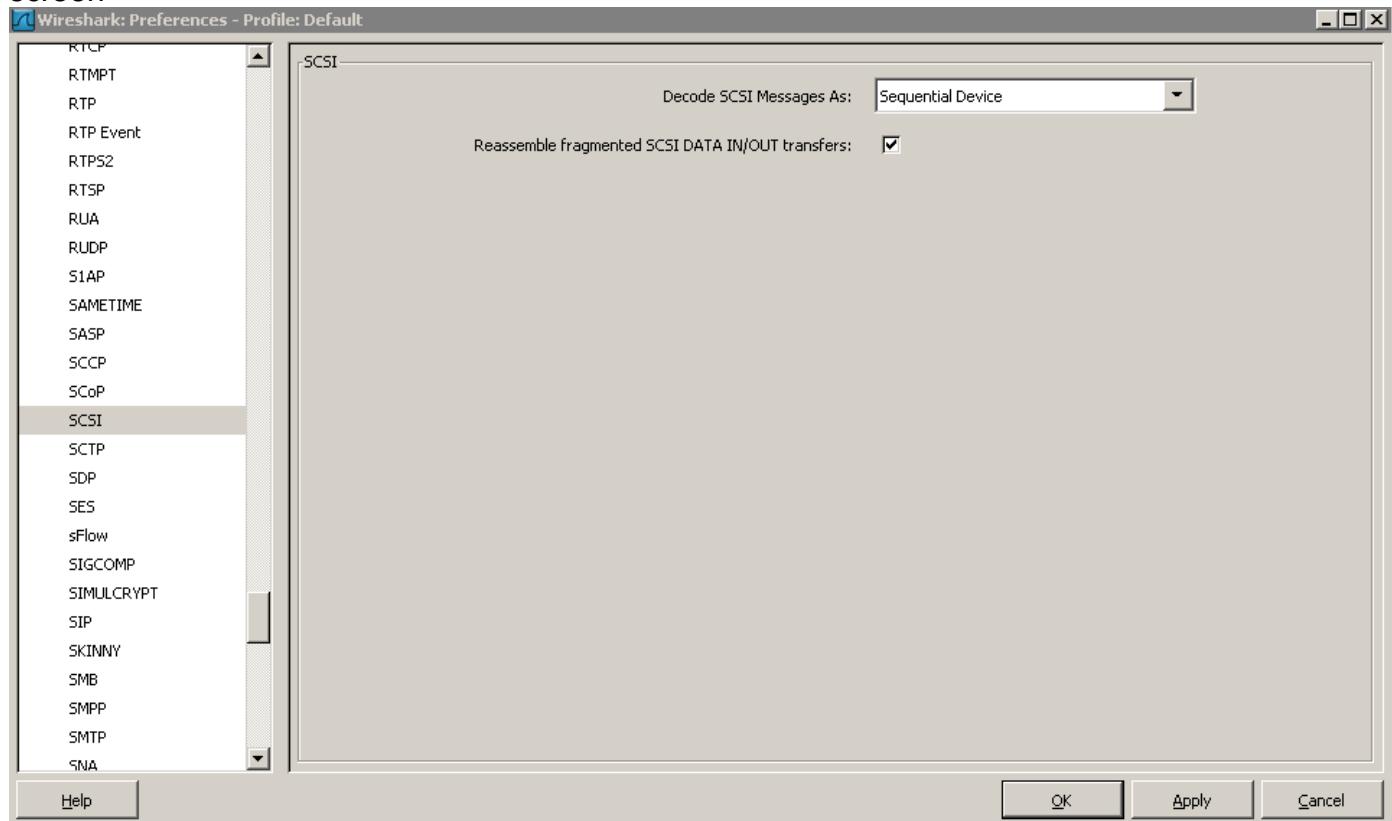
- **Access Control In:** Retrieve device Access Control setting information.
- **Access Control Out:** Change device Access Control settings.
- **Initialize Element Status With Range:** This command shall cause the media changer to check the specified elements for volume status and any other relevant status. This command enables the initiator to get a quick response from a Read Element Status command that may follow, and is useful after a power failure, if tape medium has been changed by an operator, or if subsystem configuration has changed.
- **Initialize Element Status:** This command directs the medium changer to check all existing elements for tape cartridges and any status relevant to that element. This command enables the initiator to get a quick response from a Read Element Status command that may follow, and is useful after a power failure, if tape medium has been changed by an operator, or if subsystem configuration has changed.
- **Maintenance In:** Retrieve device management setting information.
- **Maintenance Out:** Change device management settings.
- **Move Medium:** This command is used to move tape cartridges from one element address to another specific element address.
- **Open/Close Import/Export Element:** This command allows an application client to open the Import/Export element, also referred to as I/O Station. When the action code is set to Open Import/Export Element, the library will open the import/export element. The library will not return a Check Condition status when the import/export element was already open.
- **Position To Element:** This command allows the initiator to position the Medium Transport Element to a specific element address position. This destination address can be a Storage Element, Import/Export Element or a Data Transfer Element address.
- **Read Element Status:** This command is sent to a target from the initiator requesting that the target report the status of its internal elements
- **Redundancy Group (In):** Retrieve device Redundancy Group setting information.
- **Redundancy Group (Out):** Change device Redundancy Group settings.
- **Report Supported Operation Codes:** This command requests information on commands the addressed logical unit supports. An application client may request a list of all operation codes and service actions supported by the logical unit or the command support data for a specific command.
- **Report Timestamp:** This command requests the value of the logical unit's timestamp.
- **Request Volume Element Address:** This command is used to transfer the results of the SEND VOLUME TAG command. Multiple REQUEST VOLUME ELEMENT ADDRESS commands may be used to retrieve the results of a single SEND VOLUME TAG command with the translate option.
- **Send Volume Tag:** This command transfers a volume tag template to be used for a search of existing volume tag information or new volume tag information for one media changer element address.
- **Set Timestamp:** This command sets the value of the logical unit's timestamp.
- **Spare (In):** Retrieve device Spares setting information.
- **Spare (Out):** Change device Spares settings.
- **Volume Set (In):** Retrieve device Volume Set setting information.
- **Volume Set (Out):** Change device Volume Set settings.

SCSI Streaming Command Tracing (PCAP)

When tracing SCSI Projects accessing Sequential Devices and using the SCSI Streaming Command set, Wireshark should be configured to show Sequential Device commands and not

Block Device commands. NetMon will **NOT** display the PCAP files produced by the Tracing Resource on a Fibre Channel Project.

Wireshark is configured to show Sequential Device commands via the Edit > Preferences <Protocol screen



A SCSI Streaming Command Project PCAP file displayed by Wireshark would look like

The screenshot shows the Wireshark interface with a captured file named 'Client Port 0(172.17.1.161 port 1).pcap'. The packet list pane shows 14 FCP frames. The first few frames are highlighted with a red box, showing details for frame 1:

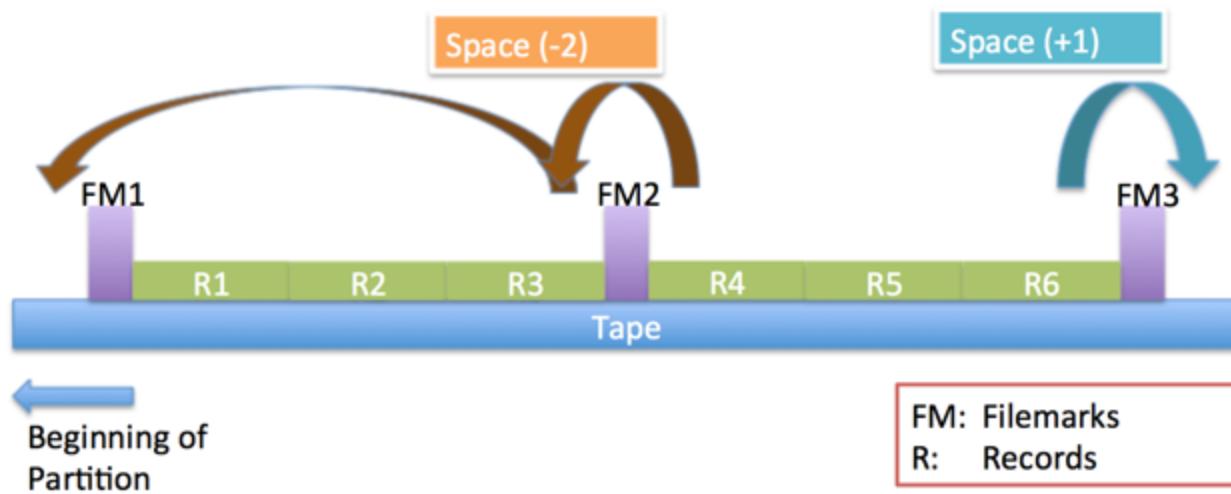
- Frame 1: 56 bytes on wire (448 bits), 56 bytes captured (448 bits)
- Fibre Channel
- FCP: FCP_CMDN
- SCSI CDB Test Unit Ready
- [LUN: 0x000b]
- [Command set: Sequential Access Device (0x01) (Using default commandset)]
- Response in: 129551
- opcode: Test Unit Ready (0x00)
- Control: 0x00

The commands are noted to be from a Sequential Access Device instead of a Block Device.

Operation of the SCSI Space command

The Space command allows the Current Position to move farther away from the beginning of the partition by specifying a positive integer, move closer to the beginning of the partition by specifying a negative integer, or remain stationary by specifying zero (0). In some cases, it may be necessary to use two Space commands to properly position the read/write head. For example, in the diagram

below, if after filemark FM2 is written, the user would like to go back to read the records R4, R5 and R6, it is necessary to move the position of the read/write head to the end of filemark FM1. To do so, according to the SSC-3 standard, a Space -2 followed by a Space +1 is required, as illustrated.



The SCSI Space command allows the user to move forward or backward in Bytes, Filemarks or Sequential Filemarks as well as to special locations like End of Data.

Concurrent Streams

A common use case is to utilize multiple concurrent streams to increase overall backup and recovery performance. To test multiple concurrent streams, use Concurrent Scenarios in the Load Profile function. See the Load Profile section on the usage and behavior of Concurrent Scenarios.

SCSI VAAI commands

There are a number of commands in the SCSI command set that are used to help improve the efficiency of virtual image creation and cloning operations. See the discussion of VAAI in [Reference: ISCSI Commands and Behaviors](#).

SCSI Mode Sense and Mode Select Actions

The Load DynamiX SCSI Mode Sense and Mode Select Actions allow the Tester to Read/Verify and Write SCSI device meta (control) data for specific LUNs on a SCSI device. There are four Actions:

- **Mode Sense (6)** - Read and optionally verify control data for a LUN for specific control information pages from a SCSI device
- **Mode Sense (10)** - Read and optionally verify control data for a LUN for specific control information pages from a SCSI device
- **Mode Select (6)** - Write control data for a LUN for specific control information pages to a SCSI device
- **Mode Select (10)** - Read and optionally verify control data for a LUN for specific control information pages to a SCSI device

Mode Sense

Specify the source LUN and control page and identify which of the control page parameters to validate and against what data. The output of the validate process are:

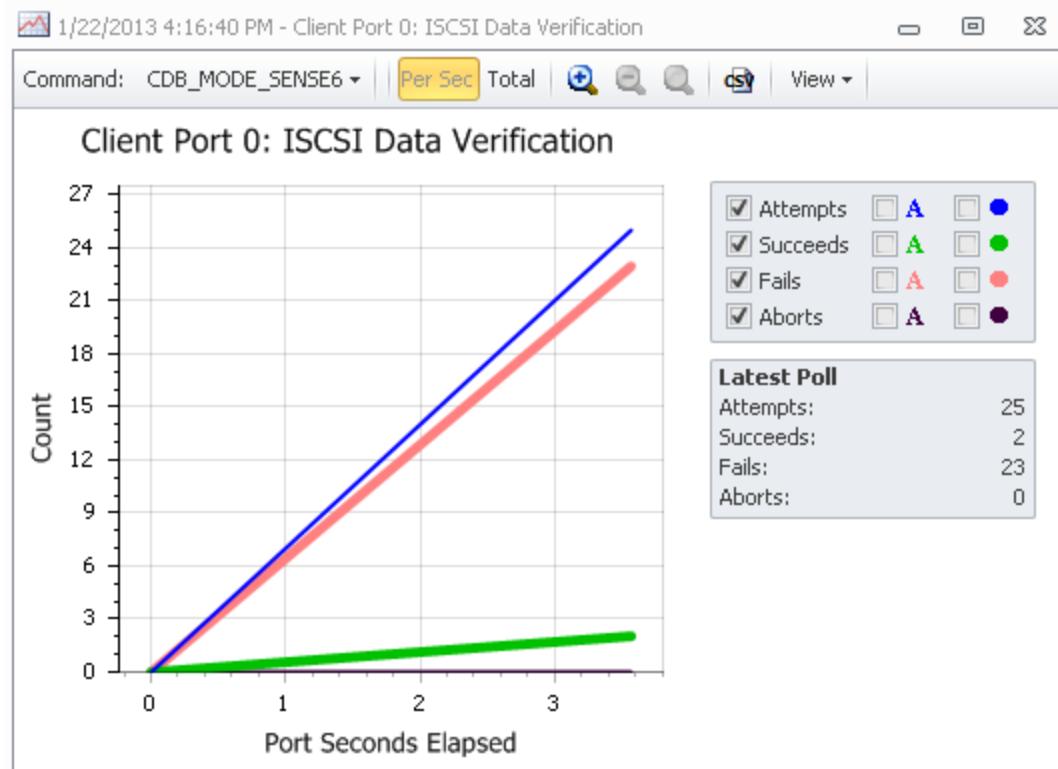
A SCSI Data Verify graph and a SCSI Modepage Data Verification Failures csv file if there are mismatches between the control page parameters and the user specified values.

Mode Sense Action

Client FC All Commands.client_scenario*

#	Protocol	Name		Name	Value
1	FC	Open FC Connection		Connection Handle	Default
2	SCSI	Test Unit Ready		LUN	=@UP(0, C)
3	SCSI	Report LUNs		PC	00b Current Values
4	SCSI	Read Capacity(10)		PageCode	03h Format Parameters Page
5	SCSI	Read Capacity(16)			
6	SCSI	Inquiry			
7	SCSI	Mode Sense(6)			
8	SCSI	Mode Sense(6)		PS	Yes
9	SCSI	Mode Sense(6)		Value	1
10	SCSI	Mode Sense(6)			
11	SCSI	Mode Sense(10)		PAGE CODE	Yes
12	SCSI	Mode Sense(10)		Validate	1
13	SCSI	Mode Sense(10)		Value	
14	SCSI	Mode Sense(10)			
15	SCSI	Mode Select(6)		PAGE LENGTH	Yes
16	SCSI	Mode Select(10)		Validate	1
17	SCSI	Start/Stop Unit		Value	
18	SCSI	Synchronize Cache(10)			
19	SCSI	Verify(10)		TRACKS PER ZONE	Yes
20	SCSI	LUN Read		Validate	1
21	SCSI	Read(6)		Value	
22	SCSI	Read(10)			
23	SCSI	Read(12)		ALTERNATE SECTORS PER ZONE	Yes
24	SCSI	Read(16)		Validate	1
				Value	

SCSI Data Verification Graph (user can select which Action to view results for)



SCSI Modepage Data Verification Failures csv

Client Port 0(172.17.1.49 port 0) SCSI-Modepage Data Verification Failures.csv

Row Index	Time(micros)	LUN	Page Code	Param Name	Received V...	Expected V...	
1	3003393	0	0x03	PS	0x0	0x1	
2	3003399	0	0x03	PAGE LENGTH	0x16	0xe3	
3	3003402	0	0x03	TRACKS PE...	0x0	0x1	
4	3003405	0	0x03	ALTERNATE...	0x0	0x1	
5	3003408	0	0x03	ALTERNATE...	0x0	0x1	
6	3003411	0	0x03	ALTERNATE...	0x0	0x1	
7	3003414	0	0x03	SECTORS P...	0xc80	0x1	
8	3003417	0	0x03	DATA BYTE...	0x200	0x1	
9	3003420	0	0x03	INTERLEAVE	0x0	0x1	
10	3003423	0	0x03	TRACK SKE...	0x0	0x1	
11	3003426	0	0x03	CYLINDER ...	0x0	0x1	
12	3003428	0	0x03	SSEC	0x1	0x0	
13	3003431	0	0x03	HSEC	0x1	0x0	
14	3003434	0	0x03	RMB	0x0	0x1	
15	3003437	0	0x03	SURF	0x0	0x1	
16	3003440	0	0x03	DRIVE TYPE	0x0	0x1	
17	3003578	0	0x0a	PS	0x0	0x1	
18	3003581	0	0x0a	SPF	0x0	0x1	
19	3003584	0	0x0a	PAGE LENGTH	0xa	0xb	
20	3003587	0	0x0a	TST	0x0	0x1	
21	3003590	0	0x0a	TMSC ONLY	0x0	0x1	

Mode Select

The Tester may use the Mode Select Action to write specific control page parameters to a SCSI device.

Client FC All Commands.client_scenario*

#	Protocol	Name
1	FC	Open FC Connection
2	SCSI	Test Unit Ready
3	SCSI	Report LUNs
4	SCSI	Read Capacity(10)
5	SCSI	Read Capacity(16)
6	SCSI	Inquiry
7	SCSI	Mode Sense(6)
8	SCSI	Mode Sense(6)
9	SCSI	Mode Sense(6)
10	SCSI	Mode Sense(6)
11	SCSI	Mode Sense(10)
12	SCSI	Mode Sense(10)
13	SCSI	Mode Sense(10)
14	SCSI	Mode Sense(10)
15	SCSI	Mode Select(6)
16	SCSI	Mode Select(10)
17	SCSI	Start/Stop Unit
18	SCSI	Synchronize Cache(10)
19	SCSI	Verify(10)
20	SCSI	LUN Read
21	SCSI	Read(6)
22	SCSI	Read(10)

Name

Value

Input

- Connection Handle: Default
- SCSIModePageHandle: 7: SCSIModePageHandle
- LUN: =@UP(0, C)
- SP: 0
- PageCode: 03h Format Parameters Page

Format Parameters Page

- PS
 - Change: Yes
 - Value: 1
- PAGE CODE
 - Change: Yes
 - Value: 3
- PAGE LENGTH
 - Change: Yes
 - Value: 22
- TRACKS PER ZONE
 - Change: Yes
 - Value: 1
- ALTERNATE SECTORS PER ZONE
 - Change: Yes
 - Value: 1

SCSI Read Capacity (10) LUN Acceptable Value Range

By default the acceptable value range for the LUN input field is 0..65535.

#	Protocol	Name
1	FC	Open FC Connection
2	SCSI	Test Unit Ready
3	SCSI	Report LUNs
4	SCSI	Read Capacity(10)
5	SCSI	Read Capacity(16)
6	SCSI	Inquiry
7	SCSI	Mode Sense(6)
8	SCSI	Mode Sense(6)
9	SCSI	Mode Sense(6)
10	SCSI	Mode Sense(6)
11	SCSI	Mode Sense(10)
12	SCSI	Mode Sense(10)
13	SCSI	Mode Sense(10)
14	SCSI	Mode Sense(10)
15	SCSI	Mode Select(6)

SCSI Read/Write using IO Manager

All iSCSI or FC Reads and Writes are SCSI Actions. SCSI Reads and Writes (**LUN Read/Write**, **Read/Write(6/10/12/16)**) all support an input field named **Use IO Manager** which is a **True/False** setting. If **Use IO Manager** input is set to **True** then the IO Manager Action that is defined in the IO Manager input field controls the behavior of this Read or Write. The IO Manager Action itself defines four elements that are input to the Read or Write operation:

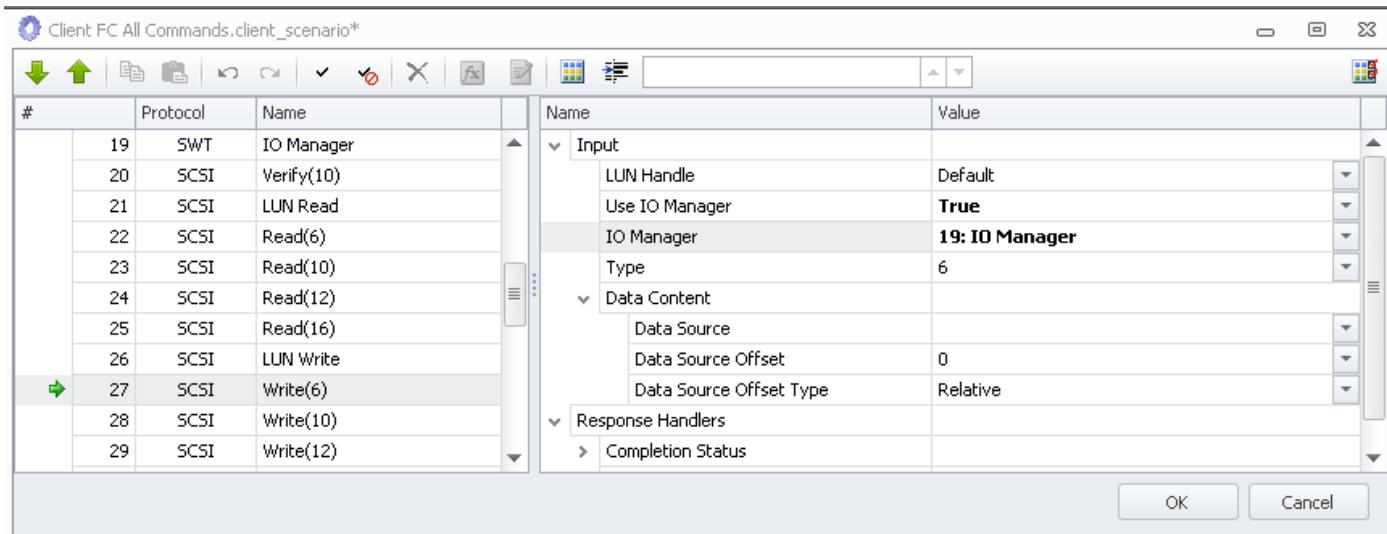
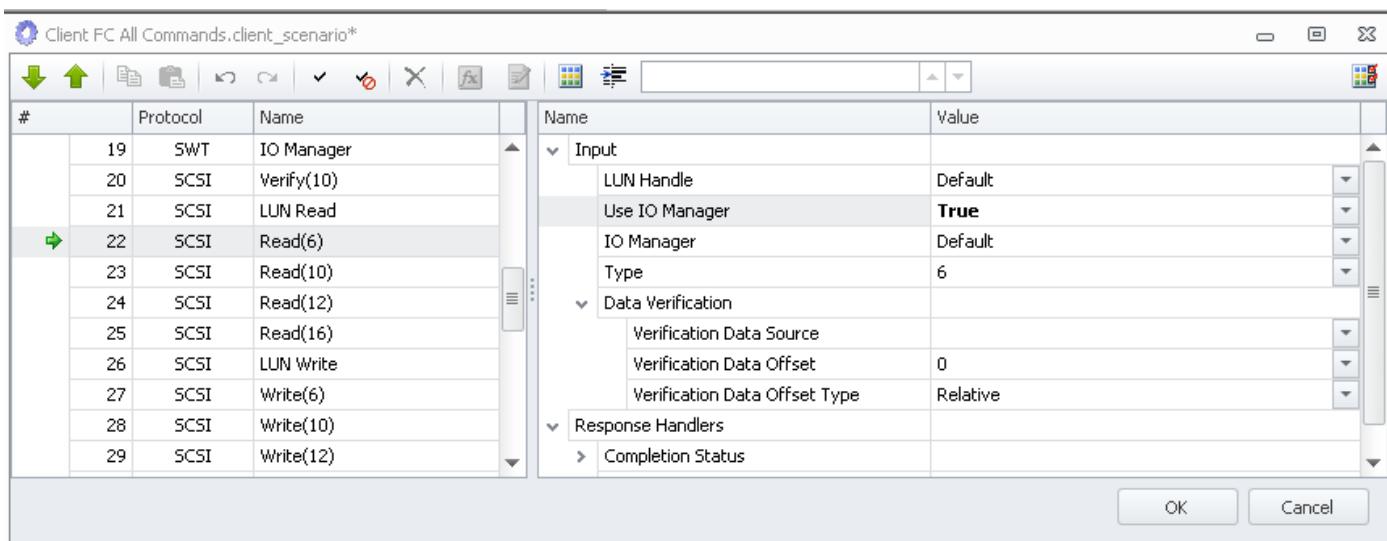
Order: Random or Sequential, the order in which Blocks are Read or Written

Start Offset: The lower bound of the Read or Write Actions

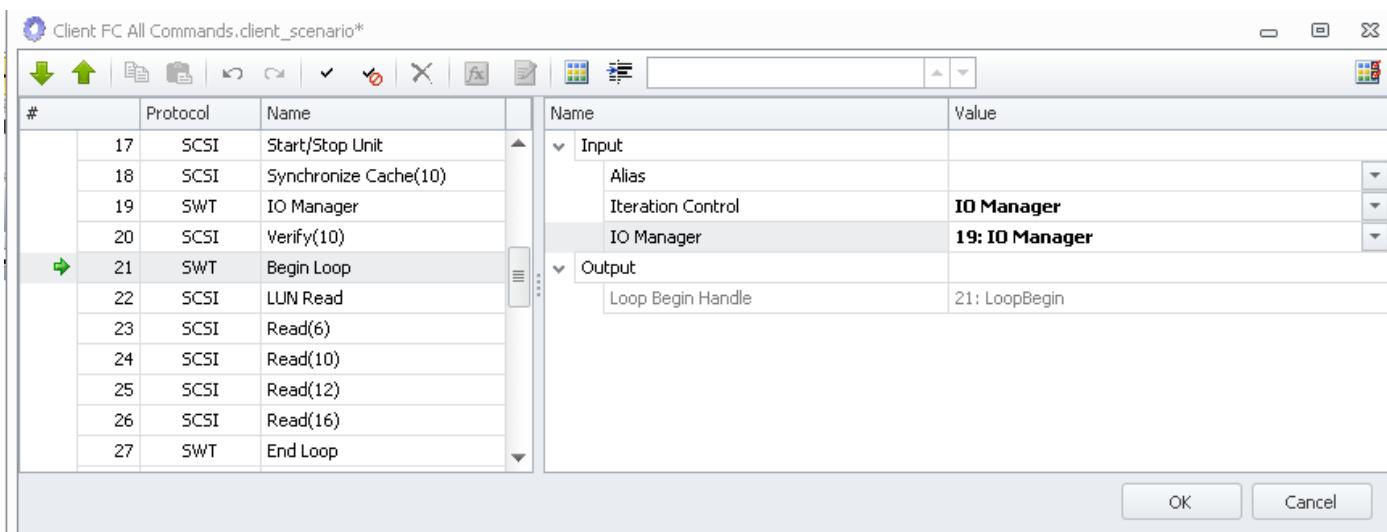
Request Size: The Chunk Size of the Read or Write Actions

Total Size: The total number of bytes to be Read or Written

When **Use IO Manager** is set to **True**, the Read or Write Action's Offset, Chunk Size, Total Transfer Size, Block Sequence and Number of Outstanding Requests input fields are hidden since those values will be provided by the IO Manager Action. When a SCSI Read or Write is controlled by an IO Manager, each successive Read or Write's inputs (offset, chunk size and total size) are provided by the IO Manager instance that it references. A Scenario can have multiple instances of IO Manager Actions that control different Read and/or Write operations. A single instance of Read or Write Action (not in a Begin-End loop) controlled by IO Manager will be executed a single time with the parameters input to the IO Manager action.



The IO Manager can also be used to control the behavior of a Begin-End loop that contains Reads and/or Writes. In the Scenario above the Begin Action's Iteration Control input is set to IO Manager which will force the Begin-End loop to continue executing as long as the IO Manager has data to Read or Write. To iterate through a region of storage on a target device using IO Manager, a Begin-End loop must be used.



The IO Manager Action is currently has an experimental status and is not recommended for broader use. It is of most value in a Scenario that requires Random reads and/or writes and a mix of ::DataContent types.

SCSI Commands Number of Outstanding Requests

All iSCSI or FC reads and writes and device status/control Actions are SCSI Actions. SCSI Reads and Writes (**LUN Read/Write**, **Read/Write(6)**, **Read/Write(10)**, **Read/Write(12)**, **Read/Write(16)**) all support an input field named Number of Outstanding Requests. This field with a value > 1 indicates to the Scenario that the read or write request is to be broken into up to Number of Outstanding Requests requests and sent to the target in parallel.

The default value of Number of Outstanding Requests is 1 which results in a single request no parallelism in requests that are sent to the target device. The generic behavior of a Read or Write with Number of Outstanding Requests > 1 is that the operation (Read or Write) is broken into up to N independent operations of size Bytes per Block where

$N = \min(\text{Total Bytes}/\text{Bytes per Block}, \text{Number of Outstanding Requests})$

whichever is smaller and these N operations are all issued at the same time. As these requests complete, new operations are issued until the total number represented by (Total Bytes/Bytes per Block) has completed. Any errors encountered during the processing of these requests will cause the entire Action to fail.

See the discussion of Number of Outstanding Requests in [Reference: iSCSI Commands and Behaviors](#) for more details.

SCSI Read and Write Data Source Absolute and Relative

All iSCSI or FC Read and Write Actions provide Relative and Absolute references to ::DataContent files to control the offset within the ::DataContent file that is used as the Write data source or the Read data comparison with what is read. Basically, Absolute references are absolute based on the Data Source Offset or the Verification Data Offset. Relative references are relative to the last reference to the ::DataContent file.

Relative Data References (default behavior)

In the following iSCSI example, the **LUN Read** and **LUN Write** Actions all use **Relative** references and read/write 512 bytes of data from Data Source Offset 0 to iSCSI disk offsets 0 and 512. The Action shown first is the second **LUN Write** Action that specifies the "Relative" offset of 0 into a ::Sequential ::DataContent file as the source of the data for the 512 byte write to Initial Offset 512. The **Relative** reference type tells the Load DynamiX Appliance Firmware to use the current location in the ::Sequential file (the location where the first **LUN Write** left off) as the starting location for the data of this write plus of an offset of 0.

The screenshot shows a software interface for defining a test scenario. On the left, a table lists a sequence of actions:

#	Protocol	Name
1	iSCSI	Open iSCSI Connection
2	iSCSI	Login Request
3	SCSI	Report LUNs
4	SCSI	Read Capacity(10)
5	SCSI	Read Capacity(10)
6	SWT	Begin Loop
7	SCSI	LUN Write
8	SWT	End Loop
9	iSCSI	Logout Request

On the right, detailed parameters for the selected action (LUN Write, row 7) are shown in a table:

Name	Value
Input	
LUN Handle	5: LUNHandle
Use IO Manager	False
Automatic Offset	False
Initial Offset	512
Chunk Size	=@UP(0, G)
Total Transfer Size	=@UP(0, H)
Block Sequence	Forward
Number of Outstanding Requests	2
Type	10
Data Content	
Data Source	::Sequential()
Data Source Offset	0
Data Source Offset Type	Relative

The corresponding **LUN Read** Action would tell the Appliance Firmware to compare the contents of the data read in this operation with the data in the ::Sequential file where the first **LUN Read** Action left off plus an offset of 0.

The screenshot shows a software interface for defining a test scenario. On the left, a table lists a sequence of actions:

#	Protocol	Name
1	iSCSI	Open iSCSI Connection
2	iSCSI	Login Request
3	SCSI	Report LUNs
4	SCSI	Read Capacity(10)
5	SCSI	Read Capacity(10)
6	SWT	Begin Loop
7	SCSI	LUN Read
8	SWT	End Loop
9	iSCSI	Logout Request

On the right, detailed parameters for the selected action (LUN Read, row 7) are shown in a table:

Name	Value
Input	
LUN Handle	Default
Use IO Manager	False
Automatic Offset	False
Initial Offset	512
Chunk Size	=@UP(0, G)
Total Transfer Size	=@UP(0, H)
Block Sequence	Forward
Number of Outstanding Requests	4
Type	10
Data Verification	
Verification Data Source	::Sequential()
Verification Data Offset	0
Verification Data Offset Type	Relative

Absolute Data References

In the following iSCSI example, the **LUN Read** and **LUN Write** Actions all use **Absolute** references and write 512 bytes of data from Data Source Offset 0 and 512. The Action shown first is the second **LUN Write** Action that specifies the "Absolute" offset of 512 into a ::Sequential file as the source of the data for the 512 byte write to Initial Offset 512. The **Absolute** type tells the Appliance Firmware to write data from the file starting at offset 512.

The screenshot shows the Test Project Explorer interface with a sequence of SCSI commands and their configuration details. The sequence includes:

- 1 iSCSI Open iSCSI Connection
- 2 iSCSI Login Request
- 3 SCSI Report LUNs
- 4 SCSI Read Capacity(10)
- 5 SCSI Read Capacity(10)
- 6 SWT Begin Loop
- 7 SCSI LUN Write** (highlighted in green)
- 8 SWT End Loop
- 9 iSCSI Logout Request

Configuration details for the LUN Write action (Step 7) are shown in the right panel:

Name	Value
Input	
LUN Handle	5: LUNHandle
Use IO Manager	False
Automatic Offset	False
Initial Offset	512
Chunk Size	=@UP(0, G)
Total Transfer Size	=@UP(0, H)
Block Sequence	Forward
Number of Outstanding Requests	2
Type	10
Data Content	
Data Source	::Sequential()
Data Source Offset	512
Data Source Offset Type	Absolute

The corresponding **LUN Read** Action is shown below. This Action specifies that data read from location 512 (Initial Offset input) is to be compared with the data at offset 512 of the ::Sequential file. This comparison would succeed because the data that was read back will be compared with same data as was written.

The screenshot shows the Test Project Explorer interface with a sequence of SCSI commands and their configuration details. The sequence includes:

- 1 iSCSI Open iSCSI Connection
- 2 iSCSI Login Request
- 3 SCSI Report LUNs
- 4 SCSI Read Capacity(10)
- 5 SCSI Read Capacity(10)
- 6 SWT Begin Loop
- 7 SCSI LUN Read** (highlighted in green)
- 8 SWT End Loop
- 9 iSCSI Logout Request

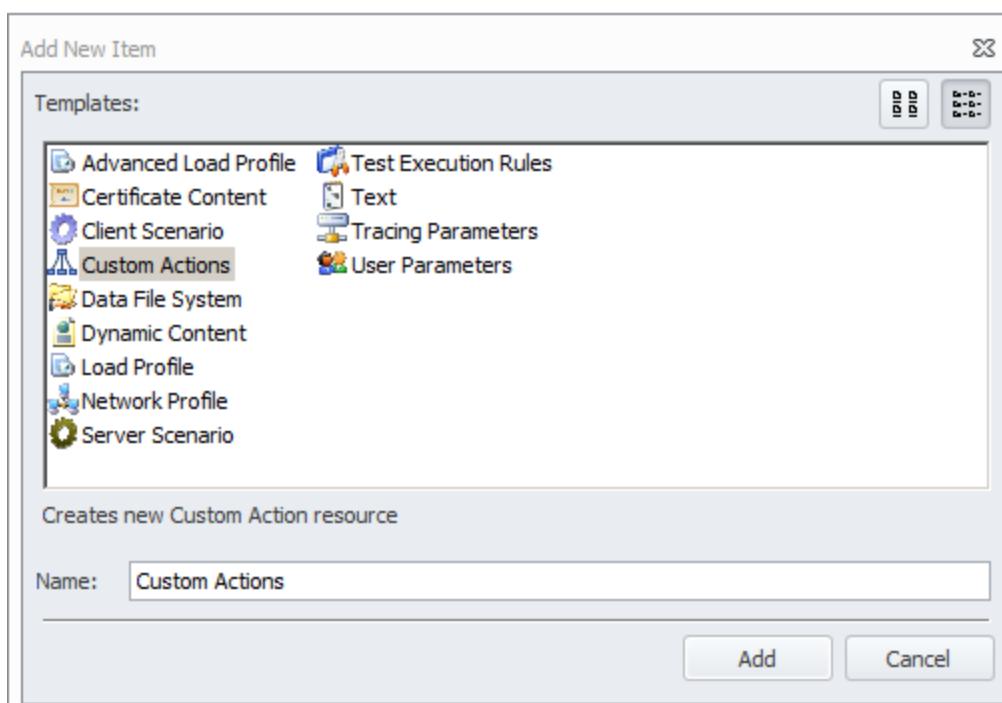
Configuration details for the LUN Read action (Step 7) are shown in the right panel:

Name	Value
Input	
LUN Handle	Default
Use IO Manager	False
Automatic Offset	False
Initial Offset	512
Chunk Size	=@UP(0, G)
Total Transfer Size	=@UP(0, H)
Block Sequence	Forward
Number of Outstanding Requests	4
Type	10
Data Verification	
Verification Data Source	::Sequential()
Verification Data Offset	512
Verification Data Offset Type	Absolute

If, instead of a Verification Data Offset of 512, a value of 0 was used then the Data Verification would fail because the data read from 512 would be compared with the data from offset 0 of the ::Sequential file which will not be the same.

Custom SCSI Actions (Custom CDB Builder)

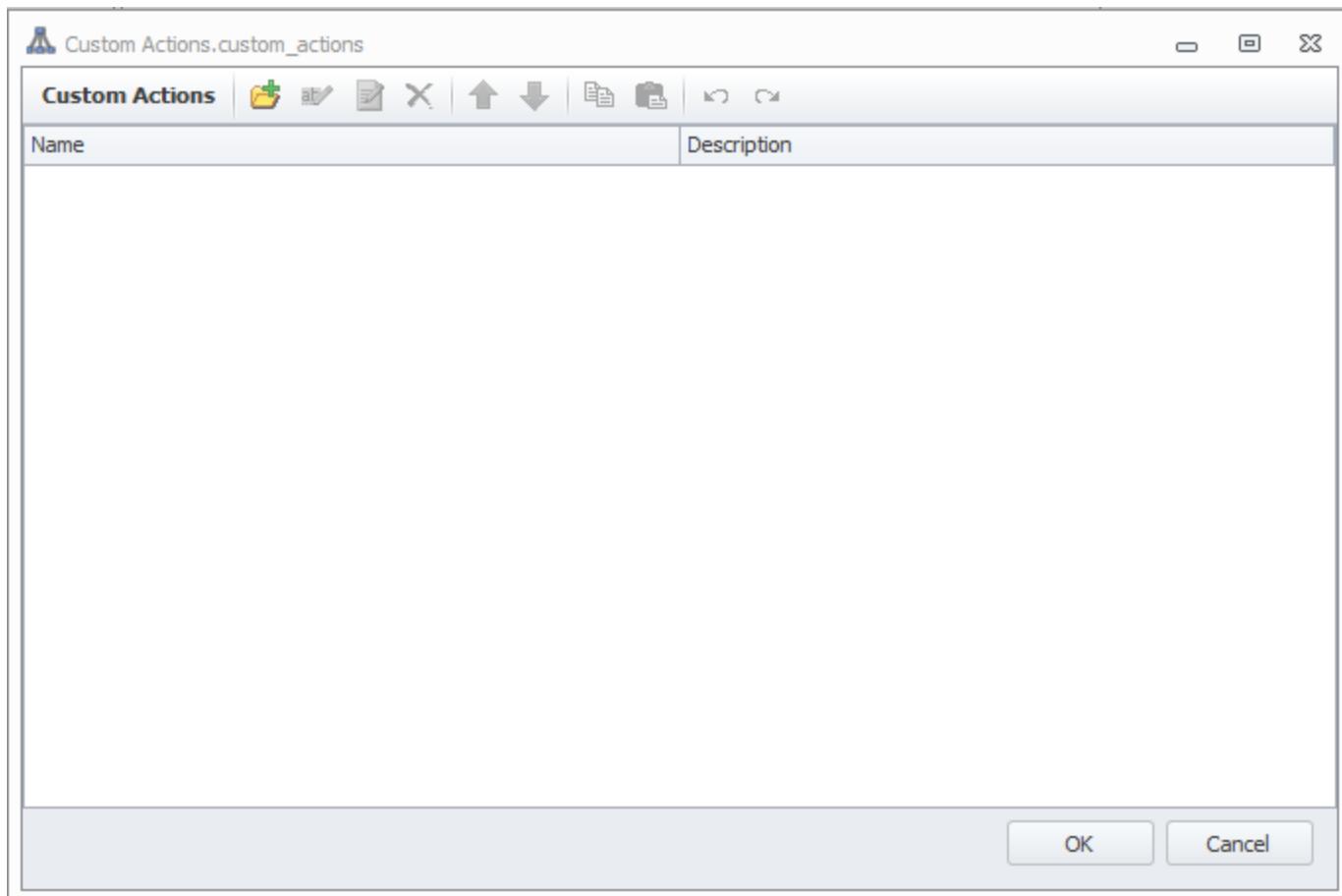
Testers that need to use SCSI commands not provided by default by the TDE or want to configure SCSI Actions to contain Project-specific default inputs, should use the Custom Actions feature of the Project Explorer. The Project Explorer does not come with Custom Actions in place by default so the Tester must first add the Custom Actions item to the Project Explorer using the Add New Item interface. Click on the Project Explorer Add New Item button or menu item. Highlight Custom Actions.



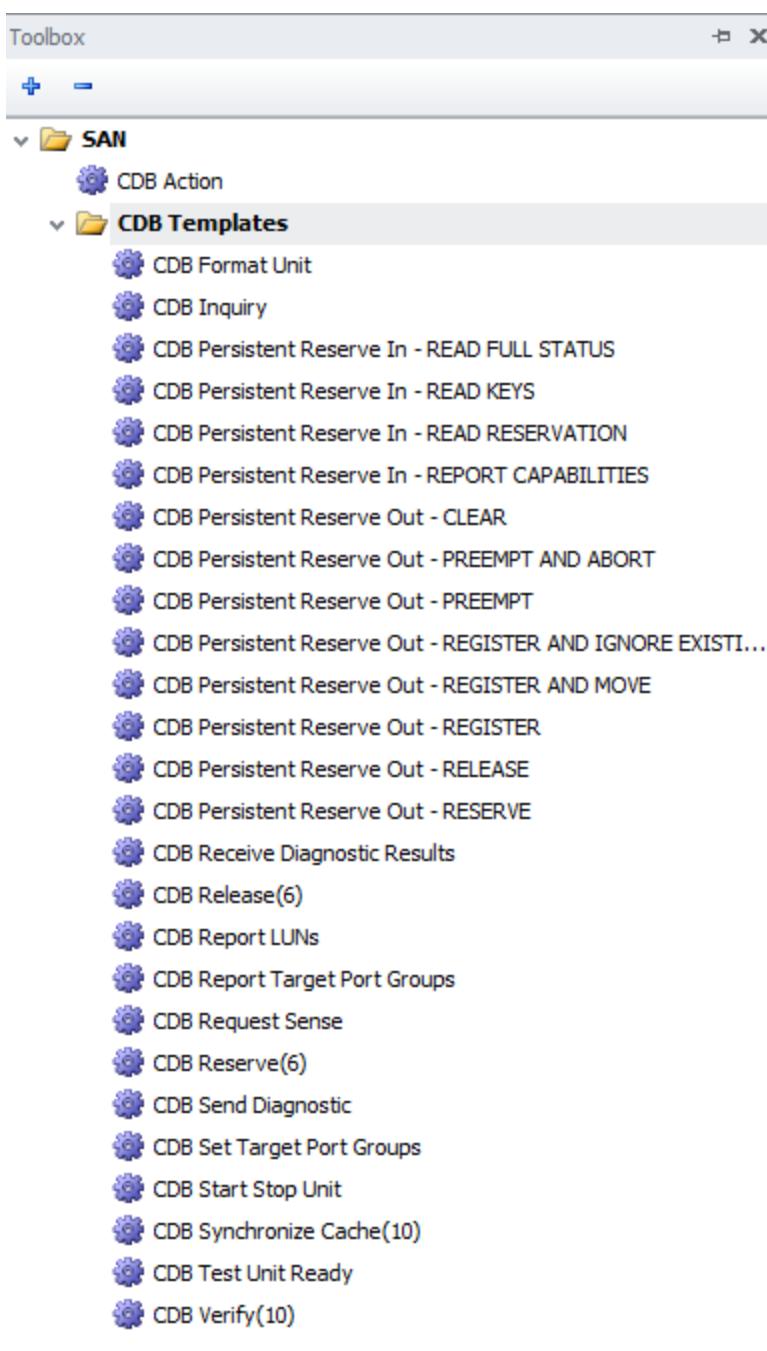
Now, click on the **Add** button to add Custom Actions to the Project Explorer.

Name	Type
One iSCSI Connection and Multiple LUNs	
Timeline	
User Parameters Map	
Dynamic Content	DynamicContent
Custom Actions	CustomActions
Project Resources	
Data File System1	DataFileSystem
Load Profile1	LoadProfile
Network Profile1	NetworkProfile
One Connection and Multiple LUNs	ClientScenario
Tracing Parameters1	TracingParameters
Unused Resources	
Client Scenario1	ClientScenario

Double click Custom Actions to open the Custom Actions repository. Initially it will be empty until Custom Actions have been added.



When the Custom Actions interface is open, the Toolbox will show Custom Action templates that can be used to create new Custom Actions.



To create a Custom Action based on an existing Action double click the template for that Action which will open the Custom Actions Editor. Make the changes necessary. To create a new SCSI Action (not based on any existing SCSI Action template),double click the CDB Action template.

Suppose that a Tester wanted to create three Test Unit Ready Actions for LUNs 0, 1 and 2 in which the LUN value is fixed.

To create the Test Unit Ready Action for LUN 2, first double click the Test Unit Ready template from the Toolbox into the Custom Actions editor and then edit that template to set the LUN Number Default Value to 2 and set the Editable field for Parameter Name LUN Number to False. Give the Action an appropriate Name like "Test Unit Ready 2" and click OK.

Action Editor | + X ↑ ↓

General Options

Name:	Test Unit Ready 2
Description:	Test Unit Ready

Parameters

Name	Value	Offset (bytes)
Connection Handle	Default	
Parameter Name	LUN Number	
Data Type	Integer	
Default Value	2	
Editable	False	
CDB		
Parameter Name	OPERATION CODE	0
Parameter Name	Reserved(1-4)	1
Parameter Name	CONTROL	5
Payload		

Once all three Custom Actions have been created:

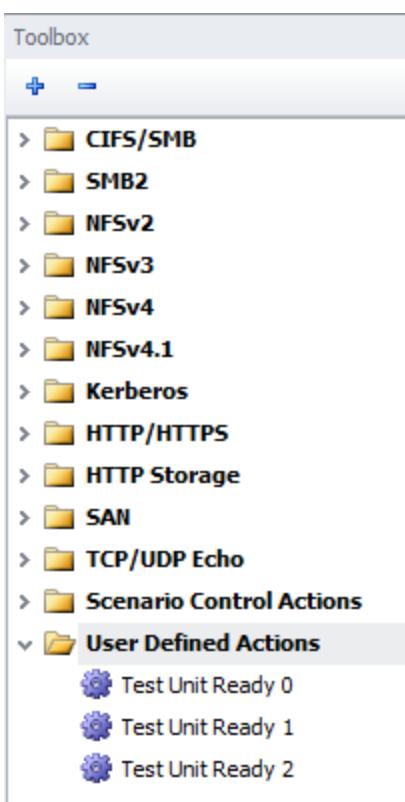
Custom Actions.custom_actions*

Custom Actions | + X ↑ ↓

Name	Description
Test Unit Ready 0	Test Unit Ready
Test Unit Ready 1	Test Unit Ready
Test Unit Ready 2	Test Unit Ready

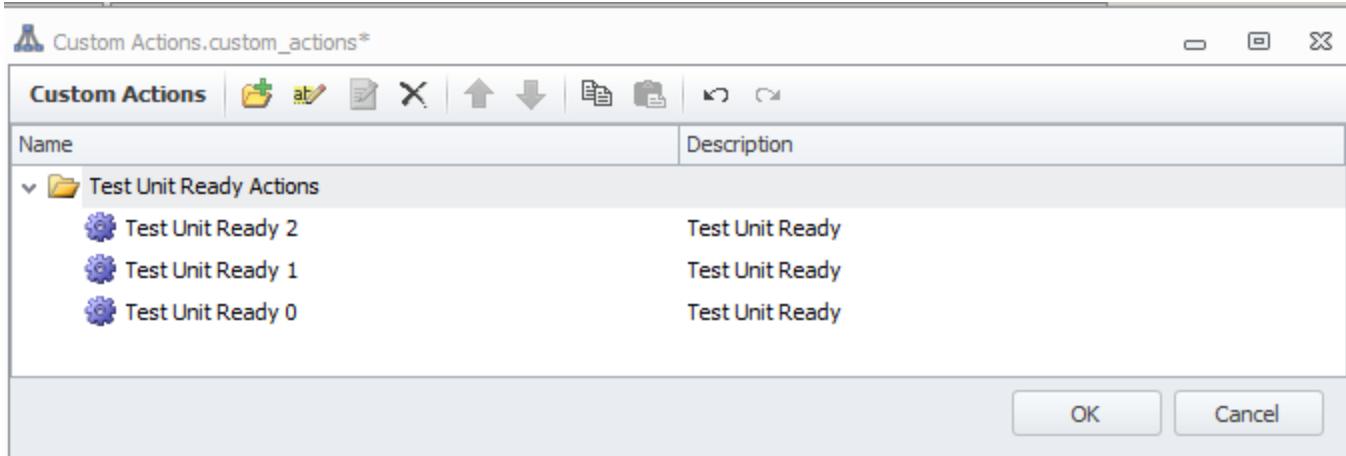
OK Cancel

These Custom Actions will also accessible in the Toolbox folder in User Defined Actions and can be dragged into Scenarios like all other Load DynamiX Actions.

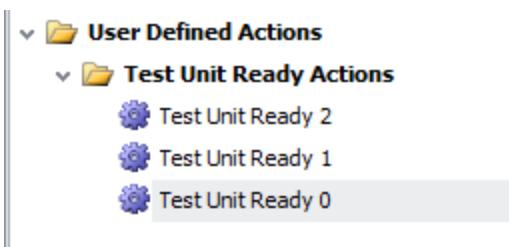


Grouping Custom Actions

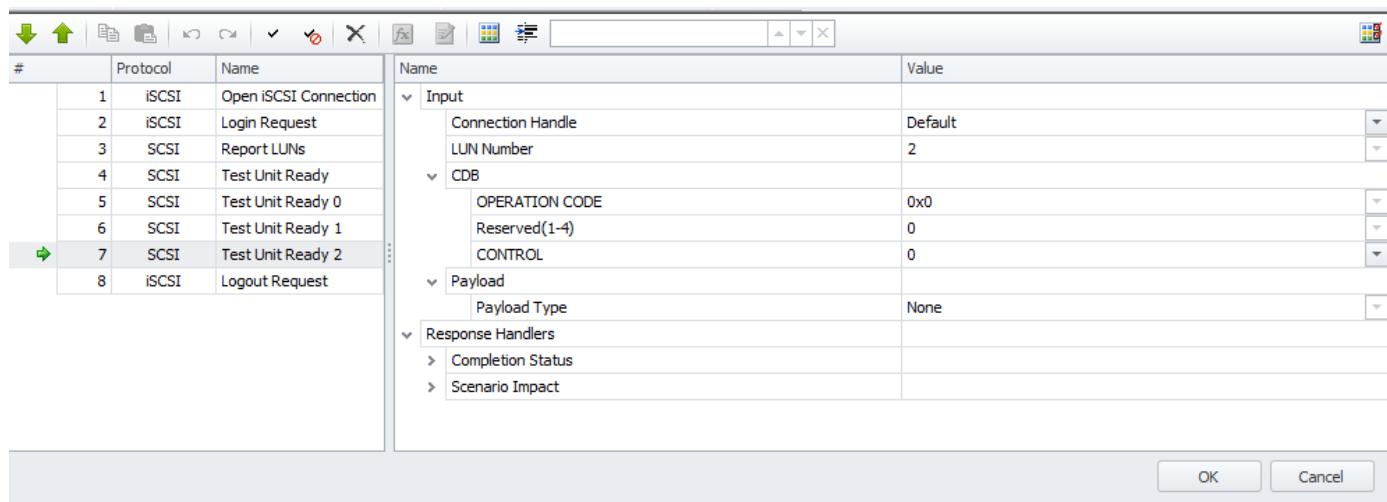
To create a Group ("folder") for Custom Actions, click the Add button in the Custom Action Editor. Once the Add button is clicked, a default folder named "Group" is created. Right click Group to rename it to something meaningful, such as Test Unit Ready Actions. All of the Test Unit Ready Actions created above could be collected in the Test Unit Ready Actions group to keep the repository of Custom Actions better organized. To move the Test Unit Ready Actions created above into the Test Unit Ready Actions group just drag and drop the Actions into the Group icon.



This change would appear in the Toolbox as



The Scenario below uses three Custom Actions and one non-custom Action to execute Test Unit Ready commands on LUNS 0,1 and 2. The non-custom (SCSI Toolbox default) Action Test Unit Ready has a default LUN value of 0 so LUN 0 is tested twice.



Custom Actions appear in Client Log file using the name given them when they are created in the Custom Action Editor

ISCSI Responses Handled:	Attempted	Succeeded
Total:	7	7
-----	-----	-----
ISCSI_LOGIN_REQ	1	1
ISCSI_LOGOUT_REQ	1	1
CDB_TEST_UNIT_READY	1	1
CDB_REPORT_LUNS	1	1
{CUSTOM} Test Unit Ready 0	1	1
{CUSTOM} Test Unit Ready 1	1	1
{CUSTOM} Test Unit Ready 2	1	1
-----	-----	-----

The PCAP for a the Test Unit Ready LUN #2 would look like

7 0.103379	01.01.00	01.14.00	FCP	36 SCSI: Response LUN: 0x01 (Test Unit Ready) (Good)
8 0.103382	01.14.00	01.01.00	FCP	56 SCSI: Test Unit Ready LUN: 0x02
9 0.103454	01.01.00	01.14.00	FCP	36 SCSI: Response LUN: 0x02 (Test Unit Ready) (Good)
10 0.103456	01.14.00	01.01.00	FCP	56 SCSI: Test Unit Ready LUN: 0x03
11 0.103659	01.01.00	01.14.00	FCP	36 SCSI: Response LUN: 0x03 (Test Unit Ready) (Good)

```
+ Frame 8: 56 bytes on wire (448 bits), 56 bytes captured (448 bits)
+ Fibre Channel
  FCP: FCP_CMND
    [Response In: 9]
      LUN: 0x02
      Command Ref Num: 0
      .... .000 = Task Attribute: simple (0x00)
      Task Management Flags: 0x00 (No values set)
      0000 00.. = Additional CDB Length: 0
      .... ..0. = RDDATA: False
      .... ...0 = WRDATA: False
      FCP_DL: 0
  SCSI CDB Test Unit Ready
    [LUN: 0x0002]
    [Command Set:Direct Access Device (0x00) (using default commandset)]
    [Response In: 9]
      opcode: Test Unit Ready (0x00)
  Control: 0x00
```

Re-Using Custom Actions

Custom Actions are saved per Project and not per TDE instance (e.g. changes made to Actions in one Project are not immediately available to other Projects). To share a set of Custom Actions, there are two simple steps necessary:

1. Drag the Custom Actions resource from the Project in which the custom CDBs were created down into the Resource Explorer, My Resources folder.
2. Open the Project that needs to use the Custom Actions and drag the Custom Actions resource from My Resources up into the Project Explorer of new Project.

NOTE: At most one Custom Actions resource in a Project at any time. There may be multiple different Custom Actions resources in My Resources as long as they all have unique names but dragging a different Custom Action resource into a Project that already has a Custom Action resource in it will replace the current Custom Action resource.

Key Features of the Custom Action Editor: Bit, Byte,Hex,Flag Parameters

Bit Parameter (a collection of 8 bits)

Name:	CDB Action Bit Boundary	
Description:		
Parameters		
Name	Value	Offset (byt...)
Connection Handle	Default	
Parameter Name	LUN Number	
CDB		
Parameter Name	Bit Bound Parameter	0
Data Type	Bitwise Group	
Size (bytes)	1	▲ ▼
Parameters	Count: 8	...
Bit7 {1 bits}	0	
Bit6 {1 bits}	1	
Bit5 {1 bits}	1	
Bit4 {1 bits}	0	
Bit3 {1 bits}	0	
Bit2 {1 bits}	1	
Bit1 {1 bits}	1	
Bit0 {1 bits}	1	

Parameter Lists

Flag Parameter (8 bits representing 8 flags)

Name:	CDB Action 8 Bit Flags	
Description:		
Parameters		
Name	Value	Offset (bytes)
Connection Handle	Default	
Parameter Name	LUN Number	
CDB		
Parameter Name	Flags Parameter	0
Data Type	Flag	
Size (bytes)	1	
Parameters	Count: 8	...
Enabled {0x01}	False	▼
8 Bit Flag {0x02}	True	▼
5 Bit Positive Integer Only {0x04}	True	▼
Integer High Order Bit 5 {0x08}	False	▼
Integer Bit 4 {0x10}	False	▼
Integer Bit 3 {0x20}	False	▼
Integer Bit 2 {0x40}	False	▼
Integer Low Order Bit {0x80}	False	▼

Byte Boundary Parameter (3 bytes, first byte has value 5, second and third have value 0)

Name:	CDB Action byte boundary	
Description:		
Parameters		
Name	Value	Offset (byt...)
Connection Handle	Default	
> Parameter Name	LUN Number	
CDB		
Parameter Name	Byte Bound Parameter	0
Data Type	Integer	
Size (bytes)	3	<input type="button" value="▼"/>
Default Value	5	<input type="button" value="▲"/>
Editable	True	<input type="button" value="▼"/>

Hex String Parameter (3 bytes containing a hex string, first byte contains 0x33, second and third bytes contain 0x00)

Name:	CDB Action hex strimng	
Description:		
Parameters		
Name	Value	Offset (bytes)
Connection Handle	Default	
> Parameter Name	LUN Number	
CDB		
Parameter Name	HEX String Parameter	0
Data Type	String	
Size (bytes)	3	<input type="button" value="▼"/>
Default Value	33	<input type="button" value="▲"/>
Mandatory	False	<input type="button" value="▼"/>
Editable	True	<input type="button" value="▼"/>

Grouped Parameters (Groups containing Groups)

Action Editor

X

Action Editor | + X ↑ ↓

General Options

Name:	groups in groups
Description:	

Parameters

Name	Value	Offset (bytes)
> <input checked="" type="radio"/> Parameter Name	Connection Handle	
> <input checked="" type="radio"/> Parameter Name	LUN Number	
✓ <input checked="" type="checkbox"/> Payload		
<input checked="" type="radio"/> Payload Type	None	<input type="button" value="▼"/>
✓ <input checked="" type="checkbox"/> Parameter Name	Group Parameter	
✓ <input checked="" type="checkbox"/> Parameter Name	Group Parameter1	
> <input checked="" type="checkbox"/> Parameter Name	Group Parameter2	
<input checked="" type="radio"/> Parameter Name	Size in Bytes Parameter	2
<input checked="" type="radio"/> Parameter Name	Size in Bytes Parameter1	3
Data Type	Size in Bytes	
Size (bytes)	1	<input type="button" value="▲"/> <input type="button" value="▼"/>
Default Value	0	
Editable	True	<input type="button" value="▼"/>

Payload

Payload field can be:

Payload Type None: no content

Payload Type Read: Payload Size (read length)

Payload Type Write:Payload Size (write length), DataSource (DialogContent or Structured)

DataContent

Action Editor

General Options

Name:	Write 16 Data Content Custom
Description:	

Parameters

Name	Value	Offset (bytes)
Connection Handle	Default	
Parameter Name	LUN Number	
CDB		
Parameter Name	OpCode	0
Parameter Name	Flags	1
Parameter Name	LBA	2
Parameter Name	Transfer Length	10
Parameter Name	Group	14
Parameter Name	Control	15
Payload		
Payload Type	Write	
Source Type	Data Content	
Content		
Payload Size	1024	
DataSource		

Structured

Action Editor

General Options

Name:	Write 16 Structured Content Custom
Description:	

Parameters

Name	Value	Offset (bytes)
Connection Handle	Default	
Parameter Name	LUN Number	
CDB		
Parameter Name	OpCode	0
Parameter Name	Flags	1
Parameter Name	LBA	2
Parameter Name	Transfer Length	10
Parameter Name	Group	14
Parameter Name	Control	15
Payload		
Payload Type	Write	
Source Type	Structured	
Content		
Structured Content		
Parameter Name	Byte Bound Parameter	0

Using key features of Custom CDB Builder to create a custom Read (6) Command

SCSI specification definition of Read(6)

Table 83 — READ (6) command

Bit Byte	7	6	5	4	3	2	1	0						
0	OPERATION CODE (08h)													
1	Reserved		(MSB)											
2	LOGICAL BLOCK ADDRESS													
3	(LSB)													
4	TRANSFER LENGTH													
5	CONTROL													

Custom CDB **MY READ (6)** (Read 1 512 byte chunk from LUN 0)

General Options

Name:	MY READ (6)
Description:	Read (6) Command Implemented by Custom CDB

Parameters

Name	Value	Offset (bytes)
> <input checked="" type="radio"/> Parameter Name	Connection Handle	
> <input checked="" type="radio"/> Parameter Name	LUN Number	
∨ <input checked="" type="radio"/> Parameter Name	OPCode	0
Data Type	Integer	
Size (bytes)	1	
Default Value	8	
Editable	False	
> <input checked="" type="radio"/> Parameter Name	LBA	1
> <input checked="" type="radio"/> Parameter Name	Transfer Length	4
> <input checked="" type="radio"/> Parameter Name	Control	5
∨ <input checked="" type="radio"/> Payload		
<input checked="" type="radio"/> Payload Type	Read	
<input checked="" type="radio"/> Payload Size	512	

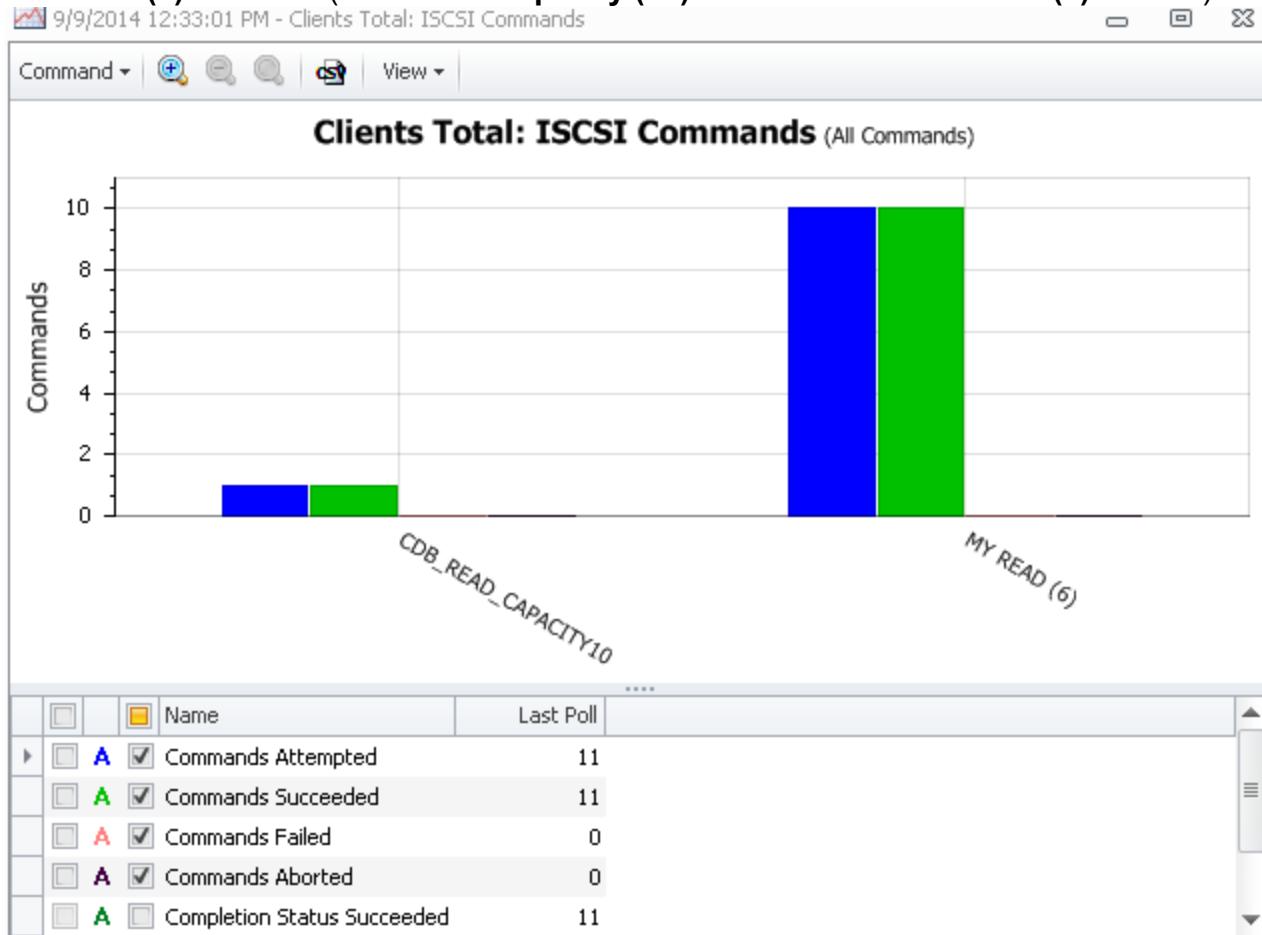
MY READ (6) Scenario (One **Read Capacity (10)** Action followed by 10 **MY READ (6)** Actions)

My Read 6.client_scenario

#	Protocol	Name
1	FC	Open FC Connection
2	SCSI	Read Capacity(10)
3	SWT	Begin Loop
4	SCSI	MY READ (6)
5	SWT	End Loop

Name	Value
Input	
Connection Handle	Default
LUN Number	0
OPCode	8
LBA	
_LBA	0
Reserved	0
transfer count	1
Control	
Payload	
Payload Type	Read
Payload Size	512
Response Handlers	
Completion Status	
Scenario Impact	

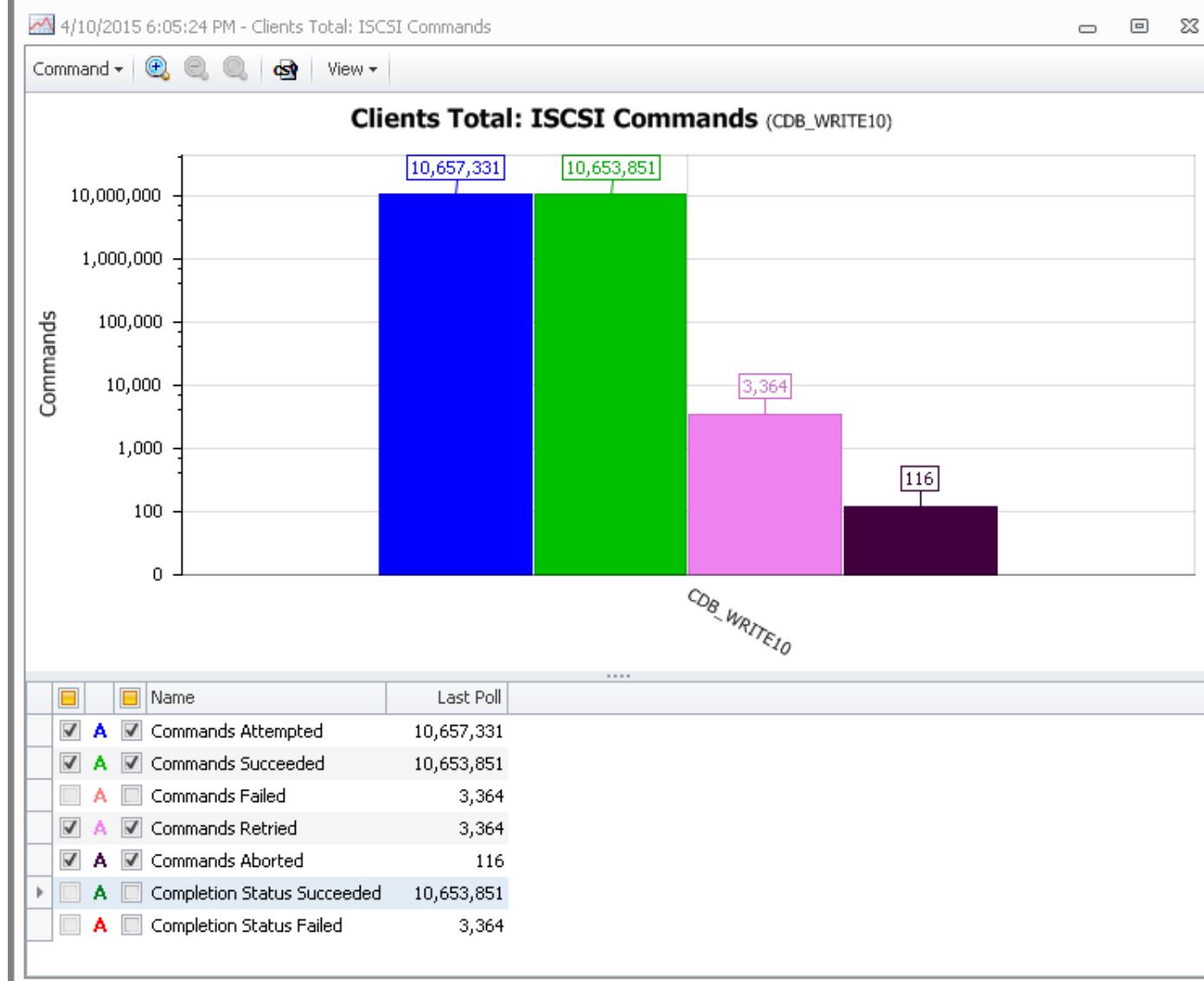
MY Read (6) Statistics (One Read Capacity (10) Action and 10 MY READ (6) Actions)



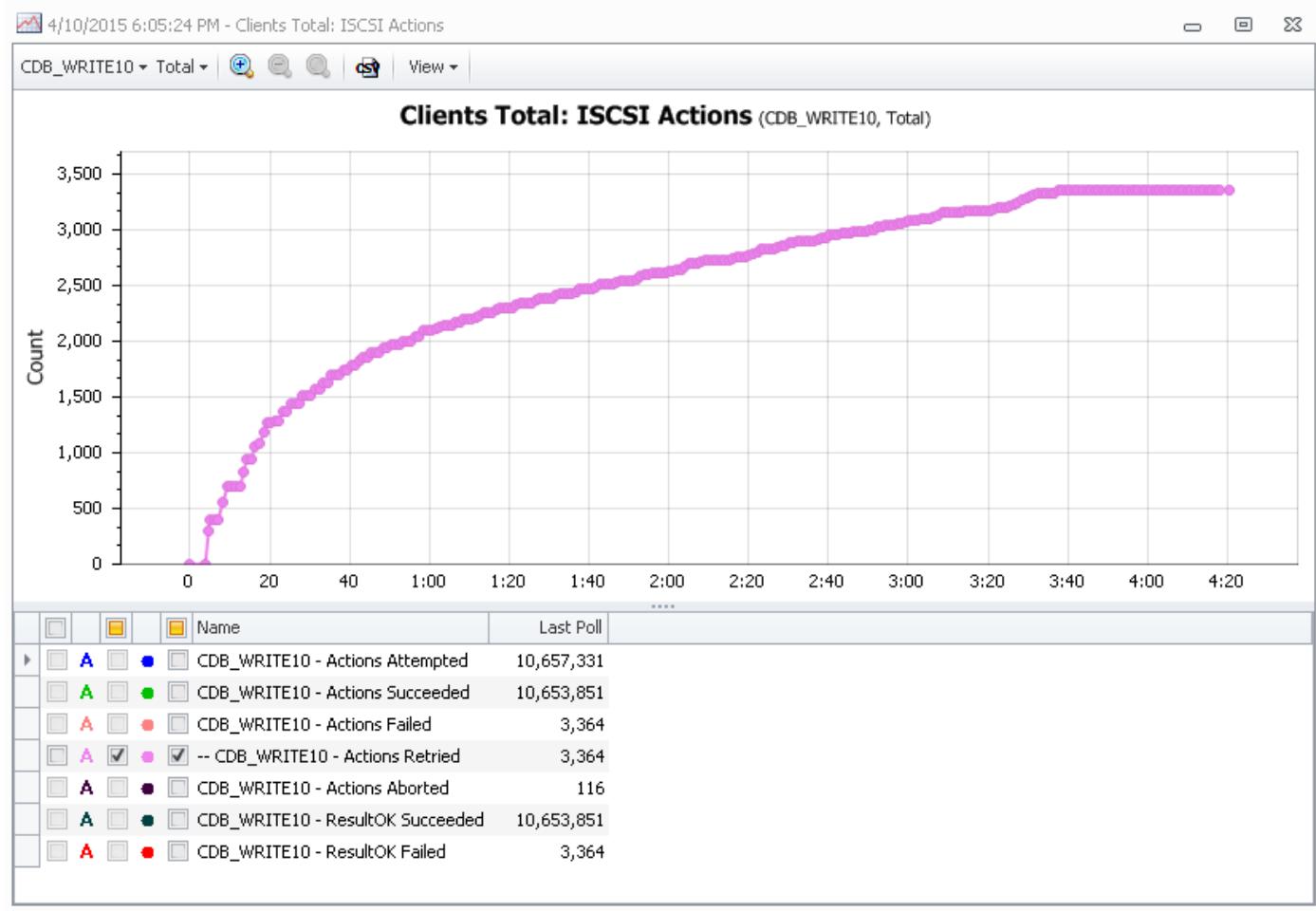
SCSI Retries

SCSI command retries are captured as part of the standard set of SCSI statistics. The Load DynamiX SCSI Firmware counts and accounts for SCSI command retries in the statistics and counters that are captured during the execution SCSI (iSCSI or Fibre Channel) Projects. SCSI Action Retries can be visualized in the following graphs:

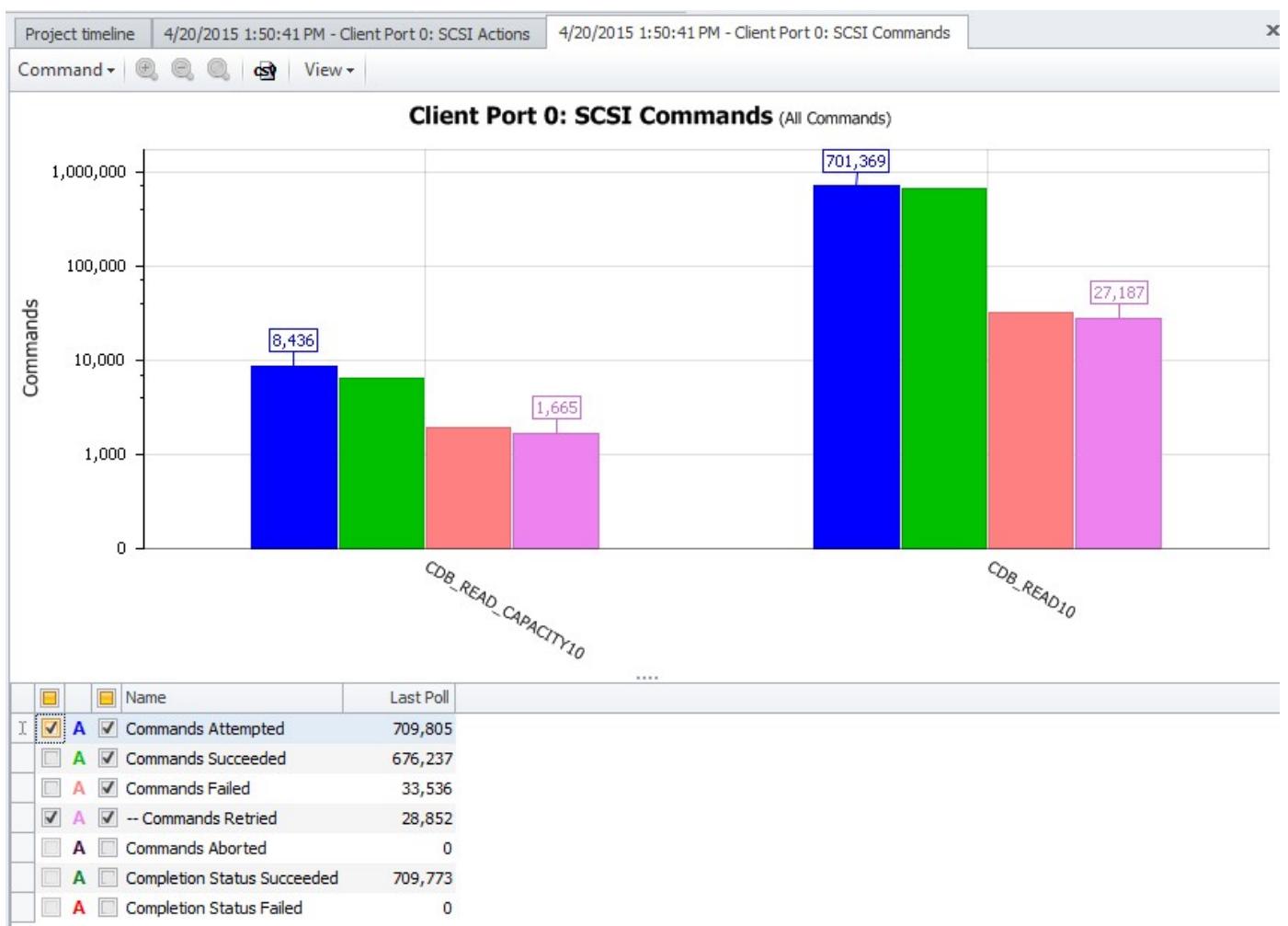
iSCSI Commands



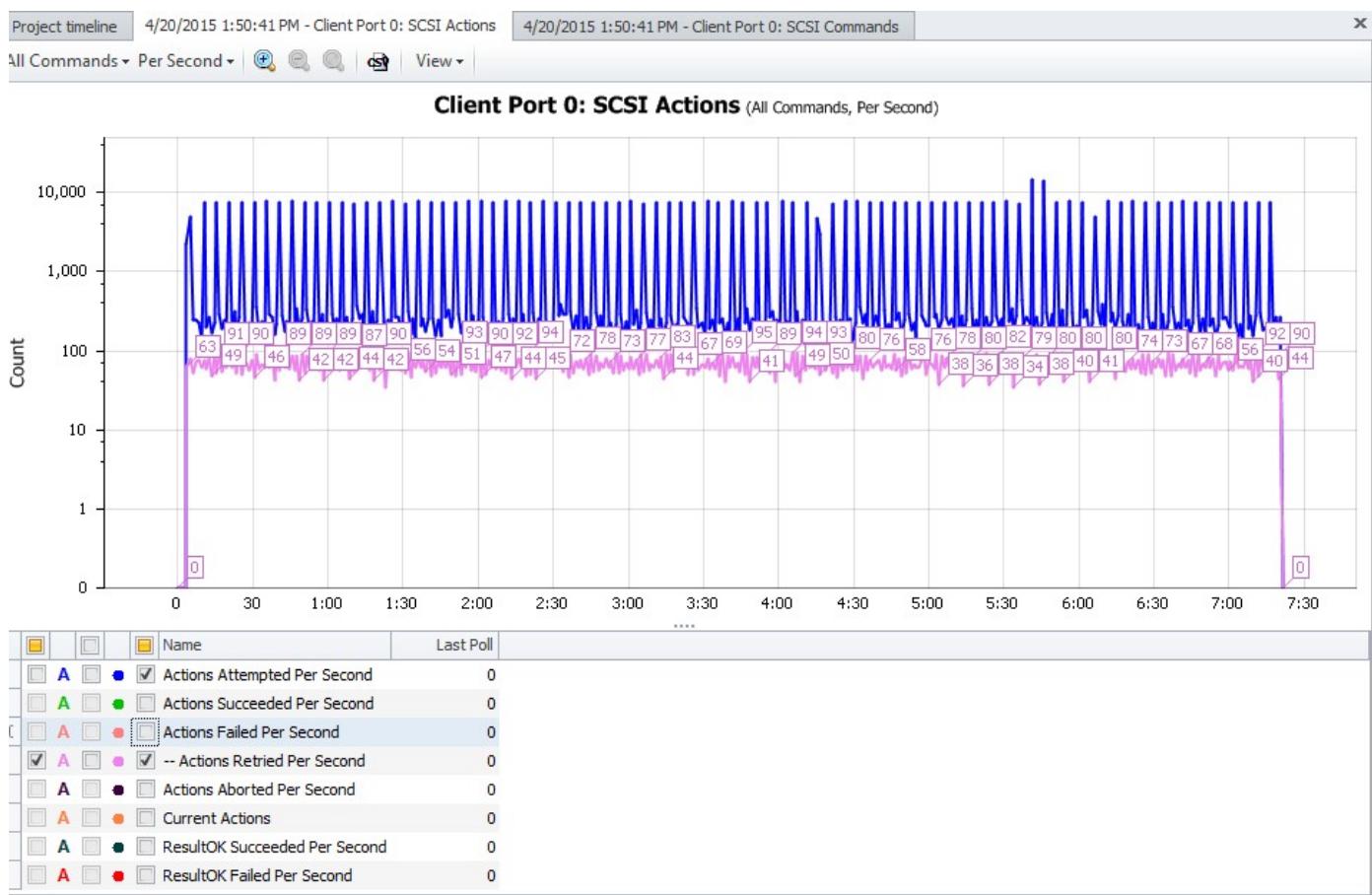
iSCSI Actions



Fibre Channel Commands



Fibre Channel Actions



Retries are also displayed in the Per Lun statistics shown in the Client Log file (as shown below).

The impact to iSCSI and Fibre Channel statistics as a result of capturing the Retry statistics are as follows:

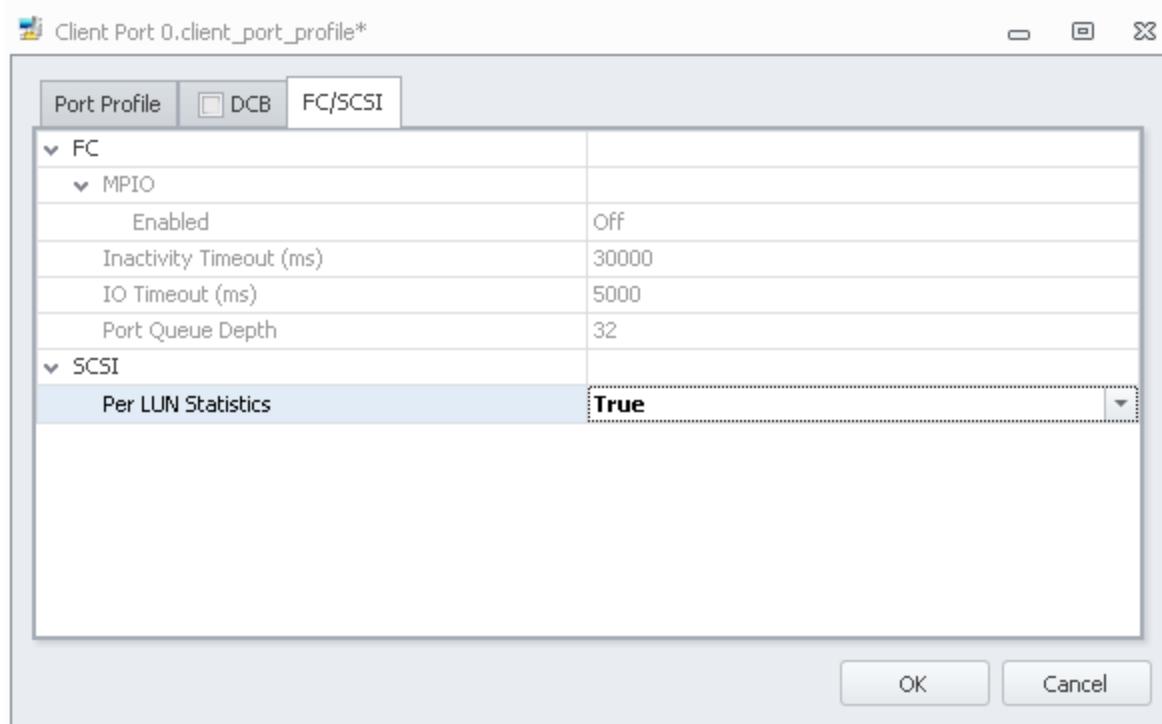
- Attempt counts includes Retries
- Failed counts includes Retries
- New Action Retries and Action Retries Per Second statistics have been added to the set of SCSI/iSCSI Test Execution Rules stats (see [Advanced Concepts Test Execution Rules](#) for more details)
- SCSI Action statistics captured starting in this release are not going to be identical to those captured in previous releases using the same Project
- Measured Latency will be slightly lower than the actual latency
- Byte and Packet counters include Retries

Non-impacted areas of Fibre Channel or iSCSI Projects are as follows:

- Succeeds are counted the same as in previous Firmware releases
- PCAP contents and capture are unaffected
- Load Profile behavior is not impacted

SCSI Per LUN Statistics

When set to True in the Port Profile, generates additional statistics in the Client Log file.



Per LUN statistics are generated in the Client Log File of an iSCSI or Fibre Channel Project. Per Lun Stats do not contain statistics for Actions that are defined in the iSCSI, Fibre Channel (FC) and MPIO/ALUA Toolbox folders. For example, the iSCSI Login, Open FC Connection and MPIO Open and Config Actions are not counted in Per Lun Stats.

=====					
SCSI Per LUN Statistics					
=====					
=====					
Initiator:	iqn.2009.09.com.SWFTT:snc002210				
Target:	iqn.1986-03.com.sun:02:99d9ab85-c310-4fe5-8c6d-979447aae809				
=====					
Transactions Total:	Attempted	Succeeded	Failed	Aborted	Retried
LUN0	1	1	0	0	0
LUN2	194	193	0	1	0
TOTAL	195	194	0	1	0
=====					
Bytes Total:		Transmitted	Received	Total	
LUN0		48	64	112	
LUN2		12297312	9280	12306592	
TOTAL		12297360	9344	12306704	
=====					

SCSI Check Condition Return Value Response Handling Processing using the Custom extension feature

The SCSI actions that might receive a CHECK_CONDITION response from server have an additional Completion Status field named "Custom". This "Custom" extension allows a Tester to configure the Completion status (Success or Failure) for the CHECK CONDITION response based on the KCQ (Key Code Qualifier) value related to the Check Condition response. This feature allows a Tester to specify up to 10 KCQ values and handle each response independently. A Tester also can specify a default Completion Status behavior to all other KCQ values which are not in the list of

defined KCQ's.

SCSI Initiator and Target Processing of Check Condition

When a SCSI Initiator (iSCSI or FC Scenario) sends a SCSI command to a SCSI Target, the response from the Target might contain the error code: Check Condition. The SCSI Initiator responds with a SCSI Request Sense Data command and embedded in the Request Sense Data response is the KCQ value (20 bits of data).

Field Name	#Bits	Value Range
Key ("K")	4	0 - 0x0E
ASC ("C")	8	0 - 0xFF
ASCQ ("Q")	8	0 - 0xFF

The Key ("K" value) defines the category of the Key Code Qualifier response codes.

K Value	Category
0x00	No Sense
0x01	Soft Error
0x02	Not Ready
0x03	Medium Error
0x04	Hardware Error
0x05	Illegal Request
0x06	Unit Attention
0x07	Data Protection
0x0B	Aborted Command
0xE	Other

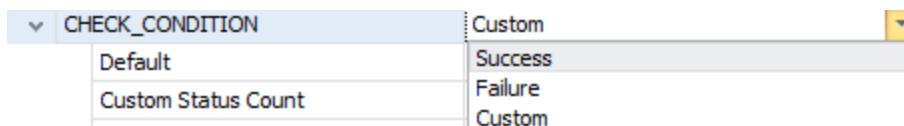
The "C" (ASC) and "Q" (ASCQ) values further define the meaning of the Request Sense data. See [Wikipedia KCQ Value Table](#) for details on ASC and ASCQ values. This Request Sense (FCQ) information is made available to the Tester via the Custom extension of the Check Condition response handler so that the Tester can apply more fine grained control of error handling in a FC/iSCSI Scenario. The following iSCSI example shows how the Custom exception feature can be used.

SCSI Check Condition Custom Extension Example

This Scenario connects to two LUNs (Lun #0 and Lun #1) on the SCSI Target opened in line #1 and in a loop, Reads from Lun #0 and Writes to Lun #1. Using the Custom extension to the Check Condition response handler for the Read, the Tester can check for several responses that indicate Read Errors that succeed with error correction but nonetheless caused the return of the Check Condition error. The three cases that are singled out in this example Scenario are:

1. KCQ = 0x0/0x18/0x0, Recovered read error using error correction
2. KCQ = 0x0/0x18/0x1, Recovered read error using error correction and retries
3. KCQ = 0x0/0x18/0x7, Recovered read error using error correction, data rewritten

In the Completion Status field of the Read(10) Action, the Check Condition status is set to "**Custom**" using the drop down menu.

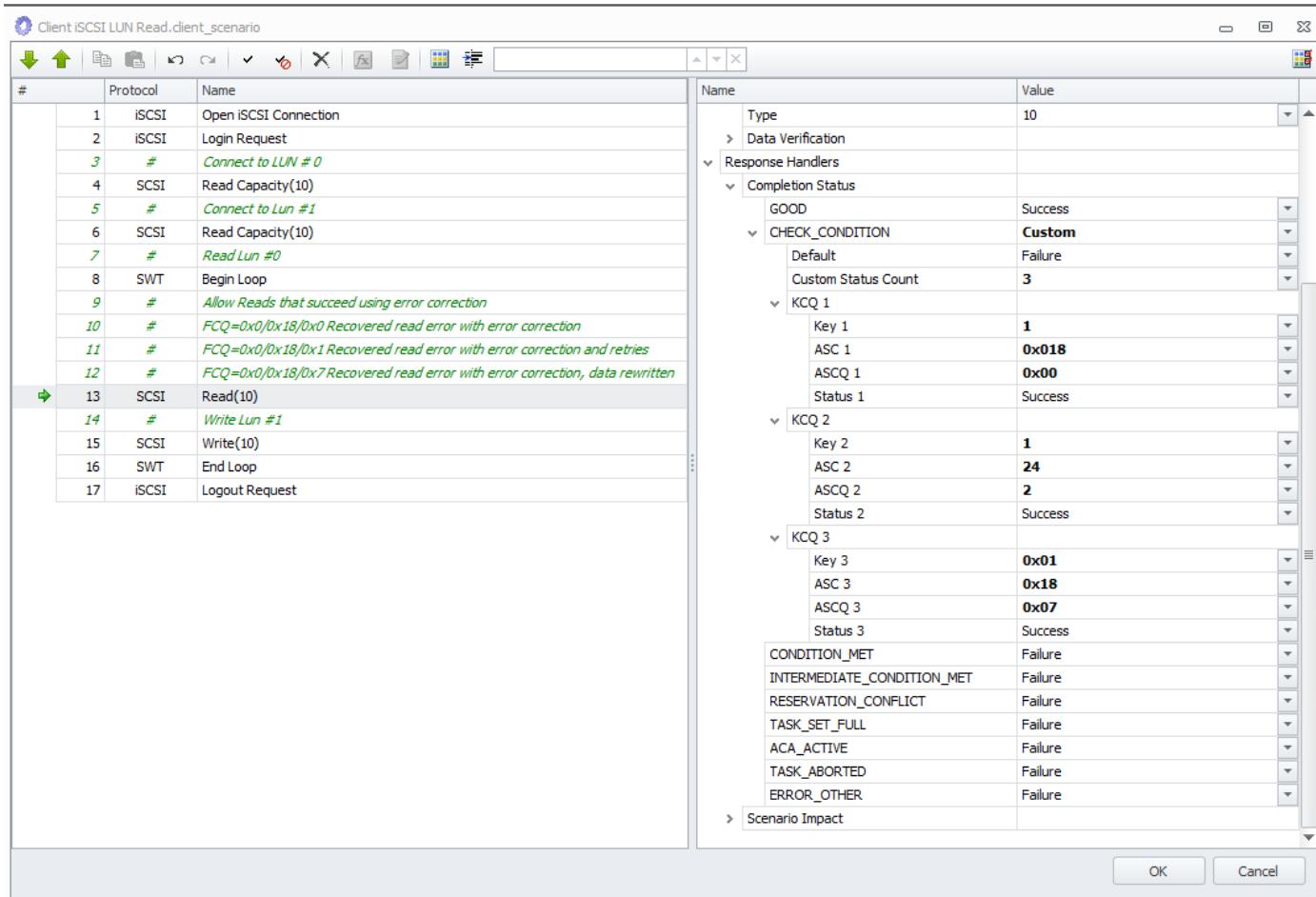


The default value of the Custom Status Count field is 1. This input controls the number of KCQ

values that will be handled independently of the other KCQ values that might be returned. The Custom Status Count field is set to 3 (maximum 10) using the drop down menu for that input field, to create three KCQ input fields. Each KCQ field has four inputs.

1. Key - the K value of the response
2. ASC - the C value of the response
3. ASCQ - the Q value of the response
4. Status - how this KCQ value is to be interpreted (Success or Failure)

In the example below, three KCQ values are flagged as Success. All of the other KCQ values that might be returned are to be treated using the Default status which in this case is Failure.



KCQ field inputs

Notice that the input values to the KCQ fields can be specified as integer or hexadecimal values. Without the 0x... preceding a value, the input is interpreted as an integer.

Custom Extension Stats

Like all Response Handlers, the Custom Extension results appear in the Action stats and log file. See the [Advanced Concepts: Response Handling](#) for all general Response Handling feature information.

SAN Actions that do not support the Custom extension

The following list of SAN Actions do not receive Check Condition responses from SCSI Servers and, thus, do not support the Custom extension processing described above.

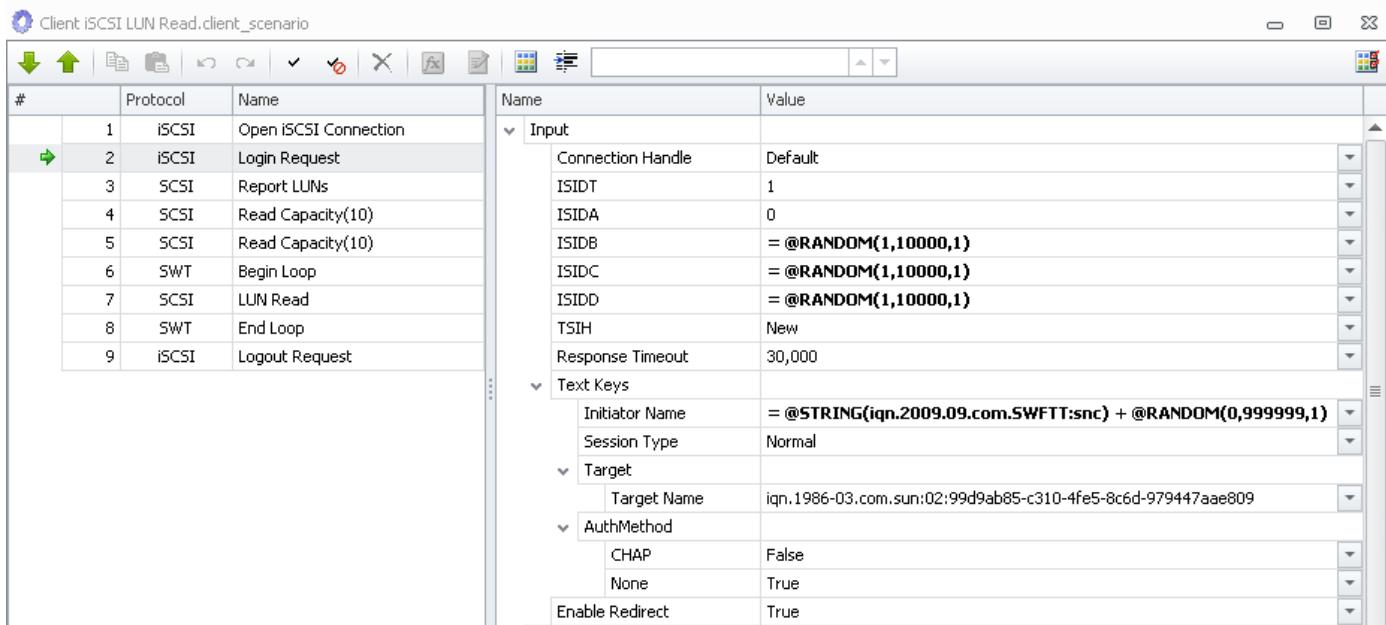
iSCSI Protocol	Fibre Channel Protocol	MPIO/ALUA Protocol
Open iSCSI Connection	Open FC Connection	MPIO Open and Config
Close iSCSI Connection	Close FC Connection	MPIO Config
NOP-Out		
Login Request		
Logout Request		
Text Request		

iSCSI

Load DynamiX iSCSI Protocol support allows the Tester to send and receive SCSI Block Protocol commands over an TCP/IP transport to iSCSI/SCSI compatible devices.

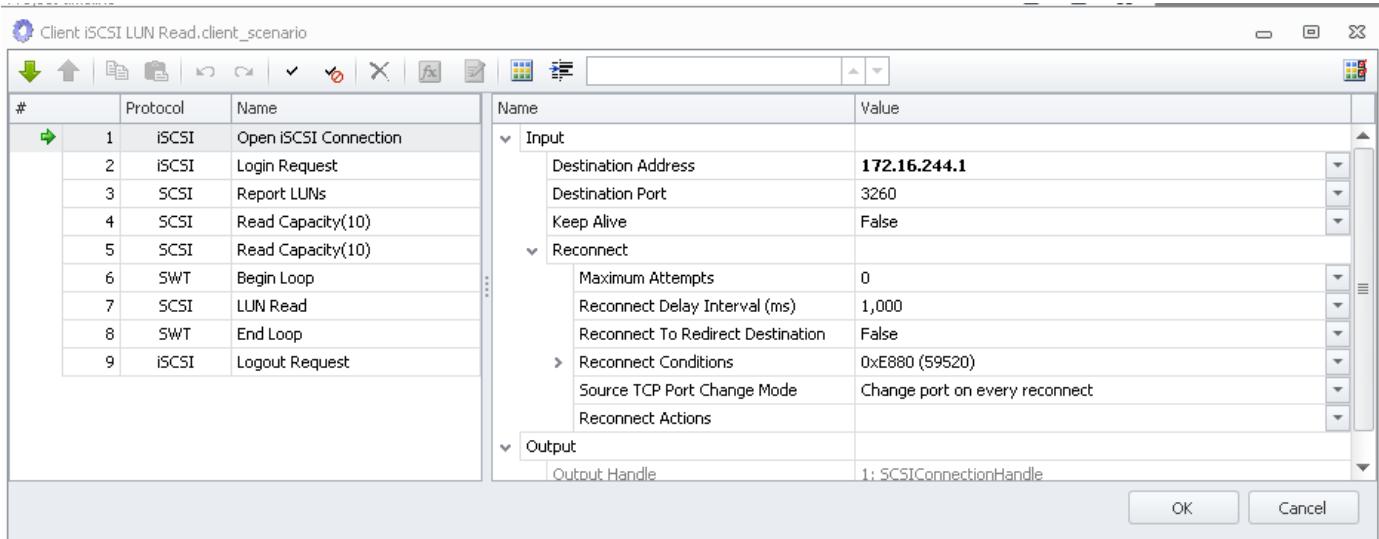
iSCSI Login Request

The **iSCSI Login Request** Action allows the Tester to define the kinds of authentication that it will support: CHAP or None. Both forms of authentication can be True or False so that the Client can handle Server responses that allow both or either.



An example iSCSI Scenario would contain an **Open iSCSI Connection**, followed by the iSCSI **Login Request** and then a set of SCSI Block Protocol commands. See [Reference: iSCSI Commands and Behaviors](#) for more details on iSCSI authentication.

An iSCSI Logout Request will close the Scenario.



iSCSI MPIO

The Load DynamiX Appliance supports MPIO and ALUA operations over a single iSCSI Physical Test Port (as opposed to FC MPIO which allows multiple FC Physical Test Ports to be used). iSCSI MPIO does not require the Port-level MPIO control (Enable On or Off) that FC does require. Since iSCSI MPIO only supports a single Physical Test Port, multiple connections and multiple logins will be used to open paths to the LUNs that will be accessed in an iSCSI MPIO Scenario. It is recommended that the MPIO/ALUA section below, which explains MPIO and ALUA behaviors in greater detail, be read before proceeding to review the iSCSI MPIO example.

iSCSI MPIO Fail Over example

See the iSCSI Scenario below for an example of iSCSI MPIO Fail Over. The iSCSI server targeted by this Scenario has been configured to have multiple ALUA groups, "east" and "west". LUN 0 belongs to the "east" group and LUN 1 belongs to the "west" group.. LUN 0 and LUN 1 share the same physical storage so that access to one is the same as access to the other. In the **MPIO Config** Action in Line 15, the Primary path is configured to have Path ID 10 and access to LUN 0, the Secondary path is configured to have Path ID 20 and access to LUN 1. The MPIO policy is set to Fail Over Only which will cause the Secondary path to take over whenever the Primary path is not operational.

Client iSCSI LUN Read.client_scenario*

#	Protocol	Name
1	#	
2	iSCSI	Open iSCSI Connection
3	iSCSI	Login Request
4	SCSI	ALUA Discovery
5	#	<i>Identify LUN 0, LUN 0 belongs to "east" ALUA group</i>
6	SCSI	Read Capacity(10)
7	iSCSI	Open iSCSI Connection
8	iSCSI	Login Request
9	SCSI	ALUA Discovery
10	#	<i>Identify LUN 1, LUN 1 belongs to "west" ALUA group</i>
11	SCSI	Read Capacity(10)
12	#	<i>Primary path (path 10) uses the "east" ALUA group</i>
13	#	<i>Secondary path (path 20) uses the "west" ALUA group</i>
14	#	<i>LUN 0 and LUN 1 are the same physical storage</i>
15	SCSI	MPIO Config
16	#	<i>Any disruption of primary path during the read operations</i>
17	#	<i>Will result in secondary path being utilized</i>
18	#	<i>Until the primary path has been restored</i>
19	SWT	Begin Loop
20	SCSI	LUN Read
21	SWT	End Loop
22	iSCSI	Logout Request
23	iSCSI	Close iSCSI Connection

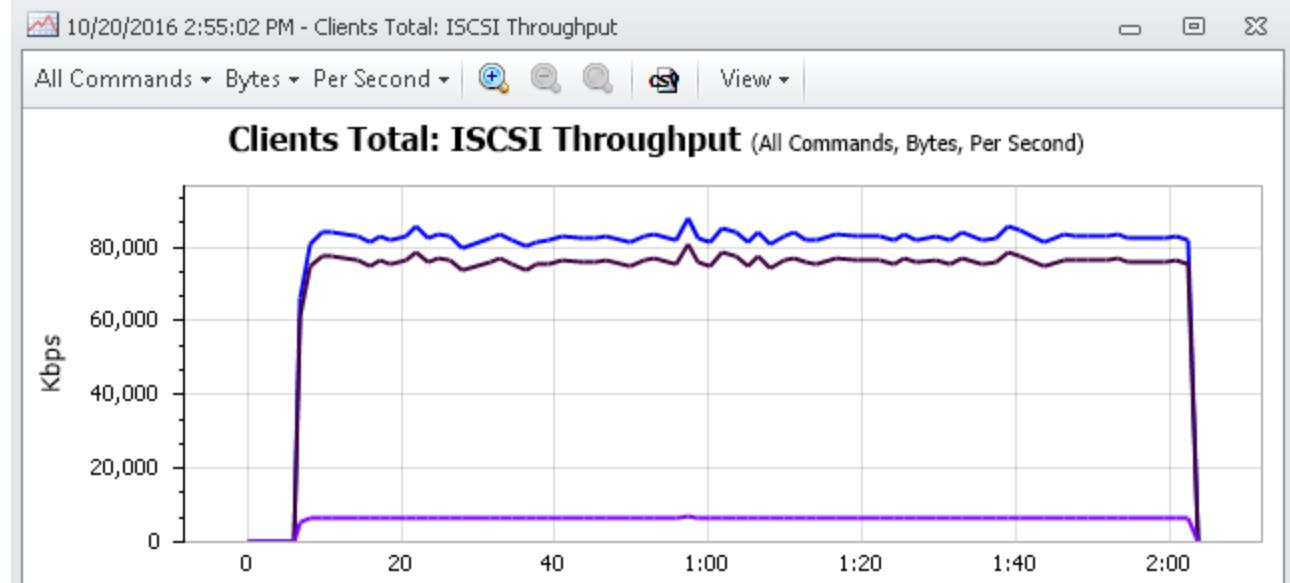
Name Value

- Input
 - Alias
 - Policy Fail Over Only
 - Enable ALUA Reconfig True
- Primary Path
 - Primary Path LUN 6: LUNHandle
 - Primary Path ID 10
- Secondary Path
 - Secondary Path LUN 11: LUNHandle
 - Secondary Path ID 20
 - Additional Paths 0
- Output
 - Output Handle 15: LUNHandle

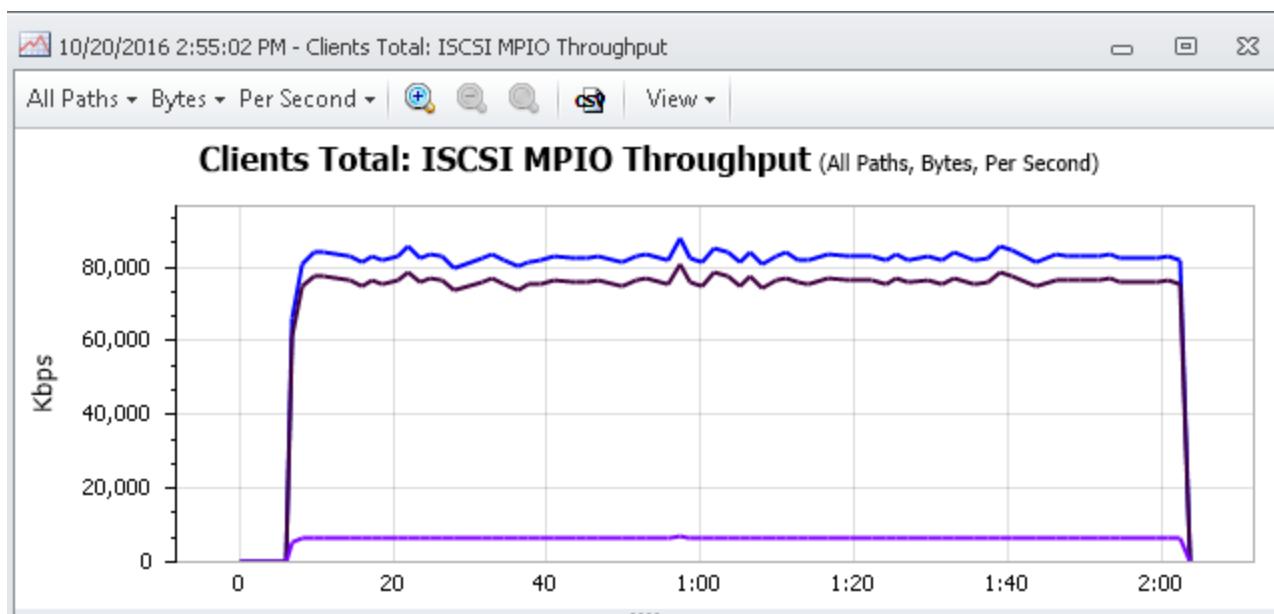
OK Cancel

During the execution of this Scenario, the Primary path to the target iSCSI server was made non-operational for short periods of time. The graphs below show iSCSI Throughput during the execution of this Scenario.

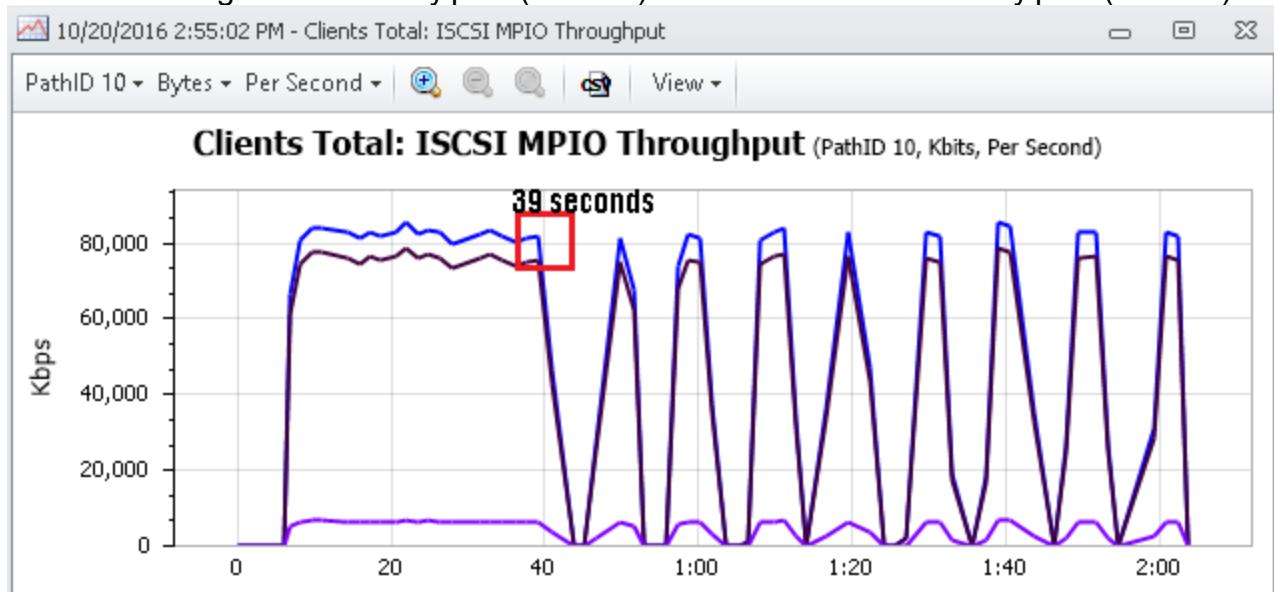
First iSCSI Throughput, Bytes Per Second, with no MPIO filters



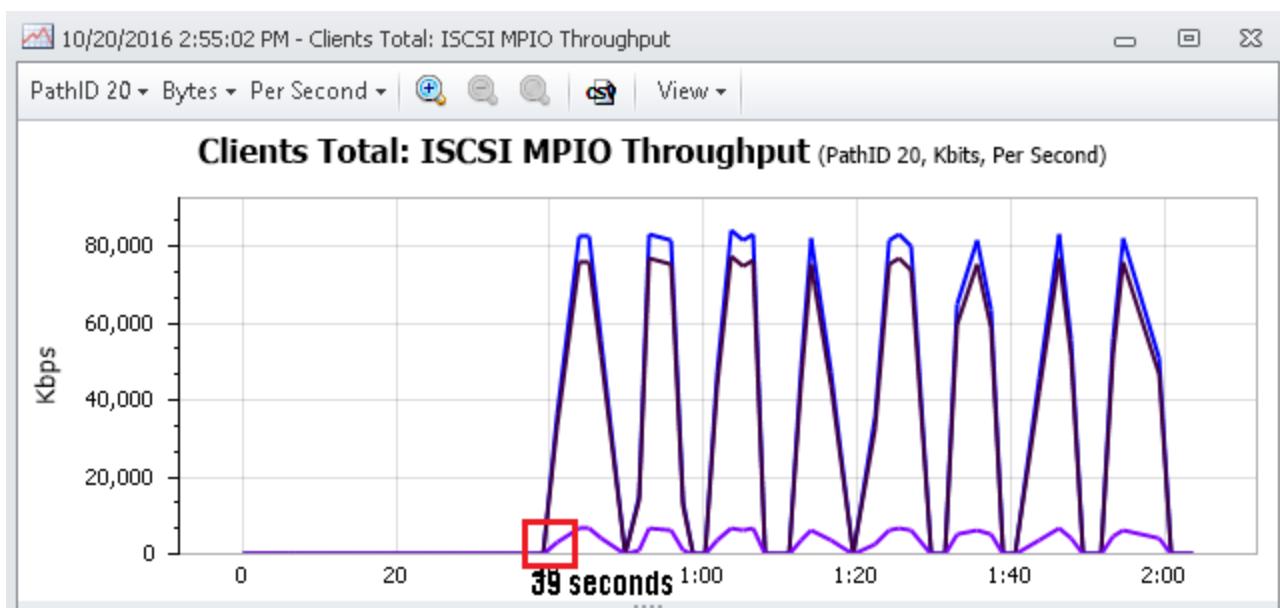
Second, iSCSI Throughput for all MPIO paths. Notice that these graphs are identical so that when all MPIO paths are considered, the MPIO Throughput is equal to iSCSI Throughput.



The Primary Path was made non-operational at various times during the execution of this Scenario forcing the Secondary path (Path 20) to take over for the Primary path (Path 10)



The Secondary path Throughput graph shows that when the Primary path starts to go off line at (for example) at 39 seconds into the Scenario execution, the Secondary Path takes over until the Primary Path recovers.



The Client Log MPIO statistics show that there were 9 Fail Overs and 8 Fail Backs as seen in the graphs above.

MPIO Failover/Failback:	Failover	Failback Add'l Transition	
<hr/>			
Total:	9	8	0
<hr/>			
MPIO_path_10	9	8	0
<hr/>			

iSCSI Reconnect

See [Reference: iSCSI Commands and Behaviors](#).

iSCSI Redirect

When iSCSI Scenarios connect to an iSCSI device, it is possible for the device to redirect the connection to a different device. The Tester can control whether or not the Scenario will respond to the redirection by setting the Login Request input field Enable Redirect to True to allow redirection (default) or False to not allow redirection.

The screenshot shows the 'Client iSCSI LUN Read.client_scenario' configuration window. On the left, a list of actions is shown:

#	Protocol	Name
1	iSCSI	Open iSCSI Connection
2	iSCSI	Login Request
3	SCSI	Report LUNs
4	SCSI	Read Capacity(10)
5	SCSI	Read Capacity(10)
6	SWT	Begin Loop
7	SCSI	LUN Read
8	SWT	End Loop
9	iSCSI	Logout Request

On the right, parameters are defined in a table:

Name	Value
Input	
Connection Handle	Default
ISIDT	1
ISIDA	0
ISIDB	= @RANDOM(1,10000,1)
ISIDC	= @RANDOM(1,10000,1)
ISIDD	= @RANDOM(1,10000,1)
TSIH	New
Response Timeout	30,000
Text Keys	
Initiator Name	= @STRING(iqn.2009.09.com.SWFTT:snc) + @RANDOM(0,999999,1)
Session Type	Normal
Target	
Target Name	iqn.1986-03.com.sun:02:99d9ab85-c310-4fe5-8c6d-979447aae809
AuthMethod	
CHAP	False
None	True
Enable Redirect	True

iSCSI Statistics

See the [Advanced Concepts: Test Execution Rules](#) for a complete list of the statistics generated when iSCSI Actions are used in a Scenario.

Fibre Channel

Load DynamiX Fibre Channel Protocol support allows the Tester to send and receive SCSI Block Protocol commands over a Fibre Channel transport to FC/SCSI compatible devices.

Fibre Channel Logical Port Resource

The screenshot shows the 'Client Port 0.client_port_profile*' configuration window. It has tabs for Port Profile, DCB, and FC/SCSI. The FC/SCSI tab is selected and displays the following settings:

FC	
MPIO	
Enabled	Off
Inactivity Timeout (ms)	30000
IO Timeout (ms)	5000
Port Queue Depth	32
SCSI	
Per LUN Statistics	True

At the bottom are OK and Cancel buttons.

MPIO Enabled : Off/Pair/All (see below for more detailed MPIO discussion)

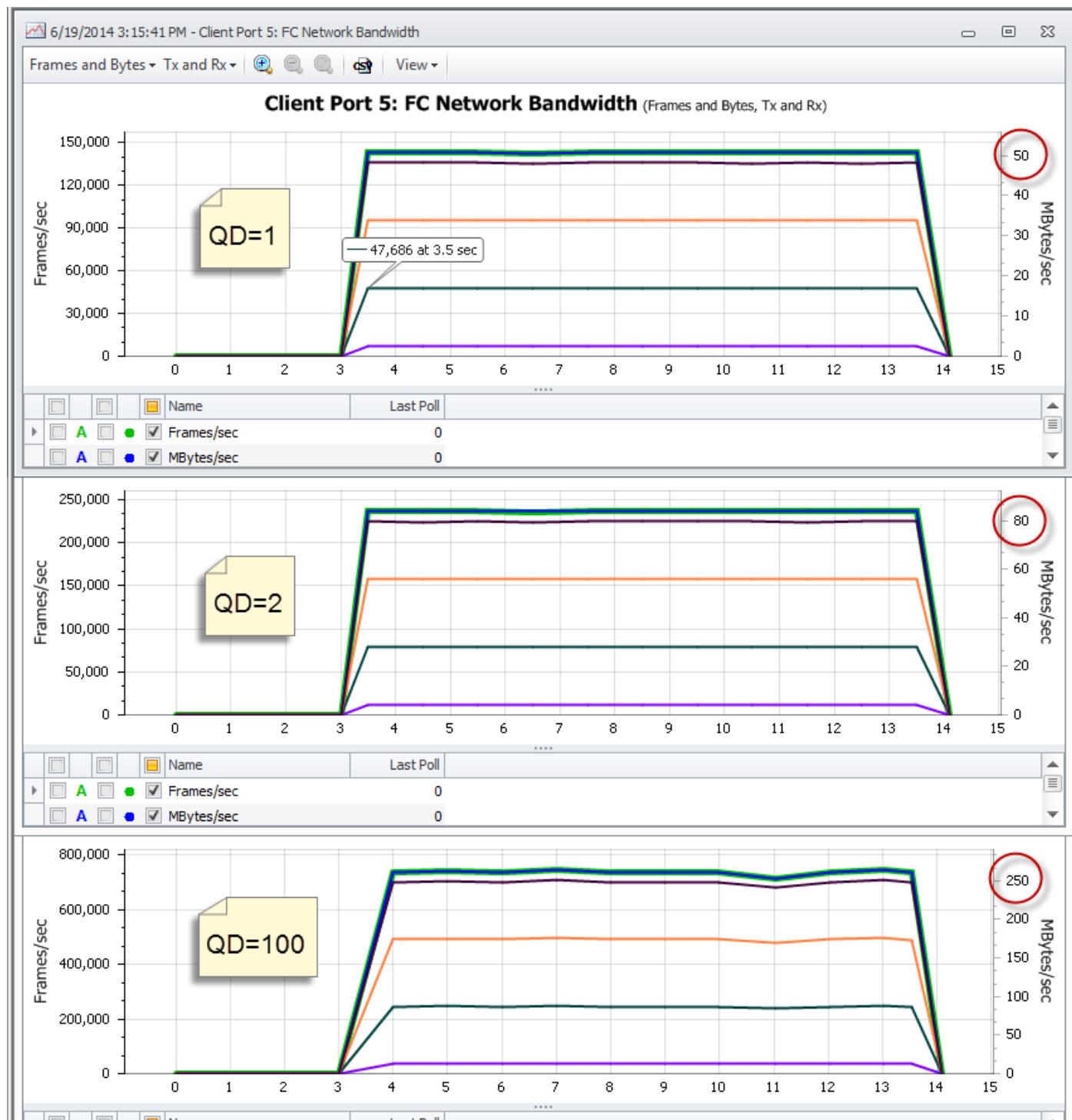
Inactivity Timeout(ms): 30000 ms default (see below for more detailed discussion).

IO Timeout (ms): 5000 ms default (see below for more detailed discussion).

Per LUN Statistics: False default (see above for more detailed discussion).

Port Queue Depth: Controls the maximum number of SCSI commands that can be pending on a Target for this Logical Port. It may control the command queue for a single Fibre Channel physical port or it may control command depth for multiple Fibre Channel physical ports if MPIO is enabled. Port Queue Depth allowable range is 1 to 2048.

Port Queue Depth setting can have a major impact on performance. See the graphs below that demonstrate the impact on Bandwidth utilization when different Port Queue Depth settings are used.



Fibre Channel Sample Scenario

An example FC Scenario would contain an Open FC Connection then a set of SCSI Block Protocol commands. A Close FC Connection Action will close the Scenario. Target WWPN is the target Fibre Channel device. Source WWPN (is supplied) is the Initiator WWPN (otherwise it is the Initiator WWPN of the Port that the Project is using). The Connect Timeout input is the time in milliseconds between attempts to connect to the Target WWPN.

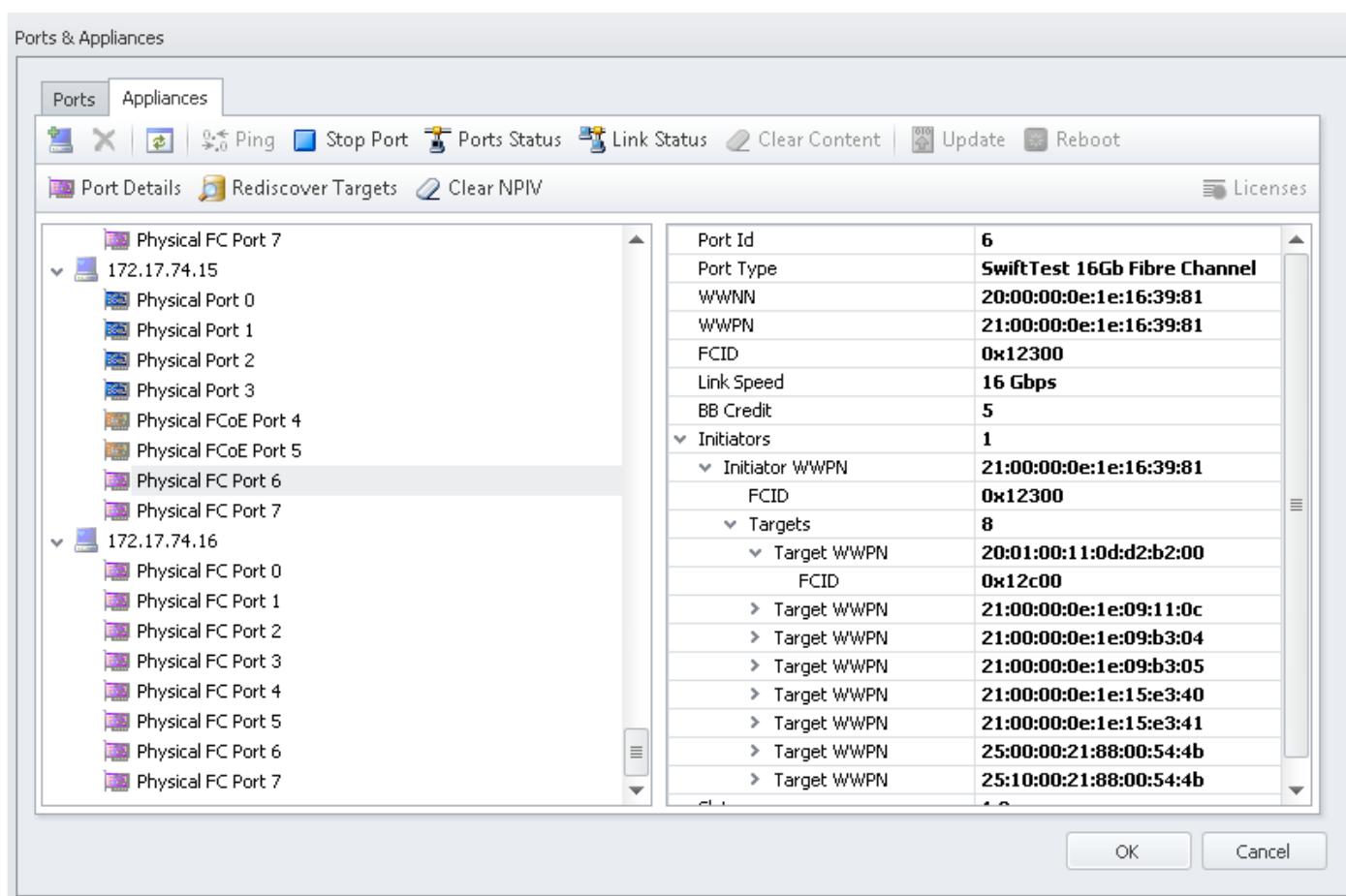
The example below demonstrates the form of the Open FC Connection Action in which the Source WWPN is the WWPN of the port on which this Scenario is running.

#	Protocol	Name		Name	Value
1	FC	Open FC Connection		Source WWPN Defined	False
2	SCSI	Read Capacity(10)		Target WWPN	=@UP(0, TARGET0)
3	SCSI	Test Unit Ready		Connect Timeout (ms)	30,000
4	SCSI	Test Unit Ready			

If it is necessary to use a different Source port WWPN (as in an MPIO Scenario described below), the alternate form of the Open FC Connection Action would be used in which the Source WWPN is specified in the Open FC Connection Action as shown below. The Source WWPN input field is exposed by setting the Source WWPN Defined field to **True**. In the example below, the Source WWPN input comes from a User Parameter file with a column labeled **PORT0**.

#	Protocol	Name		Name	Value
1	FC	Open FC Connection		Source WWPN Defined	True
2	SCSI	Read Capacity(10)		Initiator WWPN	=@UP(0, PORT0)
3	SCSI	Test Unit Ready		Target WWPN	=@UP(0, TARGET0)
4	SCSI	Test Unit Ready		Connect Timeout (ms)	30,000
5	SCSI	Test Unit Ready			

The Target WWPN value comes from the Ports & Appliances > Appliances Tab for the Fibre Channel capable Load DynamiX Appliance (the Load DynamiX 6202/6204/6208/6202E Appliances). As soon as the Fibre Channel interface on the Load DynamiX 6202/6204/6208/6202E Appliance is connected to a Fibre Channel capable device (switch or server) and the Appliance is powered on, it discovers the Fibre Channel target devices that are accessible through this connection and provides that information in the Appliances Tab. The Tester is responsible for the getting that information into the Target WWPN input field (direct copy and paste, User Parameter files or Drag & Drop from the FC Ports Info window). The rest of the Fibre Channel Scenario is a set of SCSI Block Protocol commands to read, write or get/set device control information.



The Fibre Channel Ports & Appliances > Appliances Tab shows information associated with the Target WWPNs:

Initiators - the number of Initiators that this Physical Port may appear to be (1 physical card WWPN + the number of NPIV WWPNs configured)

Initiator WWPN - the WWPN of the FC Initiator that is configured to access this Target WWPN

FCID (Fibre Channel ID) - this is the ID of the Initiator assigned by the switch

Targets - the number of Target WWPNs that this Physical Port has access to

Target WWPN - the WWPN of a target FC device that this Appliance Physical Port has access to

FCID (Fibre Channel ID) - this is the ID of the Target assigned by the switch

LUN(s) - the Logical Unit(s) that this Target provides

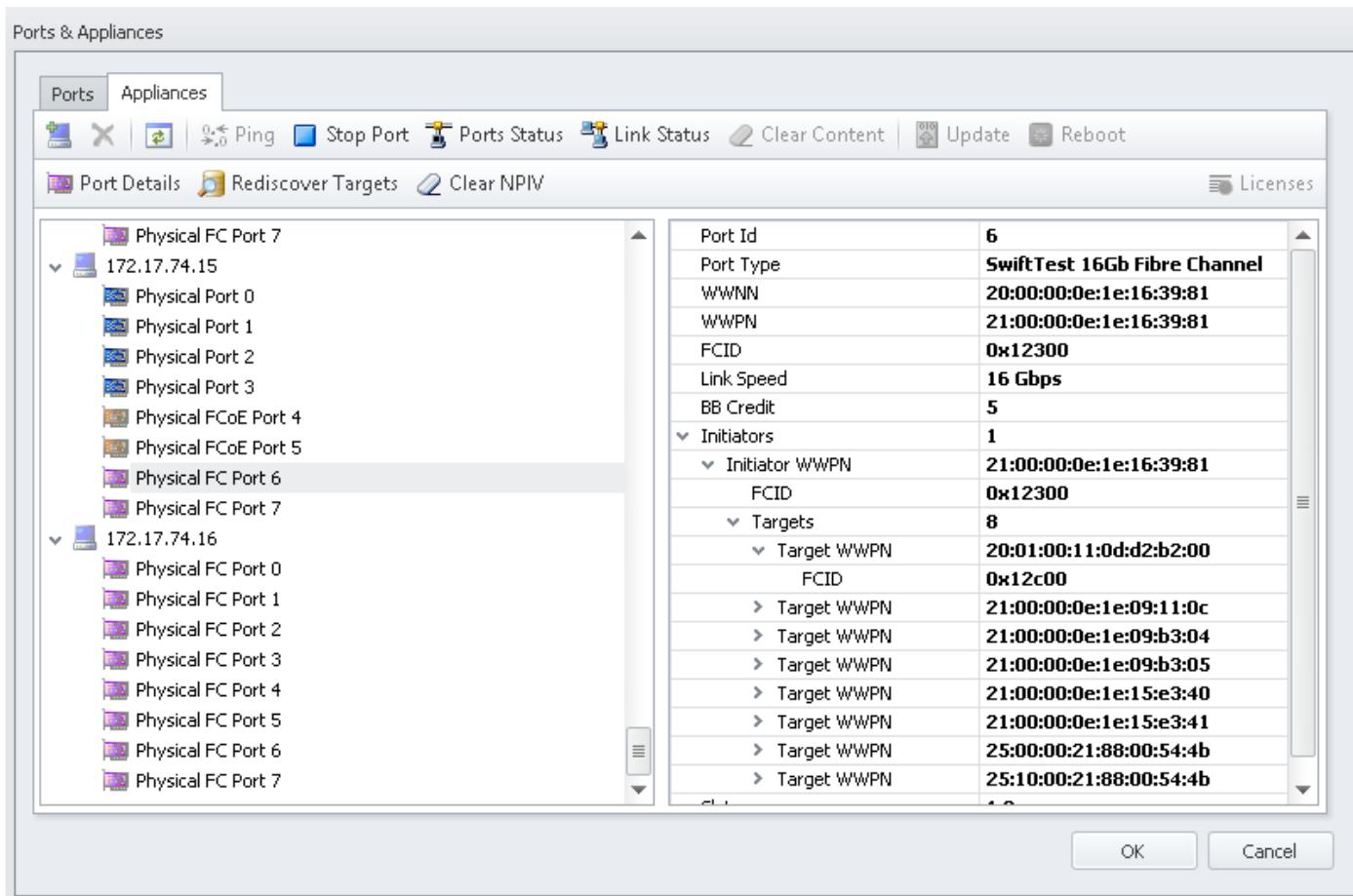
Testers will use this information in the creation of Fibre Channel Projects.

Fibre Channel Projects produce statistics like other Projects visible at run time or after the Project has completed.

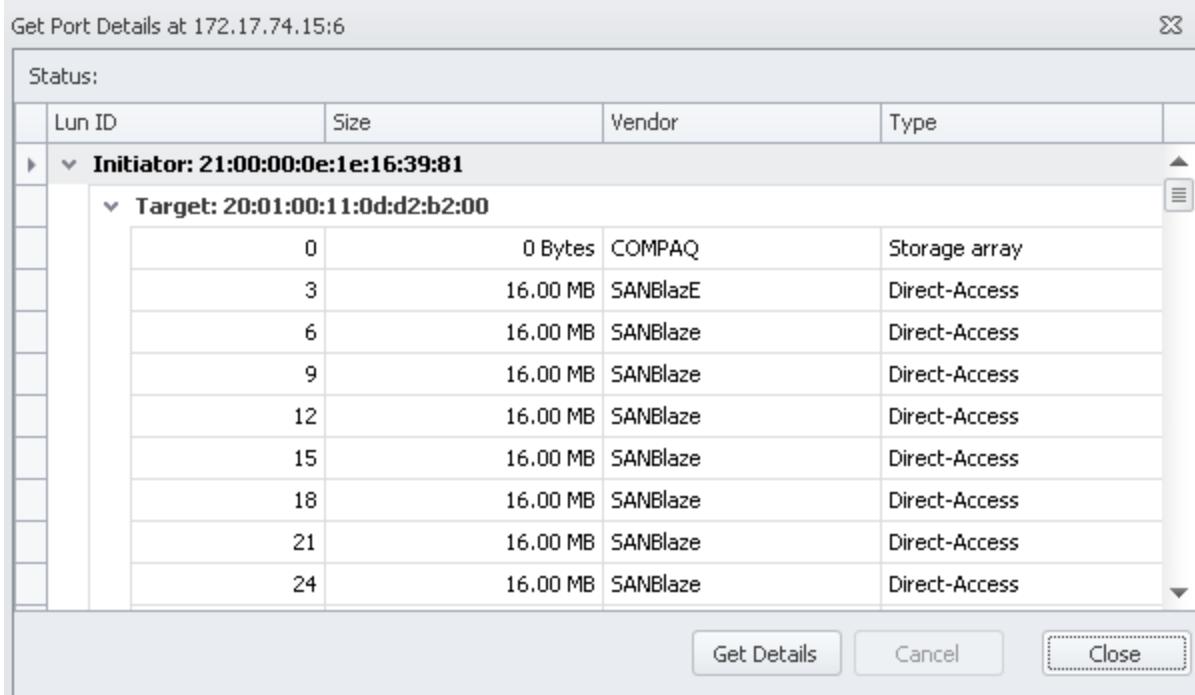
Port Information

FC Port information (initiator WWPN and possibly multiple Target WWPNs are required in FC Open and MPIO Open and Config Actions. The FC Port information can be visualized in several ways.

1. By expanding the information displayed in the Ports and Appliances > Appliances tab for a specific FC Port. Highlight the FC port and the click the > symbols to expose increasing levels of detail (number of targets, target WWPN, LUNs/target and LUN size).



2. By using the Port Details function in the Ports and Appliances > Appliances tab. Highlight an FC port and click the Port Details button.



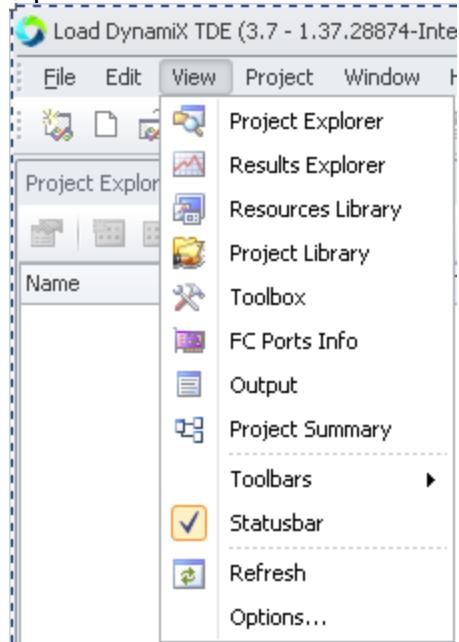
FC Port Information transfer into FC Open and FC MPIO Open and Config Actions

FC Ports Info Drag & Drop

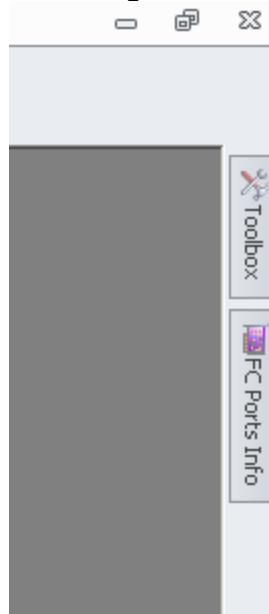
Fibre Channel Initiator and Target WWPNs can be dragged from the FC Ports Info window into Fibre

Channel Action Initiator and Target WWPN input fields for the Open FC Connection and MPIO Open and Config Actions. Simply click on the desired Initiator or Target WWPN field in the FC Port Info table and drag it onto the appropriate input field for the Open FC Connection and MPIO Open and Config Actions. Initiators from the FC Ports Info window cannot be dropped onto Target WWPN input fields and Targets from the FC Ports Info window cannot be dropped onto Initiator WWPN input fields.

Open the FC Ports Info window. Click the View menu and select FC Ports Info



On the right hand side of the TDE the FC Ports Info window will appear



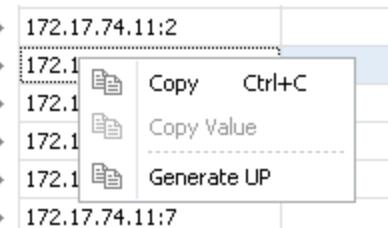
Mouse over the FC Ports Info icon to expose the FC Ports Info window

FC Ports Info		
Initiators and Targets Search... <input checked="" type="checkbox"/>		
▼ 172.17.1.160:0		
> Initiator WWPN	21:00:00:24:ff:01:ad:e6	
> 172.17.1.160:1		
> 172.17.74.10:0		
> 172.17.74.10:1		
> 172.17.74.10:2		
> 172.17.74.10:3		
> 172.17.74.11:0		
▼ 172.17.74.11:1		
▼ Initiator WWPN	21:00:00:0e:1e:14:5c:a0	
▼ Target WWPN	21:00:00:0e:1e:07:02:f8	
Lun(s)	0-4	
> Target WWPN	21:00:00:0e:1e:07:03:00	
> Target WWPN	21:00:00:0e:1e:07:03:18	
> Target WWPN	21:00:00:0e:1e:07:03:f0	
> Target WWPN	21:00:00:0e:1e:07:04:60	
> Target WWPN	21:00:00:0e:1e:09:11:0d	
> Target WWPN	21:00:00:0e:1e:09:11:1b	
> Target WWPN	21:00:00:0e:1e:09:11:20	
> 172.17.74.11:2		
> 172.17.74.11:3		
> 172.17.74.11:4		
> 172.17.74.11:5		
> 172.17.74.11:6		
> 172.17.74.11:7		
> 172.17.74.9:0		
> 172.17.74.9:1		

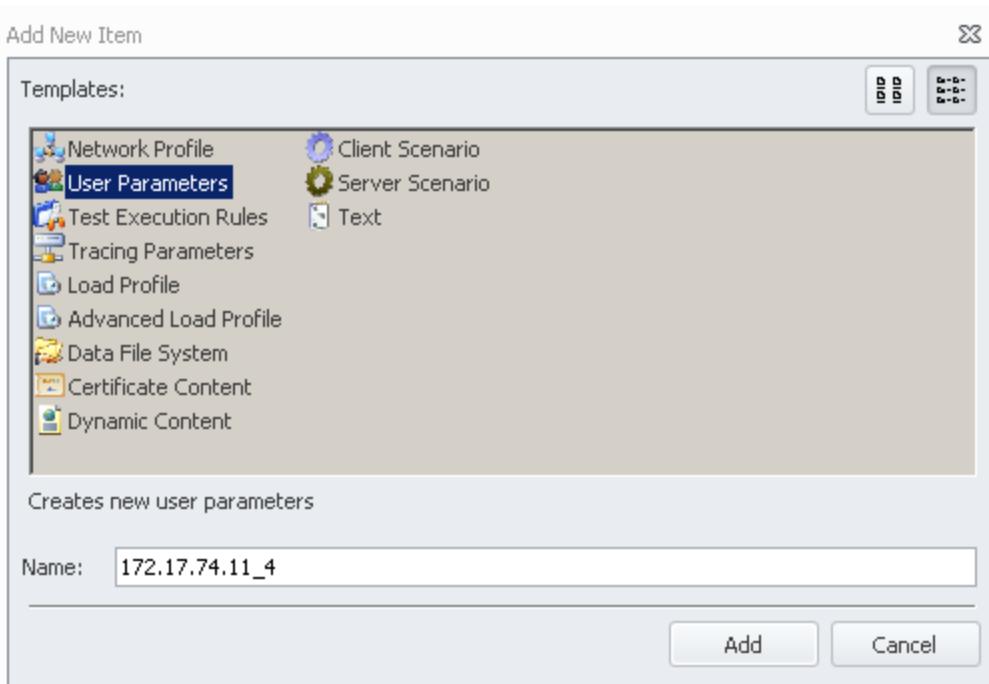
FC Port Info Generate UP File

The information displayed in the FC Ports Info window can be used in Fibre Channel tests as input to FC Open commands and other FC commands (ex: MPIO Open commands). This info can be copied into User Parameter files in two ways.

- Right Click FC Port - Generate UP. Highlight an FC Port in the FC Ports Info window and right-click Generate UP



The result is a User Parameter dialog that allows the Tester to create a UP file with the target and initiator information for that FC Port.



Click Add to get

172.17.74.11_4.user

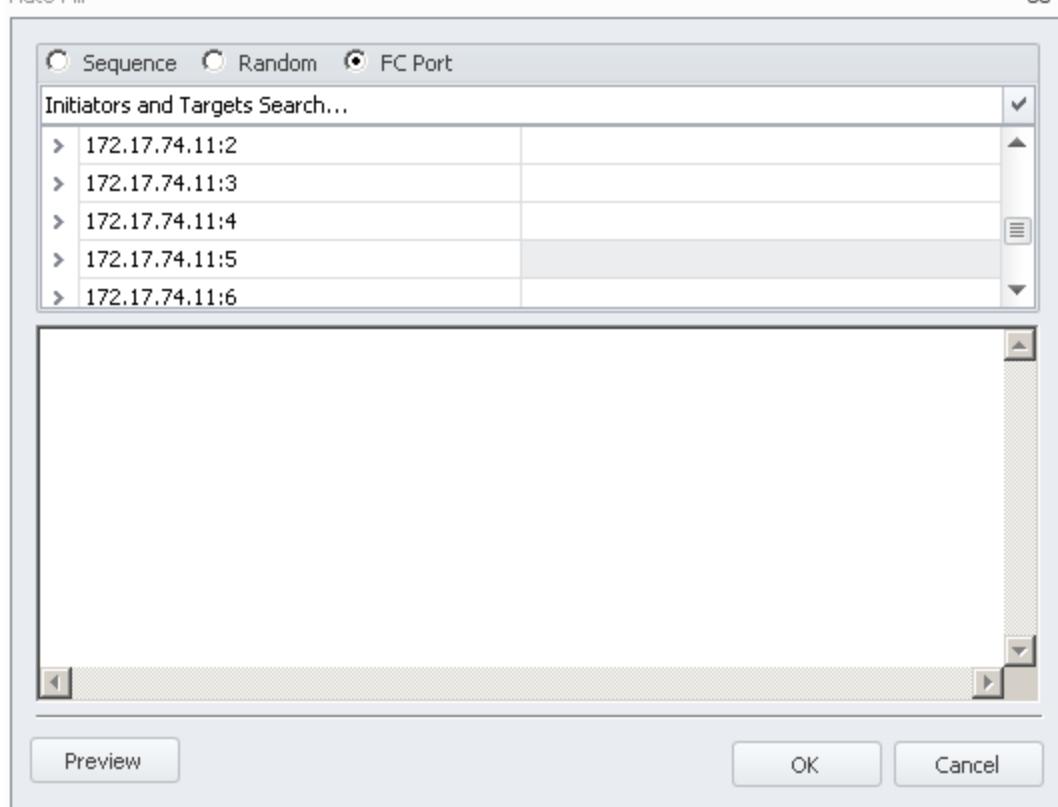
A:INITIAT...	B:TARGET_...	C:LUN_NU...	D:LUN_SIZE	
21:00:00:0...	21:00:00:0...	0	8589934592	
21:00:00:0...	21:00:00:0...	1	10737418240	
21:00:00:0...	21:00:00:0...	2	10737418240	
21:00:00:0...	21:00:00:0...	3	10737418240	
21:00:00:0...	21:00:00:0...	4	104857600	
21:00:00:0...	21:00:00:0...	0	8589934592	
21:00:00:0...	21:00:00:0...	1	10737418240	
21:00:00:0...	21:00:00:0...	2	10737418240	
21:00:00:0...	21:00:00:0...	3	10737418240	

- Autofill an empty UP file

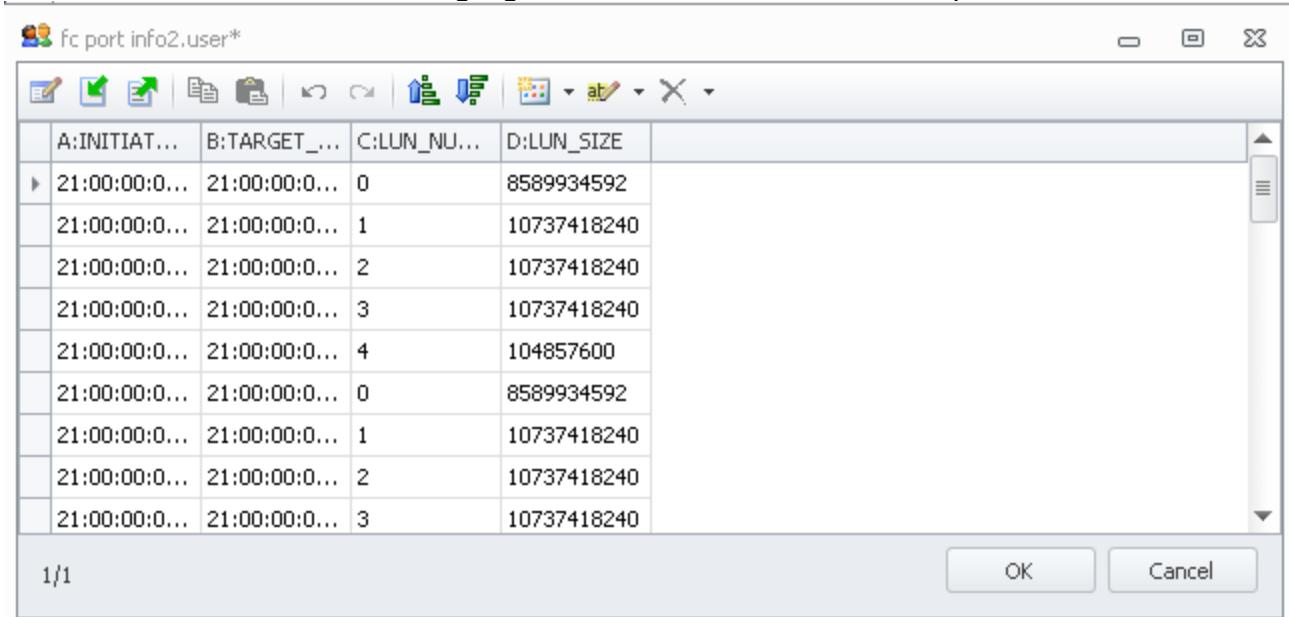
Use the Add New Item button to create a new UP file and then click the Autofill button

Auto Fill

x



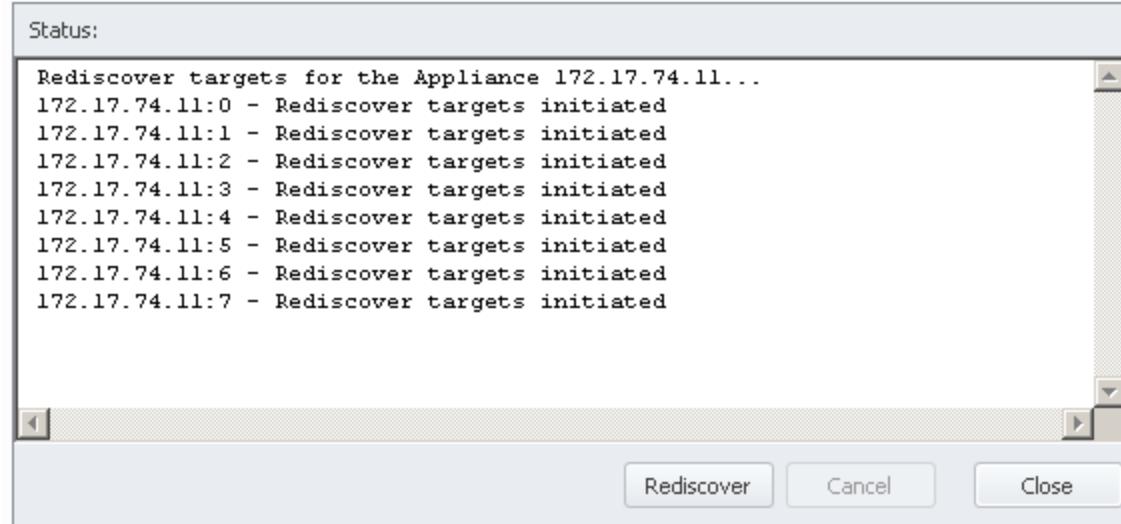
Click the FC Port radio button, highlight the FC Port information is required and click OK



Rediscover Targets

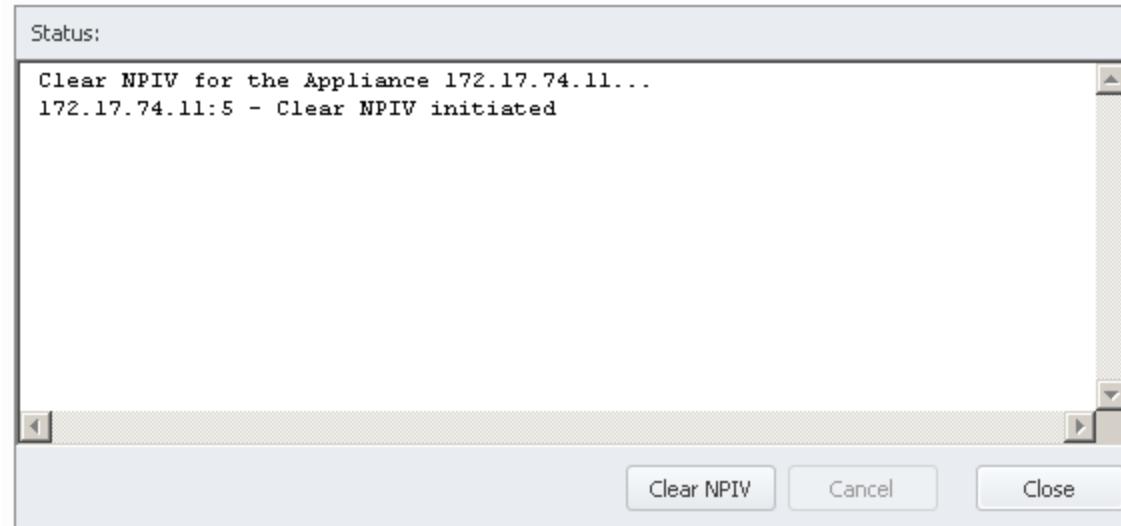
To force the Fibre Channel software in the Load DynamiX Appliance to rediscover targets, highlight the FC Appliance or FC Appliance Port and the click the Rediscover Targets button.

Rediscover Targets

**Clear NPIV**

To remove NPIV configuration information from an FC Port, highlight the FC Port and click the Clear NPIV button

Clear NPIV

**Link Speed**

Link Speed is automatically determined by the Fibre Channel NIC driver and documented in the Fibre Channel port-specific information shown in the Ports & Appliances > Appliances tab when a Fibre Channel port is highlighted. It is not configurable. Load DynamiX 6202/6204/6208 Appliances support 4/8/16Gbps Fibre Channel interfaces. Load DynamiX 6100/6108 support 2/4/8Gbps Fibre Channel interfaces.

Ports & Appliances

The screenshot shows the 'Ports & Appliances' interface with the 'Ports' tab selected. In the left pane, a tree view displays 'Physical FC Port 7' and two IP addresses: '172.17.74.15' and '172.17.74.16'. Under '172.17.74.15', there are eight physical ports labeled 0 through 7. Under '172.17.74.16', there are also eight physical ports labeled 0 through 7. The right pane shows detailed information for the selected port, which is identified as 'Physical FC Port 7' under '172.17.74.15'. The details include:

Port Id	6
Port Type	SwiftTest 16Gb Fibre Channel
WWNN	20:00:00:0e:1e:16:39:81
WWPN	21:00:00:0e:1e:16:39:81
FCID	0x12300
Link Speed	16 Gbps
BB Credit	5
Initiators	1
Initiator WWPN	21:00:00:0e:1e:16:39:81
FCID	0x12300
Targets	8
Target WWPN	20:01:00:11:0d:d2:b2:00
FCID	0x12c00
Target WWPN	21:00:00:0e:1e:09:11:0c
Target WWPN	21:00:00:0e:1e:09:b3:04
Target WWPN	21:00:00:0e:1e:09:b3:05
Target WWPN	21:00:00:0e:1e:15:e3:40
Target WWPN	21:00:00:0e:1e:15:e3:41
Target WWPN	25:00:00:21:88:00:54:4b
Target WWPN	25:10:00:21:88:00:54:4b

At the bottom right are 'OK' and 'Cancel' buttons.

FC Link Speed information is logged in the Client Port Log file during Project execution.

	Line	Type	Date / Time	Text
▶	0	Status	1/28/2014 1:28:41 PM	Load DynamiX Framework [Version 5.35.27060-Internal-Private_MPIO_ALUA]
	1	Status	1/28/2014 1:28:41 PM	(C) Copyright 2008-2014 Load DynamiX, Inc.
ⓘ	2	Info	1/28/2014 1:28:41 PM	Preparing to execute test: APPL-2518-validation
ⓘ	3	Info	1/28/2014 1:28:41 PM	Device [11]: Port type SwiftTest 16Gb Fibre Channel
ⓘ	4	Info	1/28/2014 1:28:41 PM	Device [11]: Linked speed 16 Gbps Full Duplex.
ⓘ	5	Info	1/28/2014 1:28:41 PM	Device [10]: Port type SwiftTest 16Gb Fibre Channel
ⓘ	6	Info	1/28/2014 1:28:41 PM	Device [10]: Linked speed 16 Gbps Full Duplex.
ⓘ	7	Info	1/28/2014 1:28:41 PM	Device [9]: Port type SwiftTest 16Gb Fibre Channel
ⓘ	8	Info	1/28/2014 1:28:41 PM	Device [9]: Linked speed 16 Gbps Full Duplex.
ⓘ	9	Info	1/28/2014 1:28:41 PM	Device [8]: Port type SwiftTest 16Gb Fibre Channel
ⓘ	10	Info	1/28/2014 1:28:41 PM	Device [8]: Linked speed 16 Gbps Full Duplex.
ⓘ	11	Info	1/28/2014 1:28:41 PM	Device [3]: Port type SwiftTest 16Gb Fibre Channel
ⓘ	12	Info	1/28/2014 1:28:41 PM	Device [3]: Linked speed -1 Gbps Full Duplex.
ⓘ	13	Info	1/28/2014 1:28:42 PM	Device [2]: Port type SwiftTest 16Gb Fibre Channel
ⓘ	14	Info	1/28/2014 1:28:42 PM	Device [2]: Linked speed -1 Gbps Full Duplex.
ⓘ	15	Info	1/28/2014 1:28:42 PM	Device [1]: Port type SwiftTest 16Gb Fibre Channel
ⓘ	16	Info	1/28/2014 1:28:42 PM	Device [1]: Linked speed -1 Gbps Full Duplex.
ⓘ	17	Info	1/28/2014 1:28:42 PM	Device [0]: Port type SwiftTest 16Gb Fibre Channel
ⓘ	18	Info	1/28/2014 1:28:42 PM	Device [0]: Linked speed -1 Gbps Full Duplex.
⚡	19	Debug	1/28/2014 1:28:42 PM	Device [11] subnet [0]: Initialized 255 IPv4 Addresses in range 192.169.1.2 - 192.169.1.0
ⓘ	20	Info	1/28/2014 1:28:42 PM	Expected execution duration is up to 00:01:00
ⓘ	21	Info	1/28/2014 1:28:42 PM	Client-side high-performance stack is entering active state...
ⓘ	22	Info	1/28/2014 1:29:34 PM	All scenarios ramped-down. Exiting...
ⓘ	23	Info	1/28/2014 1:29:34 PM	Client-side high-performance stack is exiting active state...
ⓘ	24	Info	1/28/2014 1:29:34 PM	Client-side high-performance stack execution time 00:00:52.

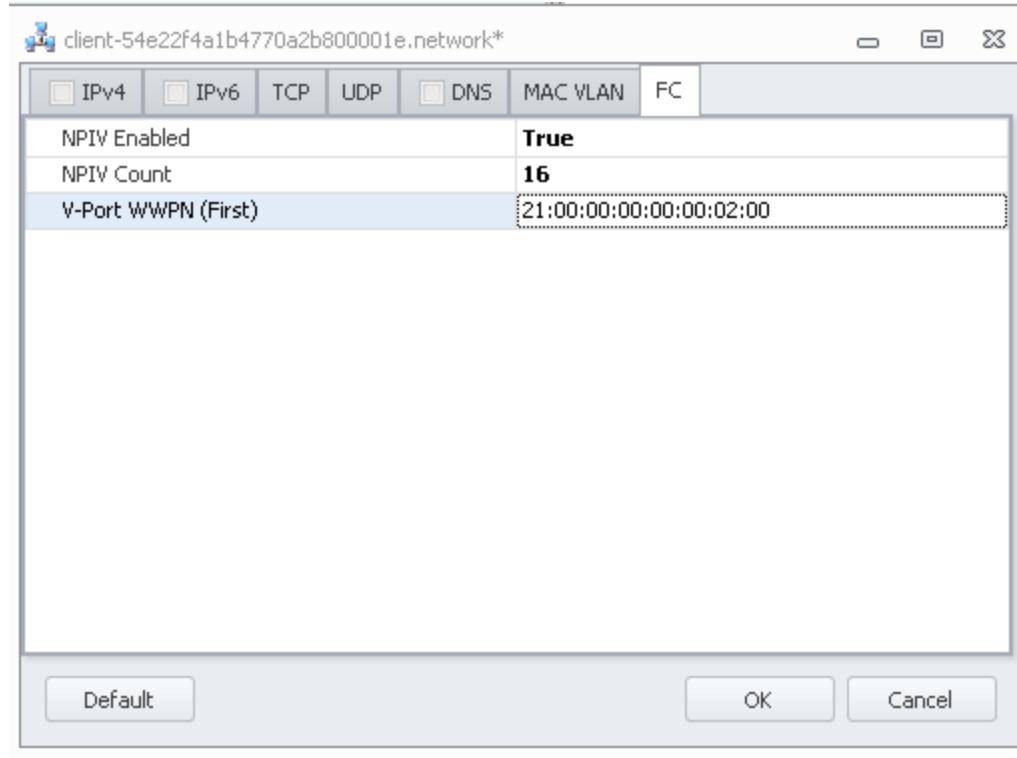
NPIV (N_Port ID Virtualization)

NPIV is a Fibre Channel facility allowing multiple FC Initiator WWPNs to share a single FC Physical Port. This allows multiple Fibre Channel initiators to occupy a single FC Physical Port. Enabling

NPIV allows the Tester to create a set of virtual WWPNs for the Load DynamiX FC Initiator to use. These virtual WWPNs must be provisioned on the Target devices so that they will accept Sessions from these WWPNs. Only the WWPNs that are provisioned on Fibre Channel Targets will be used by the Load DynamiX Fibre Channel driver (i.e. even though the Load DynamiX Appliance is configured for multiple virtual WWPNs, the virtual WWPN will only be used if the Target is provisioned to allow it). If NPIV is enabled (=True) on a Fibre Channel Physical Port, only the virtual WWPNs will be used.

To enable NPIV on the Load DynamiX Appliance, open the Network Profile resource and click on the FC tab. Change the NPIV Enabled input to True and define the number of NPIV WWPNs to use and the base NPIV WWPN.

NPIV Configuration - the processing time to configure NPIV WWPNs depends on the number of virtual interfaces defined in the Network Profile and the number of targets or other devices accessible by the Fibre Channel interface. In a fully meshed Fibre Channel SAN network, it may take the Appliance's Fibre Channel driver from several seconds up to several minutes to completely configure itself. So, it is prudent to configure some start up time for a Project by using the Port Delay feature (see the [Load DynamiX Test Development](#) chapter for a discussion of the Port Delay feature).



Appliances Tab Physical Port Information - When NPIV initiators have been configured as above, the Physical Port information shown in the Appliances tab will show both the NPIV Initiator WWPNs and the Target WWPNs that these NPIV WWPNs are configured to access. The NPIV configuration shown above would generate 16 Initiators with WWPN values starting at 21:00:00:00:00:02:00.

NPIV Notes

- Certain SAN switches will shutdown a port if more initiator WWPNs (NPIV) are used than are configured on the switch.
- NPIV and MPIO cannot be combined in the same Fibre Channel Project. Projects with both NPIV and MPIO enabled will fail to compile.
- NPIV connections do not support Reconnect even if the Reconnect Maximum Attempts is set > 0.

- The maximum number of NPIV initiators per FC port is 254.

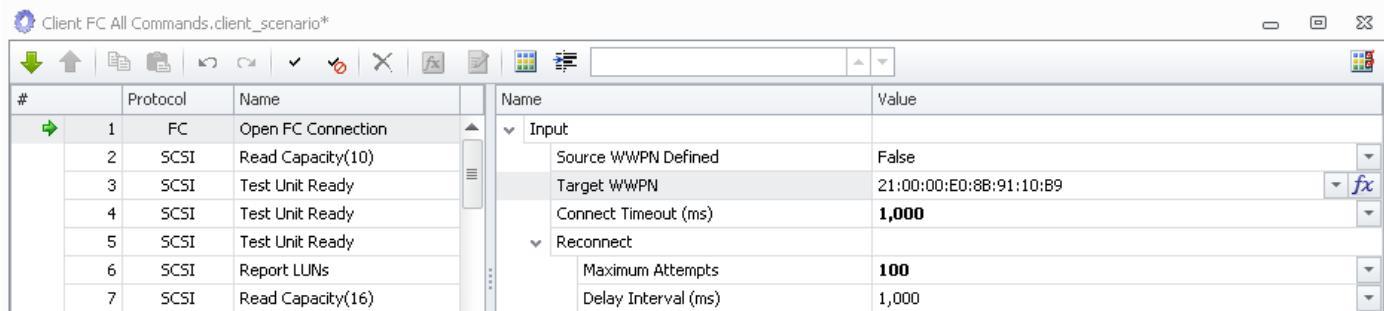
Fibre Channel Connect Timeout (ms)

An input to the Open FC Connection Action is the Connect Timeout (ms) value. This value (in milliseconds), controls the length of time that

- An open/functioning FC connection will wait after the connection has stopped before it enters the Reconnect cycle in which it will attempt to re-establish the open connection or
- A Open FC Connection Action will wait if the target LUN is not accessible before it fails the Scenario if 0 Reconnect attempts have been specified.

Fibre Channel Reconnect

Load DynamiX Fibre Channel support provides the ability for an FC connection to attempt reconnection if an Open Fibre Channel connection is closed for any reason. The Reconnect process is controlled by inputs to the **Open FC Connection** Action. Set Reconnect Maximum Attempts to a number > 0 to set the maximum number of attempts per Reconnect and to enable the Reconnect processing and then set Reconnect Delay Interval (how long to delay after the disconnect before attempting a reconnect) and Connect Timeout which sets the time threshold for a Reconnect attempt. The Scenario below will attempt 100 Reconnects after a 1 second (1000 ms) timeout and will delay 1 second (1000ms) between reconnect attempts. Reconnects provide reconnect abilities before and after a connection has been successfully opened.

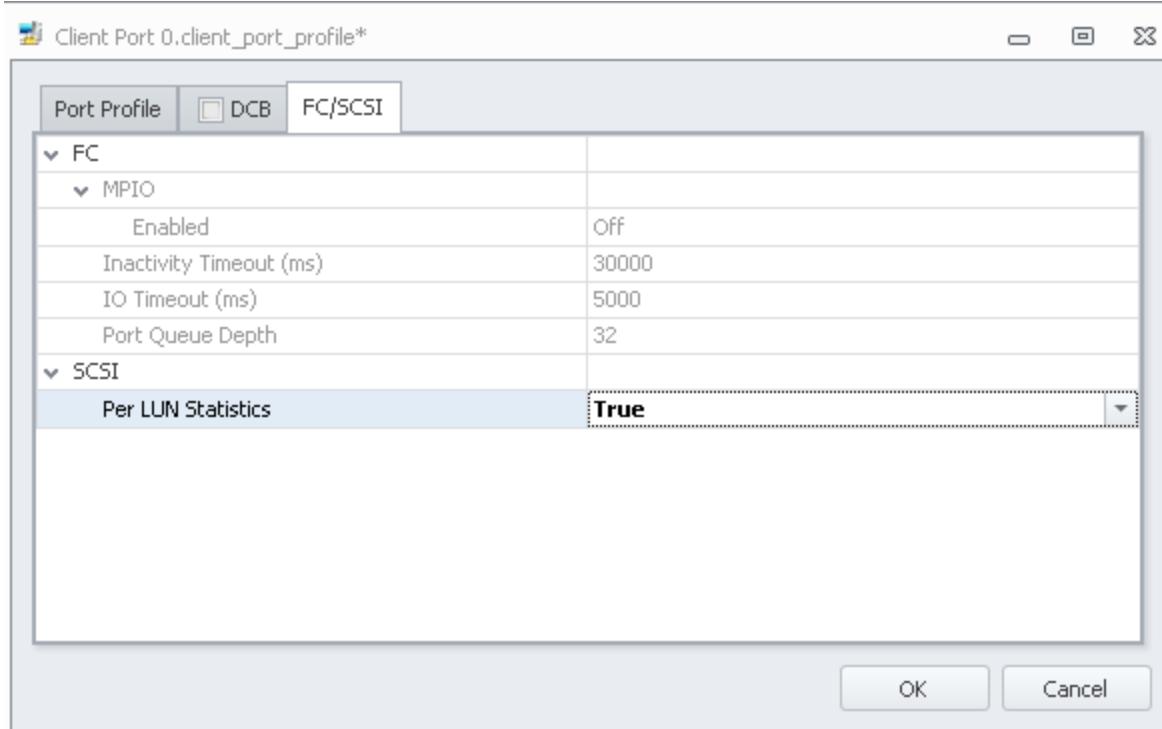


Fibre Channel Inactivity Timeout (ms)

In the Fibre Channel tab (FC) of the Client Logical Port resource is a setting for Inactivity Timer (default 30000 milliseconds). This timer controls the length of time the Fibre Channel lower level mechanisms will wait for a Target device to respond to a request before it will attempt the request again. This timer may need to be set to a much lower number in Scenarios that use NPIV Initiators or Targets because operations that contain NPIV WWPNs typically take much longer to initially provision the low level Fibre Channel mechanisms.

Fibre Channel IO Timeout (ms)

The default length of time that the Load DynamiX FC driver will wait for an IO operation to complete is 5 seconds. If the Tester wants to alter than value (up or down), the IO Timeout field is used. A value of less than 5000 means that the FC driver will wait for less than 5 seconds for all IO operations to complete. A value of greater than 5000 means that the FC driver will wait for more than 5 seconds for all IO operations to complete. The input value is in milliseconds.



Fibre Channel Tracing (PCAP)

Fibre Channel Tracing support is being introduced in v3.4. Fibre Channel Tracing differs from IP Tracing in that it only captures the SCSI traffic in a Fibre Channel connection. Fibre Channel trace is presented using `LINKTYPE_FC_2` format (Fibre Channel FC-2 frames, beginning with a `Frame_Header`). Fibre Channel traces can be inspected with `tcpdump` or Wireshark v 1.6.8 or later.

An example Fibre Channel PCAP file is displayed in Wireshark:

Client Port 0(172.17.1.162 port 0).pcap [Wireshark 1.6.8 (SVN Rev 42761 from /trunk-1.6)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	01.13.00	01.08.00	FCP	56	SCSI: Read Capacity(10) LUN: 0x00
2	1.806937	01.08.00	01.13.00	FCP	72	SCSI: Response LUN: 0x00 (Read Capacity(10)) (Check Condition) LUN:0x00
3	1.806972	01.13.00	01.08.00	FCP	56	SCSI: Read Capacity(10) LUN: 0x00
4	1.807274	01.08.00	01.13.00	FCP	32	SCSI: Data In LUN: 0x00 (Read Capacity(10) Response Data)
5	1.807276	01.08.00	01.13.00	FCP	36	SCSI: Response LUN: 0x00 (Read Capacity(10)) (Good)
6	1.807284	01.13.00	01.08.00	FCP	56	SCSI: Test Unit Ready LUN: 0x00
7	1.807454	01.08.00	01.13.00	FCP	36	SCSI: Response LUN: 0x00 (Test Unit Ready) (Good)
8	1.807464	01.13.00	01.08.00	FCP	56	SCSI: Report LUNS LUN: 0x00
9	1.807707	01.08.00	01.13.00	FCP	40	SCSI: Data In LUN: 0x00 (Report LUNS Response Data)
10	1.807711	01.08.00	01.13.00	FCP	36	SCSI: Response LUN: 0x00 (Report LUNS) (Good)
11	1.807720	01.13.00	01.08.00	FCP	56	SCSI: Report LUNS LUN: 0x00
12	1.807881	01.08.00	01.13.00	FCP	40	SCSI: Data In LUN: 0x00 (Report LUNS Response Data)
13	1.807885	01.08.00	01.13.00	FCP	36	SCSI: Response LUN: 0x00 (Report LUNS) (Good)

Frame 1: 56 bytes on wire (448 bits), 56 bytes captured (448 bits)

Fibre Channel

- [Exchange Last In: 2313]
 - R_CTL: 0x6(Device_Data/Unsolicited Command)
 - Dest Addr: 01.08.00
 - CS_CTL: 0x00
 - Src Addr: 01.13.00
 - Type: FCP (0x08)
 - F_CTL: 0x290000 Exchange originator, Seq Initiator, Exch First, Seq Last, CS_CTL, Transfer Seq Initiative
 - SEQ_ID: 0x00
 - DF_CTL: 0x00
 - SEQ_CNT: 0
 - OX_ID: 0x0000
 - RX_ID: 0xffff
 - Parameter: 0x00000000
- [FCP: FCP_CMND]
 - [Response In: 2313]
 - LUN: 0x00
 - Command Ref Num: 0
 -000 = Task Attribute: Simple (0x00)
 - Task Management Flags: 0x00 (No values set)
 - 0000 00.. = Additional CDB Length: 0
 -1. = RDDATA: True
 -0 = WRDATA: False
 - FCP_DL: 8
- [SCSI CDB Read Capacity(10)]
 - [LUN: 0x0000]
 - [Command Set:Direct Access Device (0x00) (using default commandset)]
 - [Response In: 2313]

The Source and Destination device information highlighted above are the FCID values assigned to Fibre Channel devices by the switch fabric that they are connected to. The FCID values are unique within that switched fabric and they are seen as the Source and Destination addresses in PCAP displays.. These values can also be seen by looking at the Appliances Tab Port information for a Load DynamiX 6202/6204/6208/6202E Fibre Channel Appliance (see below).

Ports & Appliances

The screenshot shows a software interface for managing network ports. On the left, a tree view lists ports under two IP addresses: 172.17.74.10 and 172.17.74.11. Under each IP address, there are multiple physical FC ports (Physical FC Port 0-7) and physical ports (Physical Port 4-7). On the right, a detailed table provides configuration information for a selected port:

Port Id	2
Port Type	SwiftTest 16Gb Fibre Channel
WWNN	20:00:00:0e:1e:14:b2:81
WWPN	21:00:00:0e:1e:14:b2:81
FCID	0x10600
Link Speed	16 Gbps
BB Credit	5
Initiators	
Initiator WWPN	21:00:00:0e:1e:14:b2:81
FCID	0x10600
Targets	
Targets	8
Target WWPN	21:00:00:0e:1e:07:02:f8
FCID	0x10300
> LUN(s)	
> Target WWPN	21:00:00:0e:1e:07:03:00
> Target WWPN	21:00:00:0e:1e:07:03:18
> Target WWPN	21:00:00:0e:1e:07:03:f0
> Target WWPN	21:00:00:0e:1e:07:04:60
> Target WWPN	21:00:00:0e:1e:09:11:0d

At the bottom right are 'OK' and 'Cancel' buttons.

MPIO/ALUA

MPIO/ALUA Toolbox

The toolbox menu is organized into categories:

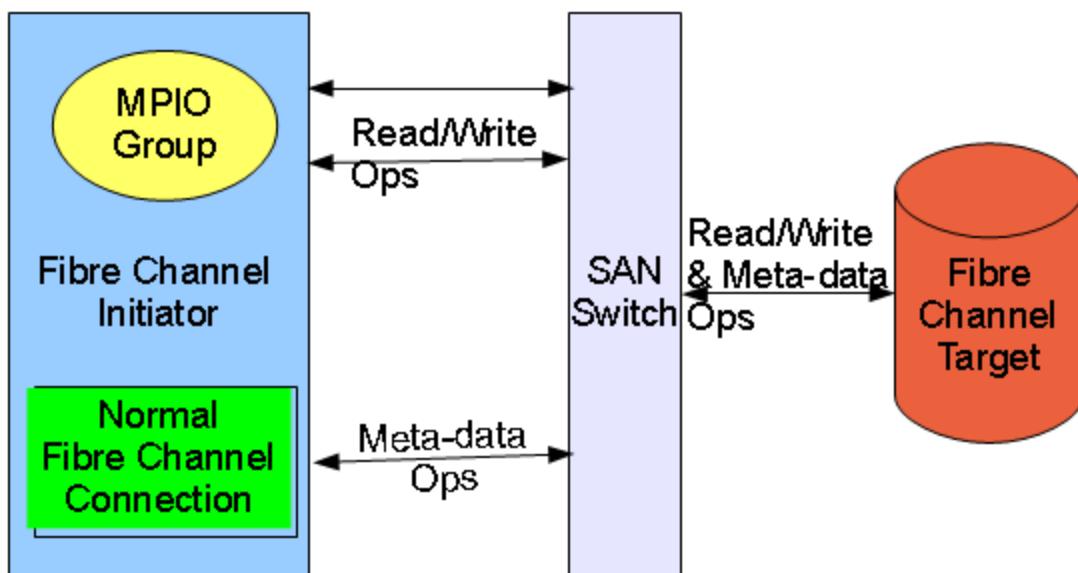
- SAN**
 - iSCSI
 - FC
 - FCoE
 - SCSI
 - SBC
 - SSC
 - SMC
- MPIO/ALUA**
 - MPIO Open and Config
 - MPIO Config
 - ALUA Discovery
 - Report Target Port Groups
 - Set Target Port Groups

MPIO

MPIO (MultiPath Input/Output) generally provides multiple data and control paths (MPIO Groups) that are all capable of successfully servicing the same Input or Output (Read or Write) request. MPIO is only applicable to SCSI Read or Write operations. How Groups are used is controlled by the Load Balancing and Fail Over/Fail Back Policies that the Load DynamiX Appliance Firmware supports. Load Balancing Policies impact the performance and scalability of Fibre Channel/iSCSI connections. Fail Over/Fail Back Policies impact the reliability and redundancy of Fibre

Channel/iSCSI connections.

Meta-data types of operations (i.e. non read or write commands such as **Mode Select (10)**, **Inquiry**, etc.) are not supported on MPIO Groups. Any Scenario that needs to contain both SCSI meta-data operations and MPIO reads and writes must have both an MPIO Group configured as well as a normal Fibre Channel connection configured. At a minimum, this Scenario would contain 3 Open Fibre Channel Connection Actions. Two for the MPIO Group and one for the non-Read/Write operations. The Meta-data operations are constrained to the normal Fibre Channel connection and the reads/writes operate on the MPIO Group. The graphic below illustrates how the combination of MPIO and meta-data operations would be combined in a Scenario.

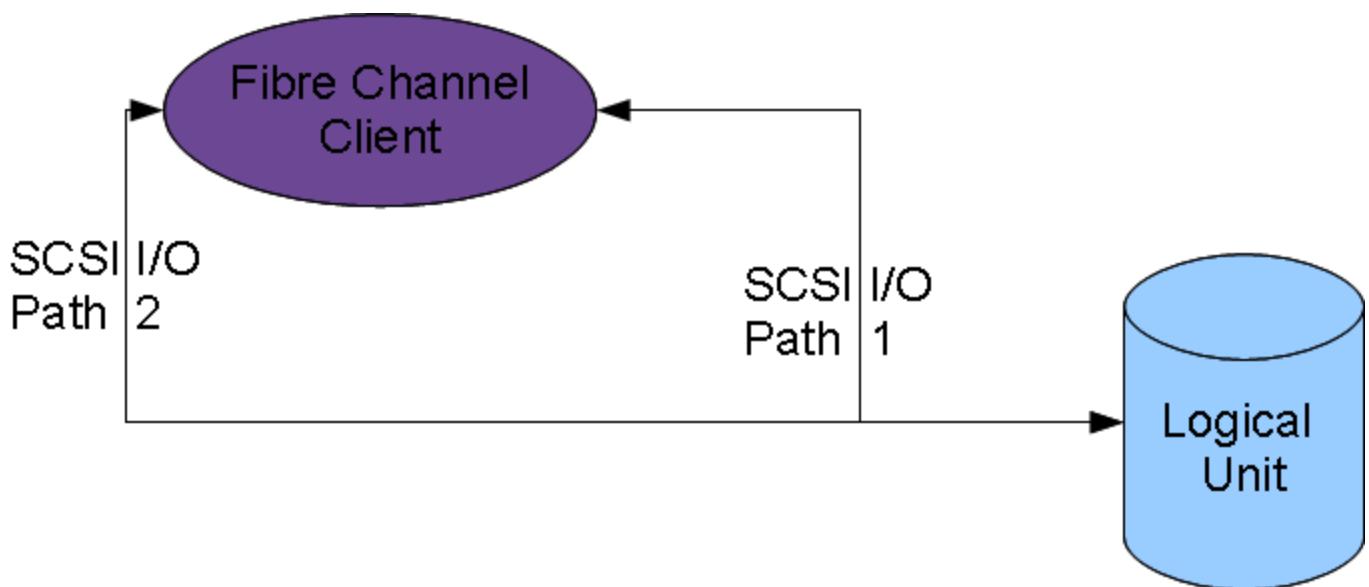


The following Fibre Channel Scenario contains both MPIO operations and non-MPIO meta-data operations. The MPIO and non-MPIO operations are mixed throughout the Scenario and are controlled by which connection of LUN Handle the Action uses.

#	Protocol	Name
2	FC	Open FC Connection
3	SCSI	Read Capacity(10)
4	FC	Open FC Connection
5	SCSI	Read Capacity(10)
6	SCSI	Test Unit Ready
7	#	<i>Configure MPIO Ports and Targets</i>
8	SCSI	MPIO Config
9	SCSI	Verify(10)
10	#	<i>MPIO Reads and Writes</i>
11	SWT	Begin Loop
12	SCSI	LUN Write
13	SCSI	LUN Read
14	SWT	End Loop

Name	Value
Input	
Policy	Round Robin
Enable ALUA Reconfig	False
Primary Path	
Primary Path LUN	3: LUNHandle
Primary Path ID	1
Secondary Path	
Secondary Path LUN	5: LUNHandle
Secondary Path ID	2
Additional Paths	0
Output	
Output Handle	8: LUNHandle

For FC MPIO-enabled Scenarios, the Load DynamiX TDE and Appliance allows the Tester to use multiple Physical Test Ports from the same Appliance in a single Scenario to emulate MPIO behavior in the real world (see MPIO Configuration below for options: **Pair** or **All**). In the image below, the Fibre Channel Initiator has two data/control paths to reach the same Logical Unit on a SCSI disc. Path 1 is the "Primary" Path and Path 2 is the "Secondary" Path. Fibre Channel MPIO is currently the only Protocol that the Load DynamiX Appliance allows the use of multiple Physical Test Ports in the same Scenario. For iSCSI MPIO Scenarios, only a single Physical Test Port may be used.



MPIO Policies

Load DynamiX software supports four MPIO policies that can be used to control I/O paths (MPIO Groups) that are defined by a Project:

- Round Robin (RR) Load Balancing - I/O operations are shared equally by the paths configured. This policy assumes near equal performance by paths configured.
- Least Queue Depth (LQD) Load Balancing - I/O operations are always added to the Queue for the Path with the least (smallest number of) pending operations. Read and Write NOR settings have significant impact on the behavior of LQD traffic. NOR == 1, LQD behavior is 100% of traffic on the Primary Path. NOR > 1 then LQD traffic will get distributed across all paths like the Round Robin policy.
- Fail Over Only (FO) Redundancy - the Primary Path handles all I/O operations unless there is a failure in which case, I/O traffic is moved to the Secondary Path. When the Primary Path is back on line, I/O operations are moved back to the Primary Path.
- Weighted Paths - Weighted Paths is a form of a Fail Over policy in which the paths used are given a "Weight" which defines their relative priority for carrying traffic. The Path with the highest priority (lowest Weighted Path value) is the Path that will carry traffic if it can. In the event that the Path with the highest priority cannot carry traffic, the traffic will be moved to the Path with the next highest Priority that can carry traffic. Traffic always reverts to the Path with the highest priority that is able to carry traffic..

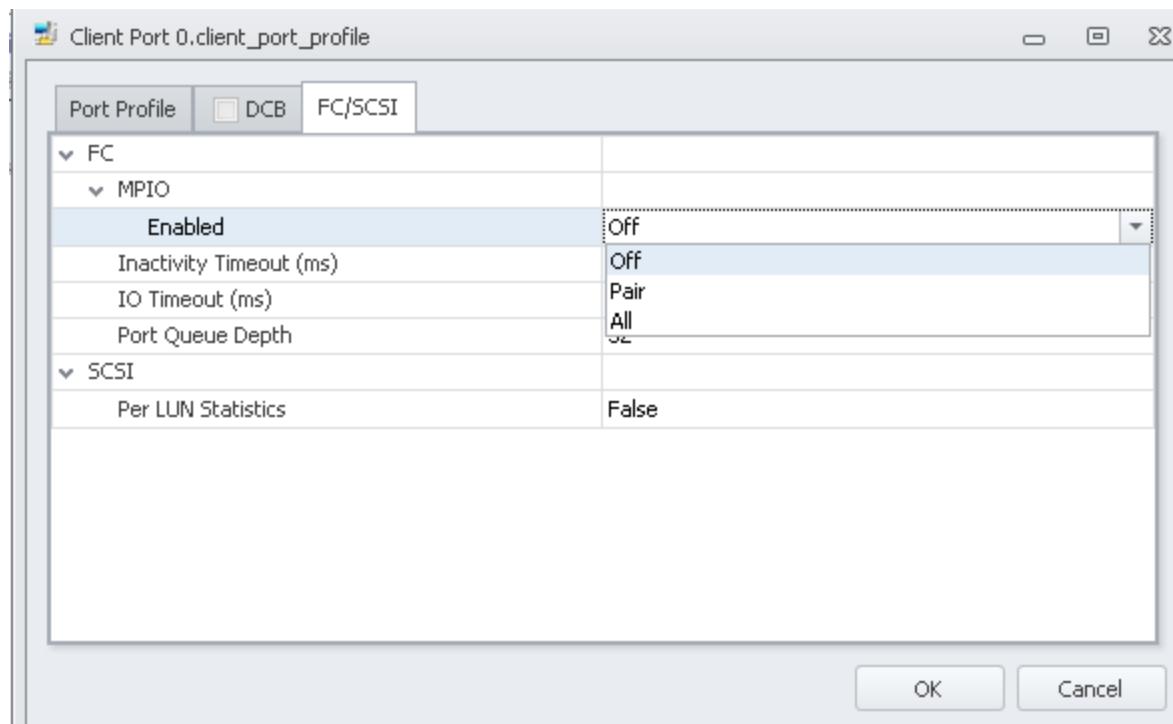
In the Scenario above the MPIO Config Action in line 7 defines the Primary and Secondary paths for Round Robin load balancing Policy based on the Open FC Connection and Read Capacity(10) Actions above it. MPIO Config takes the output of Read Capacity 10 Actions (LUN Handles) to create the Primary and Secondary Paths.

MPIO Configuration

MPIO enabled Scenarios require two MPIO-specific configuration changes. If MPIO Enabled is not set to **Pair** or **All** then the connections required by MPIO Actions (**MPIO Config** and **MPIO Open and Config**) will not succeed. These failures will manifest themselves as Fibre Channel Sessions that fail.

1. Fibre Channel Client Port - Enable MPIO. To enable MPIO the Tester must choose how many Ports are to involved in the MPIO Project: **Pair** (two) or **All**. **Off** is the default setting for MPIO Enabled. **Pair** indicates to the Fibre Channel Firmware that only two ports are to be used which leaves the rest of the ports on the Appliance (in the case of the 6204 or 6208 Appliances) are

available for use in other Projects. **All** indicates that every port on the Appliance is reserved for the MPIO Project during execution. iSCSI MPIO does not require that MPIO be specifically Enabled.



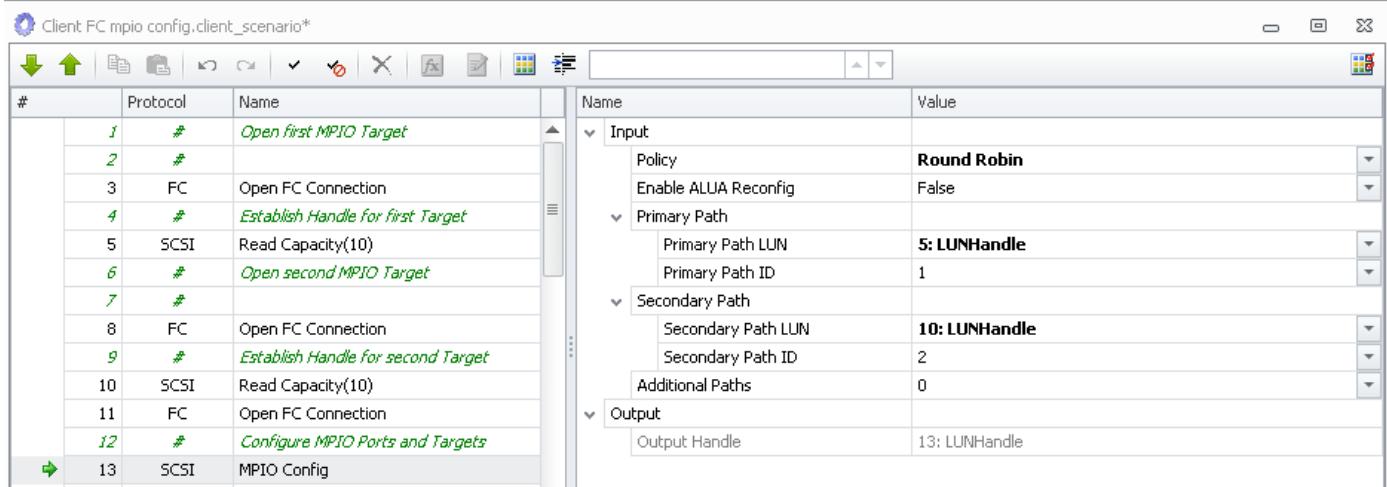
2. Configure MPIO in the Scenario

Fibre Channel MPIO requires at least two Physical Ports (two unique Initiator WWPNs) and one or more Target WWPNs. iSCSI MPIO only supports a single Physical iSCSI Port. Since MPIO only applies to Read and Write Actions and the handle that SCSI Read and Write Actions operate on is a LUNHandle, the final output of the MPIO configuration process must be a LUNHandle. MPIO configuration can be accomplished using two methods:

One MPIO Open and Config Action

#	Protocol	Name
1	#	Open Primary and Secondary Targets
2	#	Configure LUNs to be accessed
3	#	Specify MPIO Policy
4	FC	MPIO Open and Config
5	#	MPIO Reads and Writes
6	SWT	Begin Loop
7	SCSI	LUN Write
8	SCSI	LUN Read
9	SWT	End Loop
10	#	
11	#	
12	#	
13	#	
14	#	
15	#	

Or multiple **Open FC Connection**, **Read Capacity(10)** Actions followed by an **MPIO Config** Action.

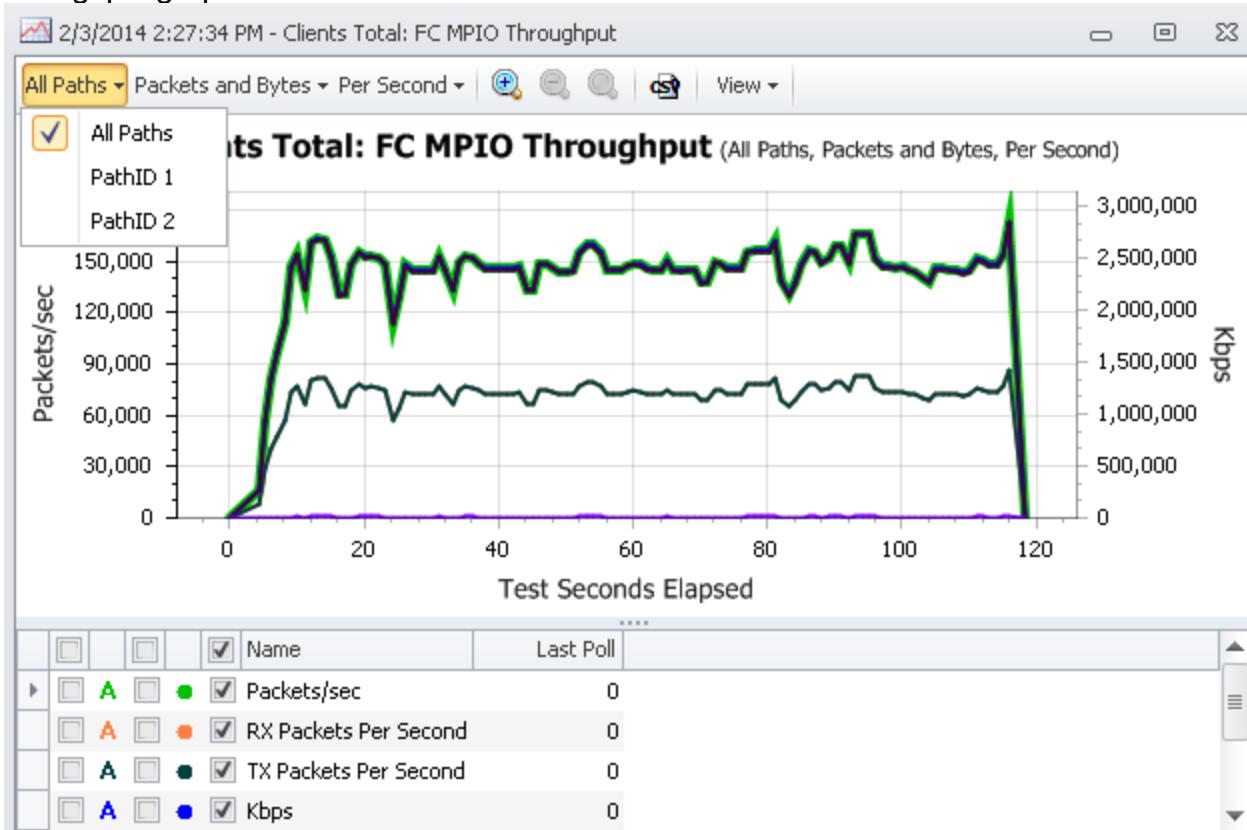


Both methods accomplish the same end result, a LUNHandle for Reads and Writes that operates on two Physical Ports according to the Policy (Round Robin, Least Queue Depth or Fail Over/Fail Back) in the **MPIO Open and Config** or **MPIO Config** Actions. It is possible to have more than 2 MPIO paths in a Scenario. A second **MPIO Config** Action or **MPIO Open and Config** Action could be added to these Scenarios to create 4 paths to the same LUN.

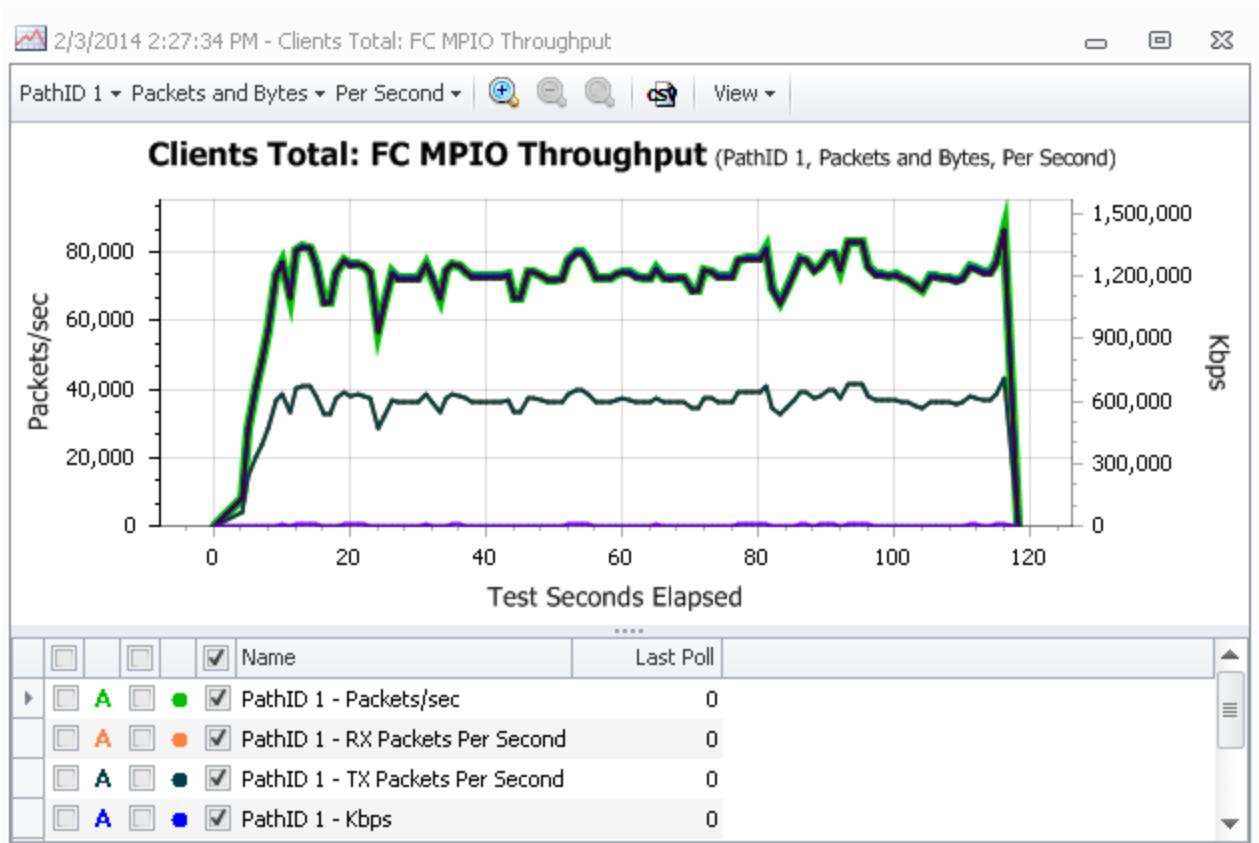
MPIO Statistics

Load Balancing Policies

Scenarios that are MPIO enabled also generate MPIO statistics on a Per Path basis. So in the context of the two scenarios above (the same Read and Write operations in both), the FC MPIO throughput graph could look like

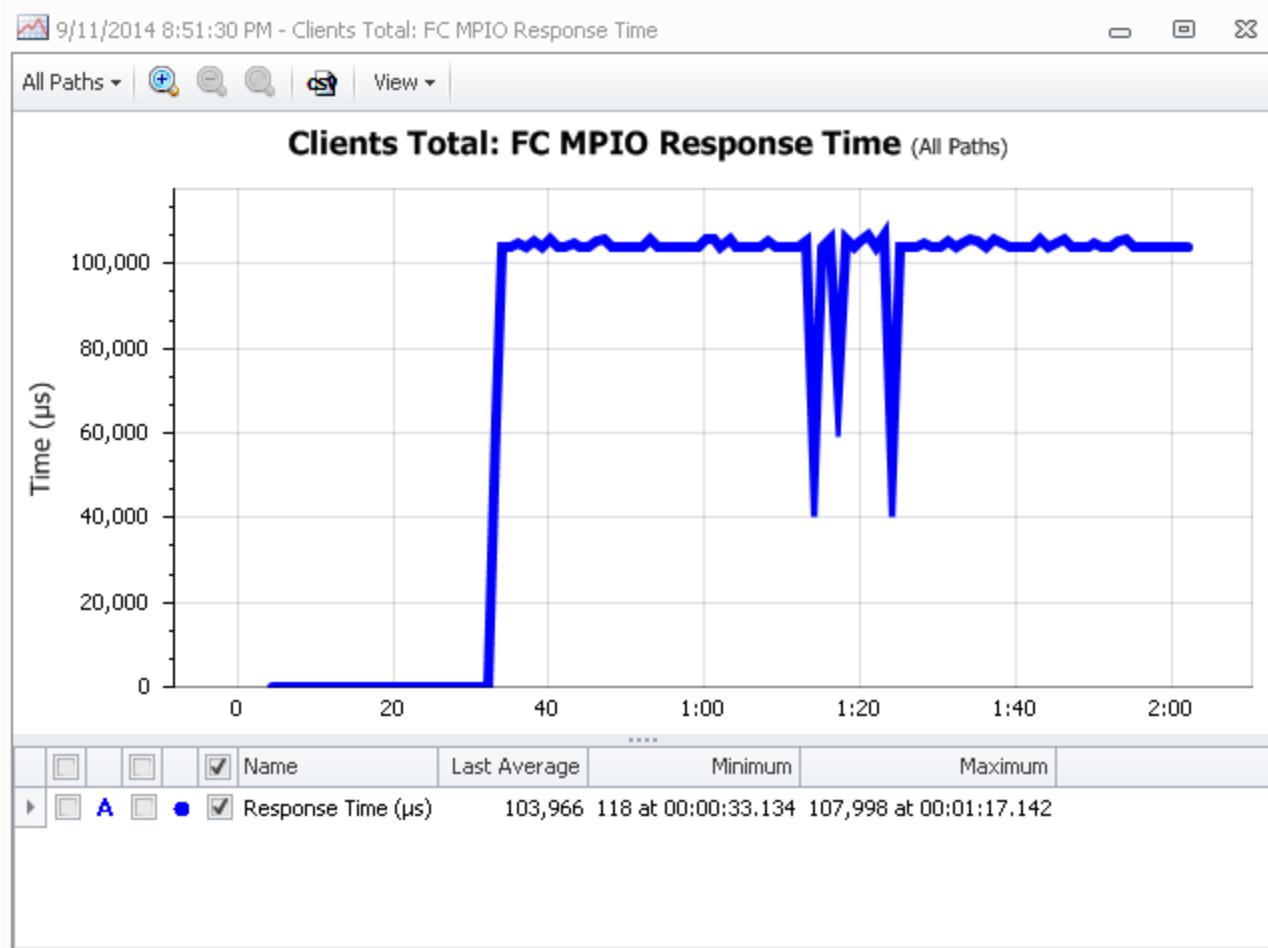


Notice that the menu choices are All Paths, Path1 and Path 2 so it is possible to visualize the throughput of this Project execution on a per-Path basis. The Path 1 view of this graph is



which indicates that Path 1 handled approximately 50% of the data sent or received, which is what would be expected using a Round Robin Policy on two equally efficient Paths.

MPIO statistics include MPIO Response Time (FC example below).



There are also new MPIO statistics in the Client Log file

MPIO Packets:	Attempted	Succeeded	Failed	Aborted
<hr/>				
Overall Total:	64000	64000	0	0
-- Received:	32000	32000	0	0
-- Transmitted:	32000	32000	0	0
<hr/>				
MPIO_path_1	32000	32000	0	0
-- Received:	16000	16000	0	0
-- Transmitted:	16000	16000	0	0
MPIO_path_2	32000	32000	0	0
-- Received:	16000	16000	0	0
-- Transmitted:	16000	16000	0	0
<hr/>				
<hr/>				
MPIO Throughput:	Attempted	Succeeded	Failed	Aborted
<hr/>				
Overall Bytes Total:	16777728000	16777728000	0	0
-- Bytes Received:	8388608000	8388608000	0	0
-- Bytes Transmitted:	8389120000	8389120000	0	0
<hr/>				
MPIO_path_1	8388864000	8388864000	0	0
-- Bytes Received:	4840751104	4840751104	0	0
-- Bytes Transmitted:	3548112896	3548112896	0	0
MPIO_path_2	8388864000	8388864000	0	0
-- Bytes Received:	3547856896	3547856896	0	0
-- Bytes Transmitted:	4841007104	4841007104	0	0
<hr/>				
<hr/>				
MPIO Response Time:	Avg Micros	Min Micros	Max Micros	
<hr/>				
Overall:	4649	2926	155094	
<hr/>				
MPIO_path_1	4450	2926	29101	
MPIO_path_2	4847	2962	155094	
<hr/>				

Redundancy/Reliability Policies

Fail Over Only

Load DynamiX MPIO support provides a Redundancy/Reliability-oriented Policy for Fail Over/Fail Back (FOFB) called Fail Over Only (the default MPIO policy). With the Fail Over Only Policy selected, multiple paths to the same LUN are used to guarantee the SCSI Client Read or Write operations will successfully reach that target LUN. With Fail Over Only selected, 100% of traffic is sent over the Primary Path until it fails. Once the Primary Path connection is disrupted, traffic is moved to the Secondary Path. Once the Primary Path connection is restored, traffic will return to the Primary Path. The following project demonstrates the use of Fail Over Only policy. The Scenario opens the Primary (and Secondary Paths (lines 3 and 8) and connects to the desired LUN (lines 5 and 10). The MPIO Config Action binds the connections and LUNs into an MPIO configuration consisting of Fail Over Only Policy and Path 1 as Primary and Path 2 as Secondary.

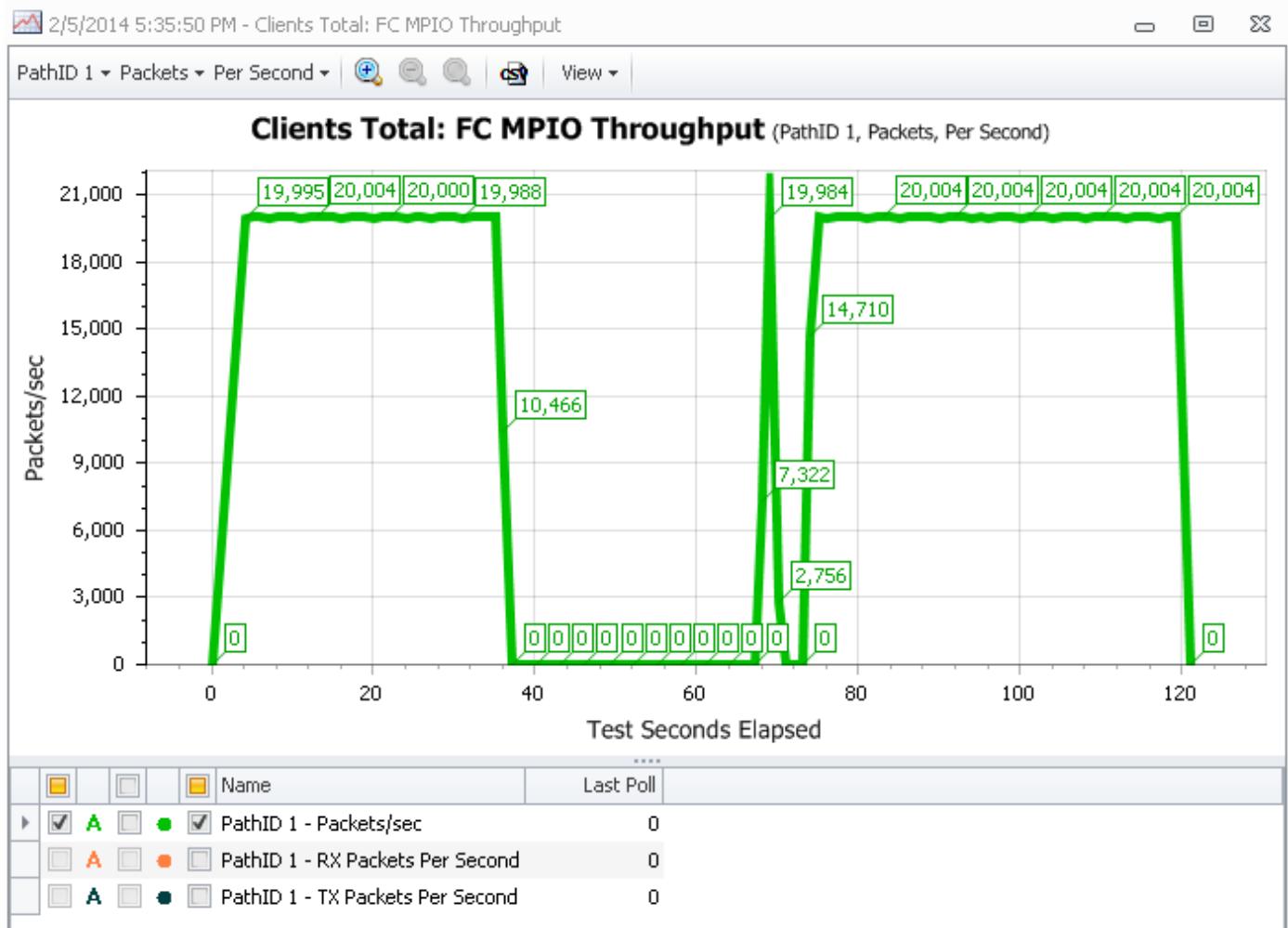
Client FC mpio config.client_scenario*

#	Protocol	Name
1	#	Open first MPIO Target
2	#	
3	FC	Open FC Connection
4	#	Establish Handle for first Target
5	SCSI	Read Capacity(10)
6	#	Open second MPIO Target
7	#	
8	FC	Open FC Connection
9	#	Establish Handle for second Target
10	SCSI	Read Capacity(10)
11	FC	Open FC Connection
12	#	Configure MPIO Ports and Targets
13	SCSI	MPIO Config

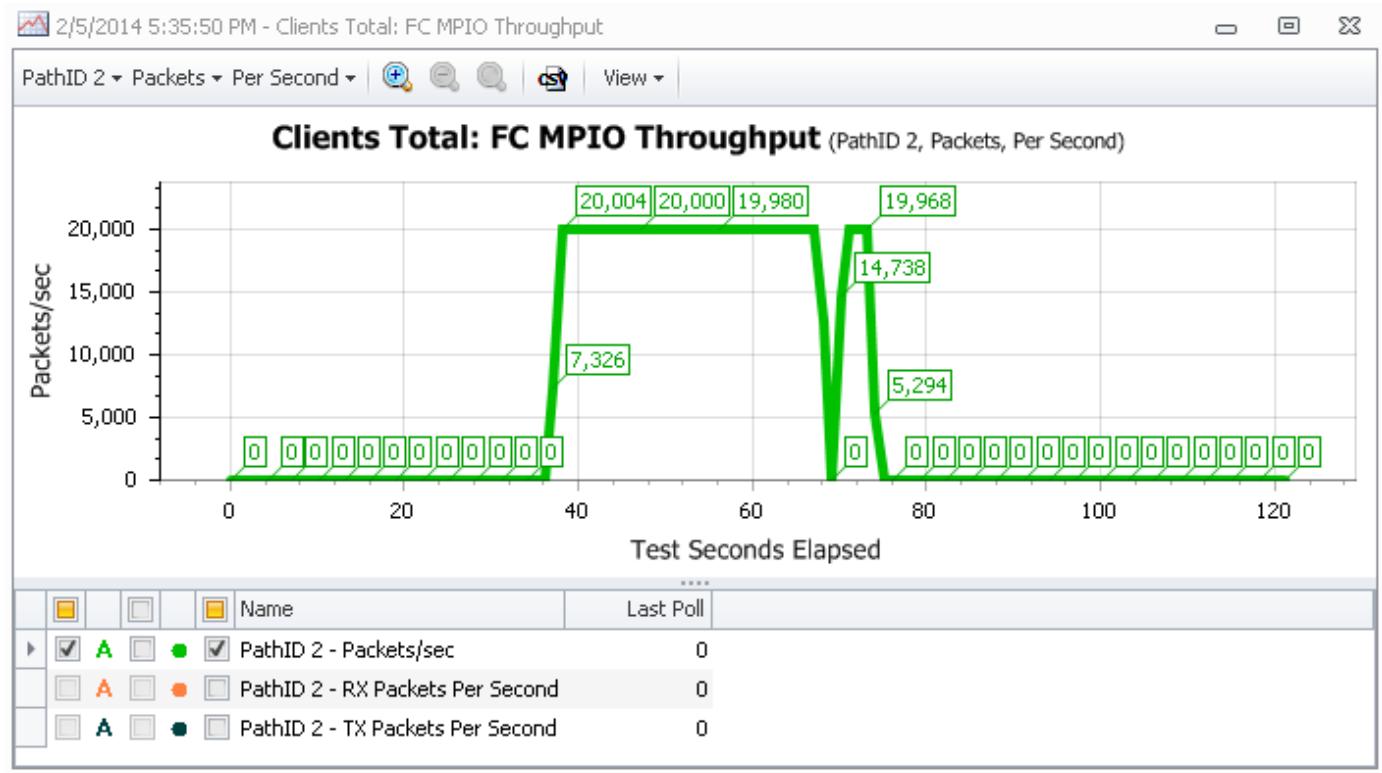
Name	Value
Input	
Policy	Fail Over Only
Enable ALUA Reconfig	False
Primary Path	
Primary Path LUN	5: LUNHandle
Primary Path ID	1
Secondary Path	
Secondary Path LUN	10: LUNHandle
Secondary Path ID	2
Additional Paths	0
Output	
Output Handle	13: LUNHandle

At 36 seconds in the run of the Project, the Primary Path fails and the Secondary Path takes over. The Primary Path is restored momentarily at 69 seconds and finally completely restored at 75 seconds. The following MPIO Throughput graphs show the behavior of Path 1 and Path 2

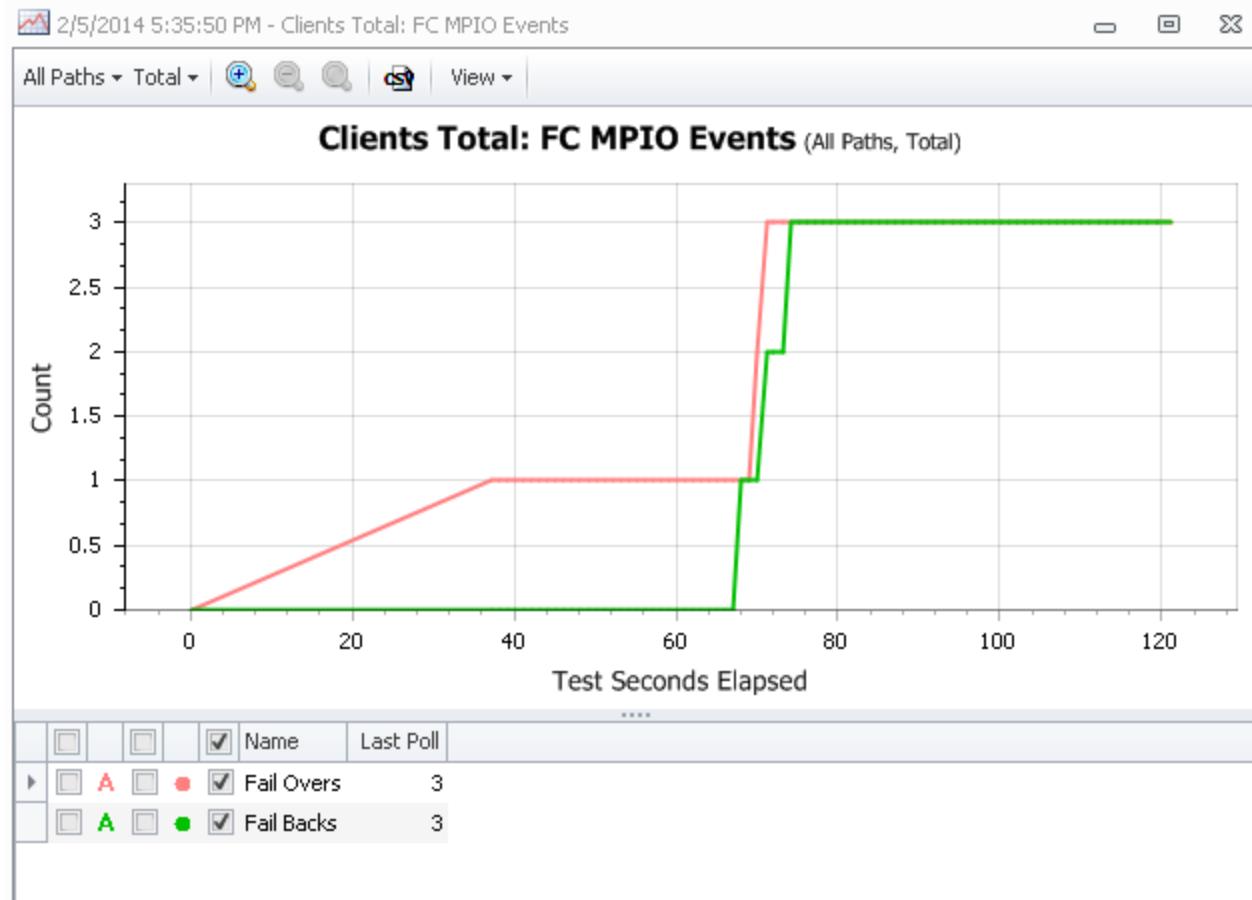
Primary Path - down from 36 seconds to 69 seconds and then again from 71 seconds to 75 seconds.



Secondary Path - active for roughly 37 seconds, 36 seconds to 69 seconds and 71 seconds to 75 seconds.



During the Project's 120 seconds of execution, there were 3 Fail Over events and 3 Fail Back events.



The 3 Fail Over Events correspond to the transitions from Primary to Secondary and the 3 Fail Back events correspond to the transitions from Secondary to Primary.

Weighted Paths

The Weighted Paths policy is a Fail Over/Fail Back policy in which Paths are given relative "Weights" that determine which Paths have a higher or lower Priority for carrying traffic. When Weighted Paths Policy is selected, Paths are assigned a Weighted Path value. The Path with the highest Priority for carrying traffic is the Path that has the lowest Weighted Path value. If/When this Path fails or is interrupted, the Path with next highest Priority (next highest "Weight") will start handling traffic. As soon as traffic can be carried by the higher priority Path, traffic will revert to the Path.

In the scenario below, four paths are connected in the **MPIO Config** Action using the Weighted Paths policy. Path ID 1 has the lowest weight thus the highest priority and is the Primary Path

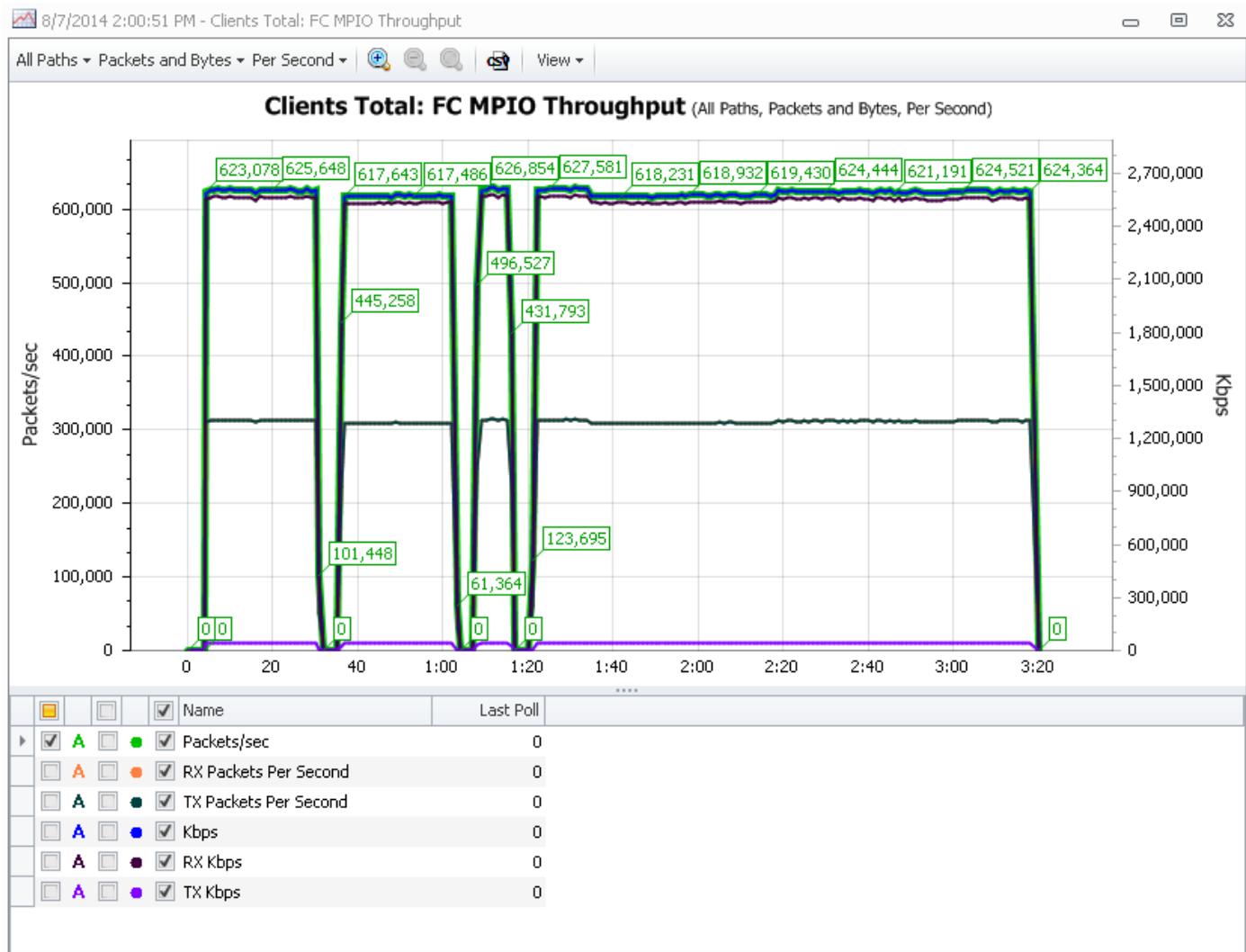
#	Protocol	Name
1	#	Open first MPIO Target
2	#	
3	FC	Open FC Connection
4	#	Establish Handle for first Target
5	SCSI	Read Capacity(10)
6	#	Open second MPIO Target
7	#	
8	FC	Open FC Connection
9	#	Establish Handle for second Target
10	SCSI	Read Capacity(10)
11	FC	Open FC Connection
12	SCSI	Read Capacity(10)
13	FC	Open FC Connection
14	SCSI	Read Capacity(10)
15	#	Configure MPIO Ports and Targets
16	SCSI	MPIO Config
17	SCSI	Verify(10)
18	#	MPIO Reads and Writes
19	SWT	Begin Loop

Name	Value
Policy	Weighted Paths
Enable ALUA Reconfig	False
Primary Path	
Primary Path LUN	5: LUNHandle
Primary Path ID	1
Primary Weight	1
Secondary Path	
Secondary Path LUN	10: LUNHandle
Secondary Path ID	2
Secondary Weight	2
Additional Paths	2
Path 3	
Path 3 LUN	12: LUNHandle
Path 3 ID	3
Path 3 Weight	3
Path 4	
Path 4 LUN	14: LUNHandle
Path 4 ID	4
Path 4 Weight	4

When traffic is interrupted on the Path ID 1 (the highest priority Path), traffic switches to Path ID 2, the Path with the next highest priority based on its "Weight" (2). When traffic is interrupted on Path ID 2 and Path ID 1 is still down, traffic moves to Path ID 3. Traffic reverts to Path ID 2 (highest priority Path that is on line) from Path ID 3 when Path ID 2 is able to carry traffic again. When Path ID 2 is interrupted again and Path ID 1 and 3 are still not operational, traffic moves onto Path ID 4. Traffic returns to Path ID 1 when it is able to carry traffic again.

The graphs below demonstrate this behavior (note that the Pkts/Sec performance of all of the Paths are relatively the same, it is the starting time and length of time that each Path handles traffic that varies).

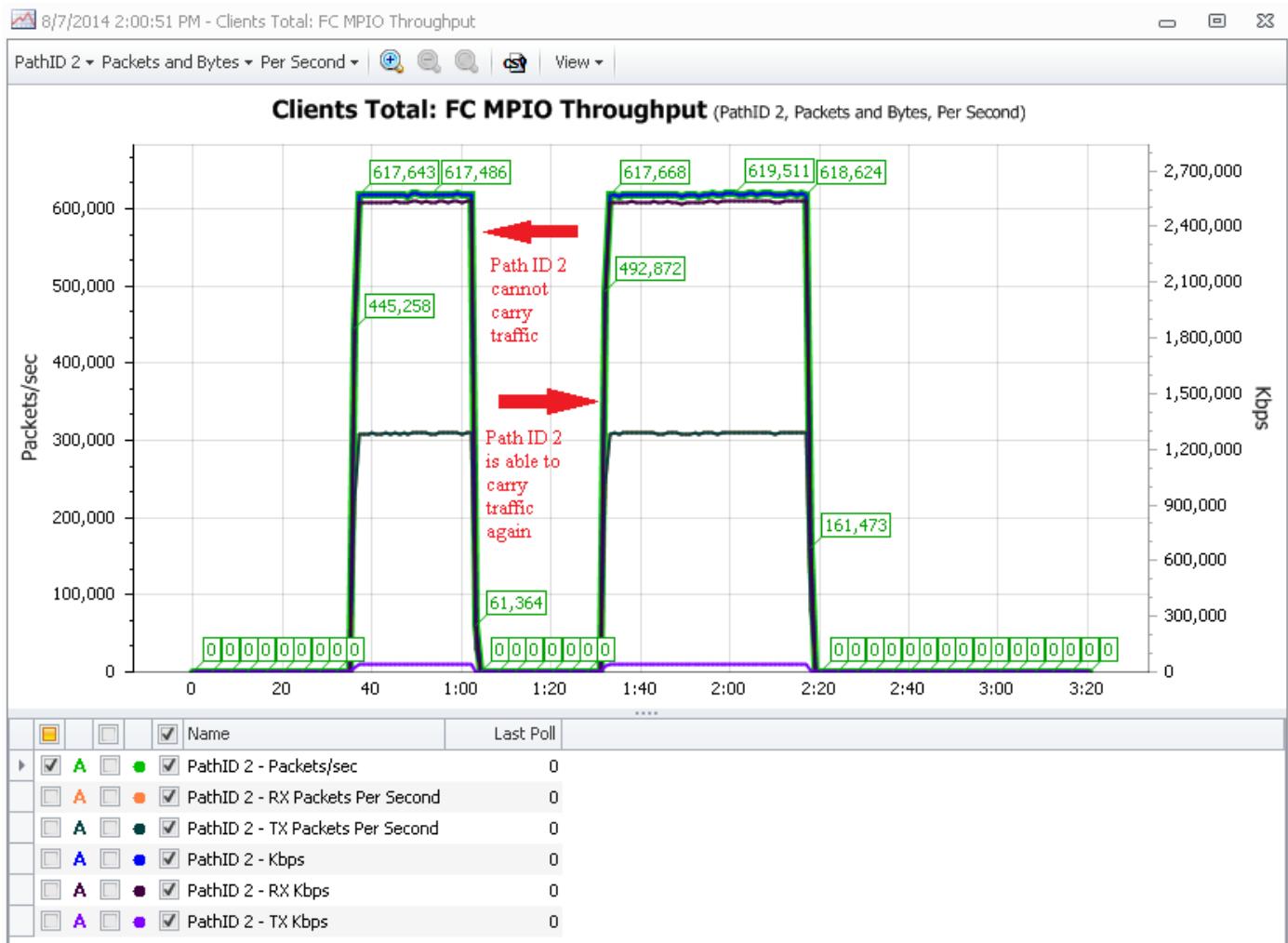
All Paths MPIO Throughput



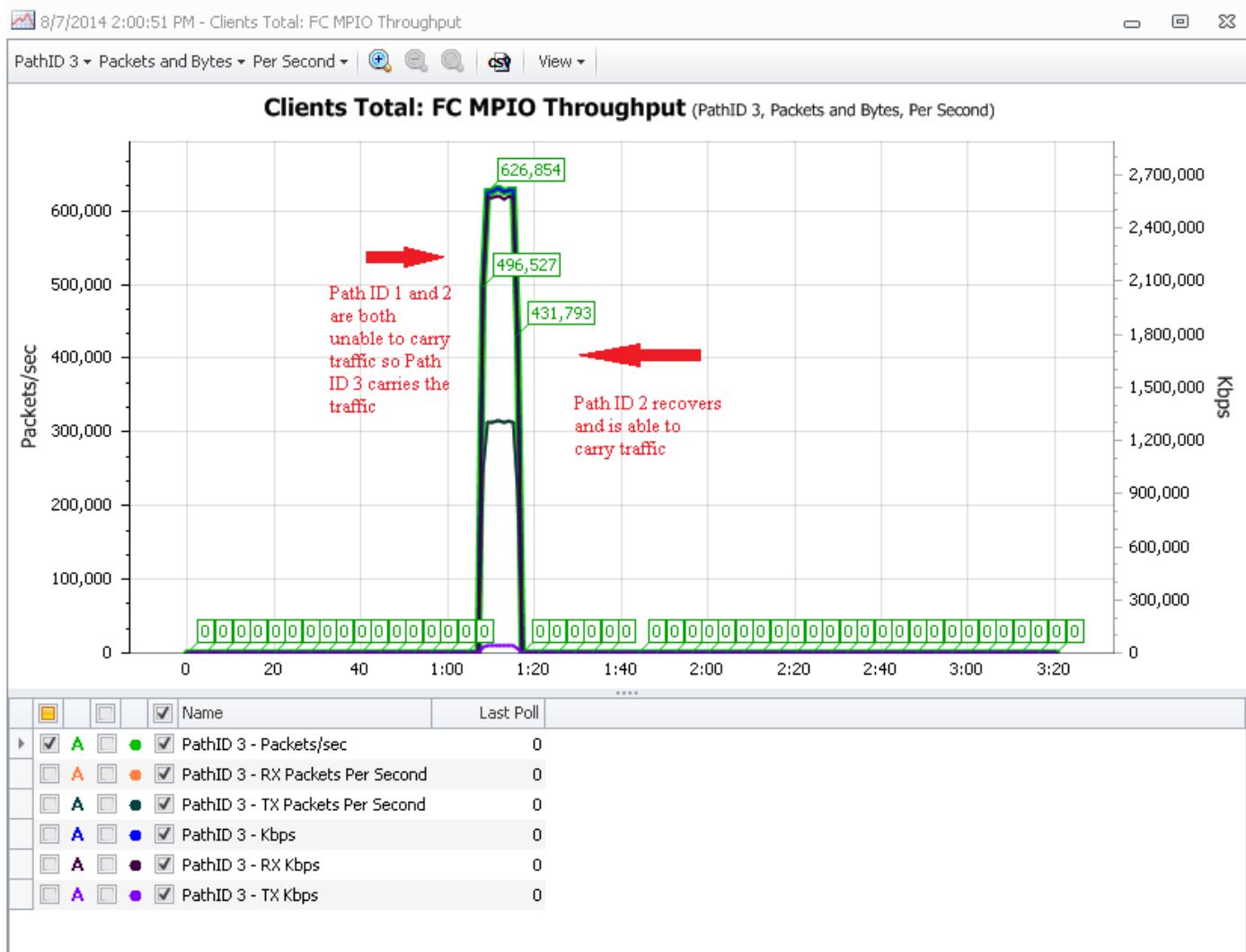
Path ID 1 MPIO Throughput



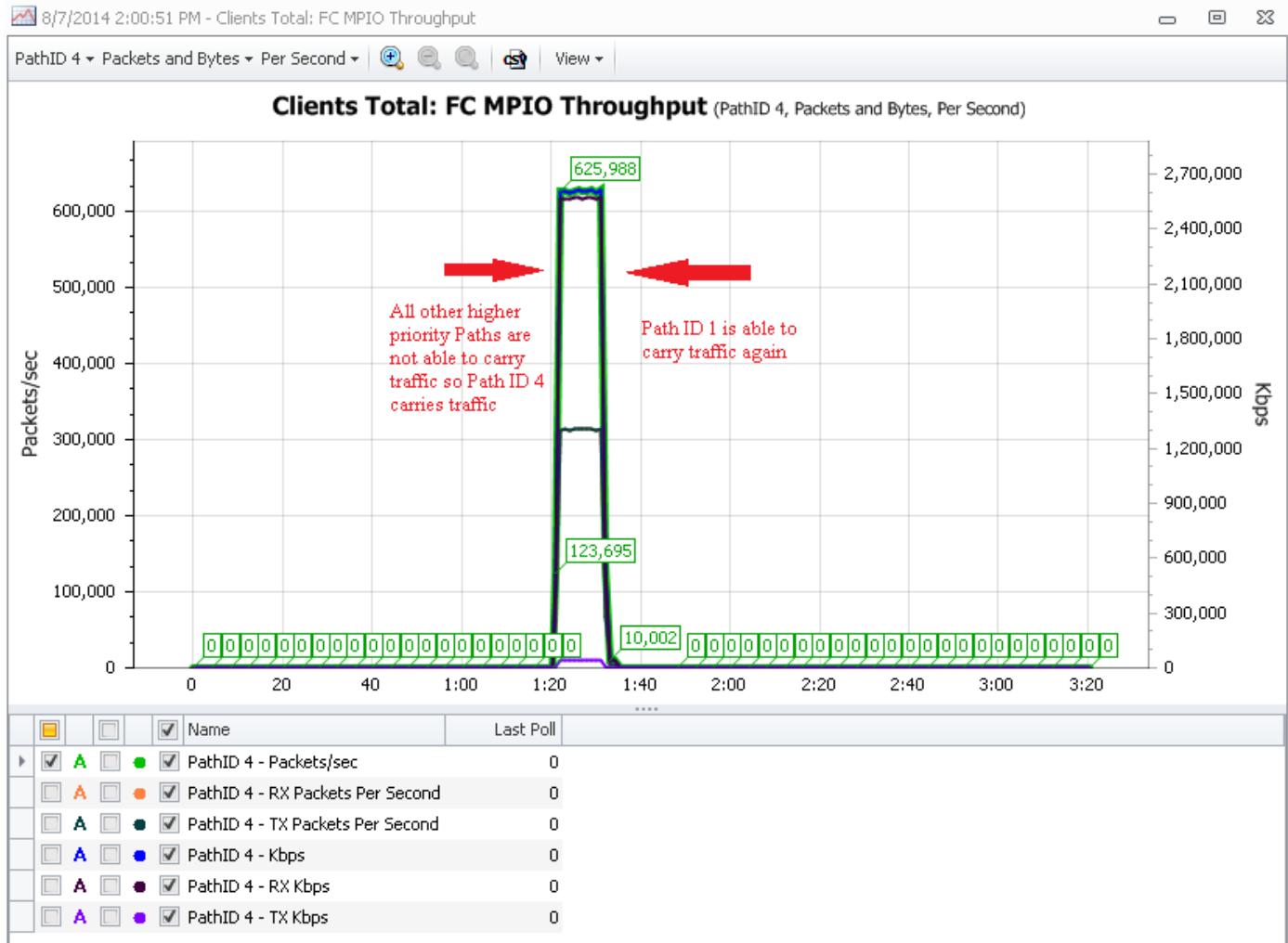
Path ID 2 MPIO Throughput



Path ID 3 MPIO Throughput



Path ID 4 MPIO Throughput



ALUA

ALUA stands for Asymmetric Logical Unit Access and it, in conjunction with MPIO Fail Over and Fail Back capabilities, provides Fibre Channel/iSCSI Clients and SAN Storage Targets with highly available storage that requires minimal configuration support. ALUA-enabled Storage Targets can inform the ALUA-enabled Fibre Channel/iSCSI Clients of the state of LUNs on a given Data Path which allows the Fibre Channel/iSCSI Clients to make decisions regarding accessing these LUNs.

ALUA-enabled LUNs exist in one of four Access states:

- Unavailable: LUN is only accessible by a limited set of SCSI meta-data commands
 - **Inquiry**
 - **Report Luns**
 - **Report Target Port Groups**
 - **Set Target Port Groups**
 - **Request Sense**
 - **Read Buffer**
 - **Write Buffer**
- Standby: LUN is only accessible by a limited set of SCSI meta-data commands
 - **Inquiry**
 - **Log Select**
 - **Log Sense**
 - **Mode Select**
 - **Mode Sense**
 - **Report LUNS**

- **Receive Diagnostic Results**
- **Send Diagnostic**
- **Report Target Port Groups**
- **Set Target Port Groups**
- **Request Sense**
- **Persistent Reserve In**
- **Persistent Reserve Out**
- **Read Buffer**
- **Write Buffer**
- Active/Optimized: LUN is available for use and responds to all commands at optimal performance
- Active/Non-Optimized: LUN is available for use but certain operations such as caching and data transfer may execute at sub-optimal performance

Load DynamiX ALUA support provides the Tester with the ability to query the state of ALUA enabled LUNs and get the results of that query in the Client Log file and to Query and Set the ALUA state of the LUN.

ALUA Discovery: Discover the ALUA Management state of a given LUN on an open Fibre Channel/iSCSI Connection

Report Target Port Group: Report the Port Group Access state of a given LUN on an open Fibre Channel/iSCSI Connection

Set Target Port Group: Explicitly set the Port Group Access state of up to four Port Groups of a given LUN on an open Fibre Channel/iSCSI Connection

The following Scenario demonstrates the use of MPIO Actions and ALUA Actions in a Fibre Channel Client Scenario

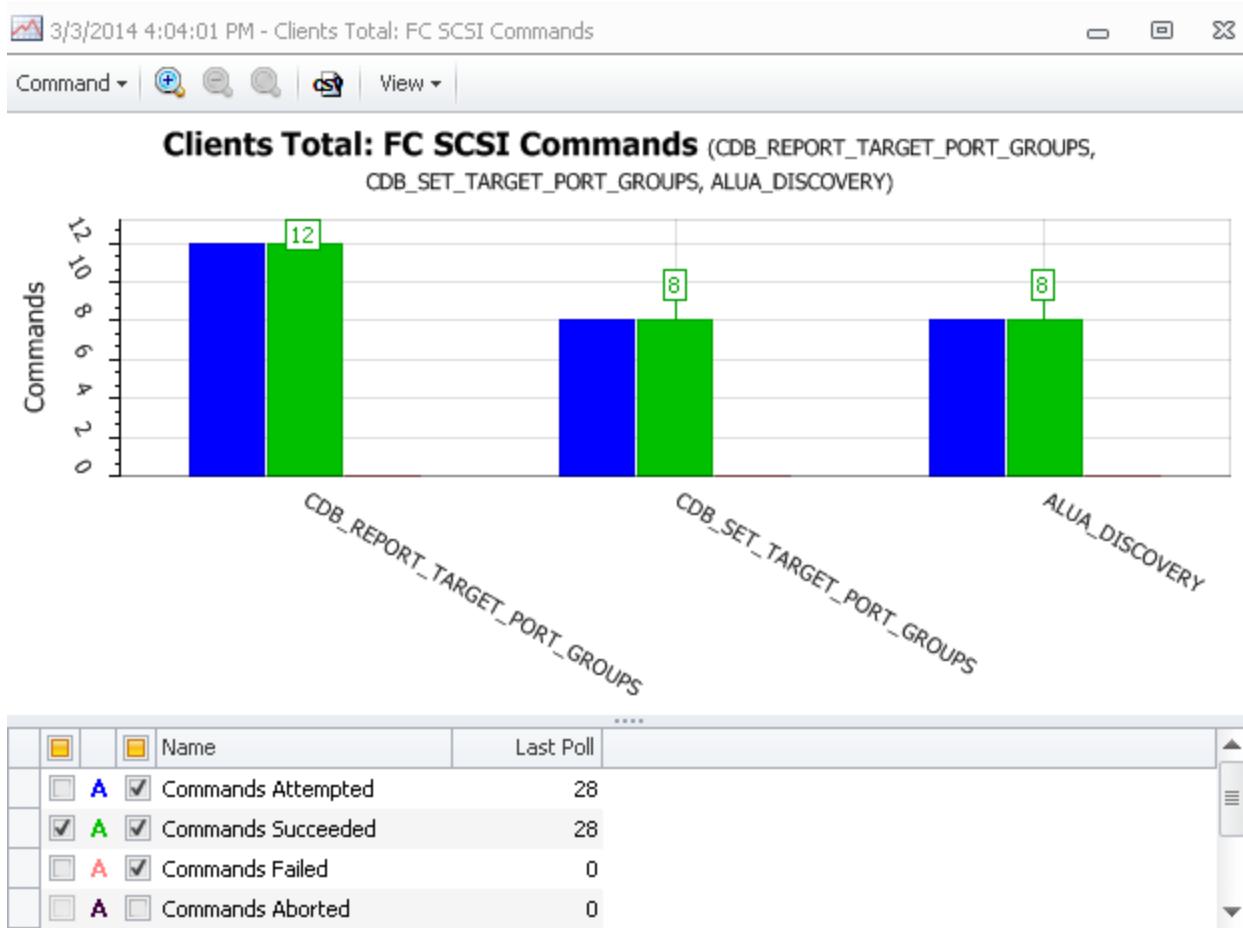
The screenshot shows a software interface for managing storage configurations. On the left, a list of actions is displayed in a table format:

#	Protocol	Name
1	#	Open first MPIO Target LUN 0
2	FC	Open FC Connection
3	#	Establish Handle for first Target
4	SCSI	Read Capacity(10)
5	#	Open second MPIO Target
6	FC	Open FC Connection
7	#	Establish Handle for second Target
8	SCSI	Read Capacity(10)
9	#	Open FC connection for ALUA operations
10	FC	Open FC Connection
11	#	Establish Handle for third Target
12	SCSI	Read Capacity(10)
13	#	Discover ALUA Management state LUN 0
14	SCSI	ALUA Discovery
15	#	Query ALUA Access state LUN 0 to Active Optimized
16	SCSI	Report Target Port Groups
17	#	Set ALUA Access state LUN 0
18	SCSI	Set Target Port Groups
19	SCSI	MPIO Config

On the right, a detailed configuration pane shows the settings for action #10: Establish Handle for first Target. It includes fields for Connection Handle (set to 10: SCSIConnectionHandle), LUN (0), Number Of Groups (1), and a Group 1 section with ID (0) and State ((0) Active/optimized). Below this, there are sections for Response Handlers (Completion Status, Scenario Impact) and a list of ALUA Actions (Discover, Query, Set).

ALUA Statistics

ALUA operations are graphed in the FC/iSCSI SCSI Actions and FC SCSI Commands graphs and in the Client Log file. Below, see the FC SCSI Commands graph showing the three ALUA Actions.



ALUA information is also logged in the Client Log file at the bottom with the Fibre Channel Port information. Highlighted information includes the Initiator and Target WWPN, which LUN is being accessed, the ALUA mode of the LUN and the ALUA information returned by the **Report Target Port Groups** and **ALUA Discovery** Actions.

```
=====
Initiator: 21:00:00:0e:1e:16:34:90
Target: 21:00:00:0e:1e:15:e3:41 (LUN0)
-----
Tx Frames: 3
Rx Frames: 6
Tx Bytes: 168
Rx Bytes: 874
ALUA : implicit/explicit
TPGID: 6
WWID : \001\0009LIO-0RG\000RAMDISK-MCP:436feb95-b852-47a9-ab1d-3a371433e7
-----
Target Port Group 0 Port ID: na Preferred bit count 0x00: 1 0x01: 0
State Status 1 (total)
0x00 active/optimized 0x01 explicit 1
-----
Target Port Group 5 Port ID: 1, 4 Preferred bit count 0x00: 1 0x01: 0
State Status 1 (total)
0x00 active/optimized 0x00 no status available 1
-----
Target Port Group 6 Port ID: 2, 3 Preferred bit count 0x00: 1 0x01: 0
State Status 1 (total)
0x00 active/optimized 0x00 no status available 1
=====
```

Real-Time Active/Passive Path switching based on ALUA messages

ALUA-enabled LUNs also have a Management mode. An ALUA-enabled port may be **Implicitly** or **Explicitly** Managed:

Implicit Management: the LUN Access State is managed **implicitly** by the SCSI target device. Implicit Management of a SCSI target is driven by target-initiated state changes. When an ALUA-enabled SCSI target that is actively being read from or written to by a Load DynamiX Project, changes state that makes it no longer a viable target (ex: changes to state from Active/Optimized to Standby or Unavailable) then the change of I/O behavior by the Load DynamiX Appliance Firmware is dictated by the configuration of the Project (policy type, ports, LUNs etc configured by the MPIO Config Action). Loss of traffic is kept to 0 and I/O continues on the alternate path.

Explicit Management: the LUN Access State is managed **explicitly** by the SCSI Client using the **Report Target Port Groups** and **Set Target Port Groups** commands.

Explicit Management of the SCSI Target is driven by Actions executed in the Project. A Project that is Explicitly Managing an ALUA-enabled target must configure the Project allow ALUA changes (Enable ALUA Reconfig field = True):

and then use the **Set Target Port Groups** Action to set the state of the target:

When a Project changes the state of a target to state in which I/O is no longer valid (ex: change state from Active/Optimized to Standby) then the processing by the Load DynamiX Firmware described above in the Implicit Management discussion takes place.

Implicit/Explicit Management: the LUN Access State can be managed Explicitly by the Client or Implicitly by the Target.

This information can be queried on a per LUN basis using the **ALUA Discovery** Action and is reported in the Client Log file for the Scenario that contains the **ALUA Discovery** Action

219	D...	1/27/2014 5:06:32 PM	=====
220	D...	1/27/2014 5:06:32 PM	Initiator: 21:00:00:1b:32:91:7d:f3
221	D...	1/27/2014 5:06:32 PM	Target: 21:00:00:24:ff:45:f3:ba (LUN0)
222	D...	1/27/2014 5:06:32 PM	ALUA not supported
223	D...	1/27/2014 5:06:32 PM	=====
224	D...	1/27/2014 5:06:32 PM	Tx Frames: 8
225	D...	1/27/2014 5:06:32 PM	Rx Frames: 20
226	D...	1/27/2014 5:06:32 PM	Tx Bytes: 448

ALUA Not Supported

	Line	Type	Date / Time	Text
248	Debug	De...	1/27/2014 4:49:43 PM	=====
249	Debug	De...	1/27/2014 4:49:43 PM	=====
250	Debug	De...	1/27/2014 4:49:43 PM	Initiator: 21:00:00:24:ff:00:dd:6d
251	Debug	De...	1/27/2014 4:49:43 PM	Target: 50:0a:09:83:89:1a:dc:ec (LUN0)
252	Debug	De...	1/27/2014 4:49:43 PM	ALUA mode: implicit
253	Debug	De...	1/27/2014 4:49:43 PM	=====
254	Debug	De...	1/27/2014 4:49:43 PM	Tx Frames: 4
255	Debug	De...	1/27/2014 4:49:43 PM	Rx Frames: 10
256	Debug	De...	1/27/2014 4:49:43 PM	Tx Bytes: 224
257	Debug	De...	1/27/2014 4:49:43 PM	Rx Bytes: 4284

LUN is IMPLICITLY managed

	Line	Type	Date / Time	Text
248	Debug	De...	1/27/2014 4:47:11 PM	=====
249	Debug	De...	1/27/2014 4:47:11 PM	=====
250	Debug	De...	1/27/2014 4:47:11 PM	Initiator: 21:00:00:0e:1e:09:b3:10
251	Debug	De...	1/27/2014 4:47:11 PM	Target: 21:00:00:0e:1e:07:02:f8 (LUN0)
252	Debug	De...	1/27/2014 4:47:11 PM	ALUA mode: implicit/explicit
253	Debug	De...	1/27/2014 4:47:11 PM	=====
254	Debug	De...	1/27/2014 4:47:11 PM	Tx Frames: 4
255	Debug	De...	1/27/2014 4:47:11 PM	Rx Frames: 10
256	Debug	De...	1/27/2014 4:47:11 PM	Tx Bytes: 224
257	Debug	De...	1/27/2014 4:47:11 PM	Rx Bytes: 4284
258	Debug	De...	1/27/2014 4:47:11 PM	=====

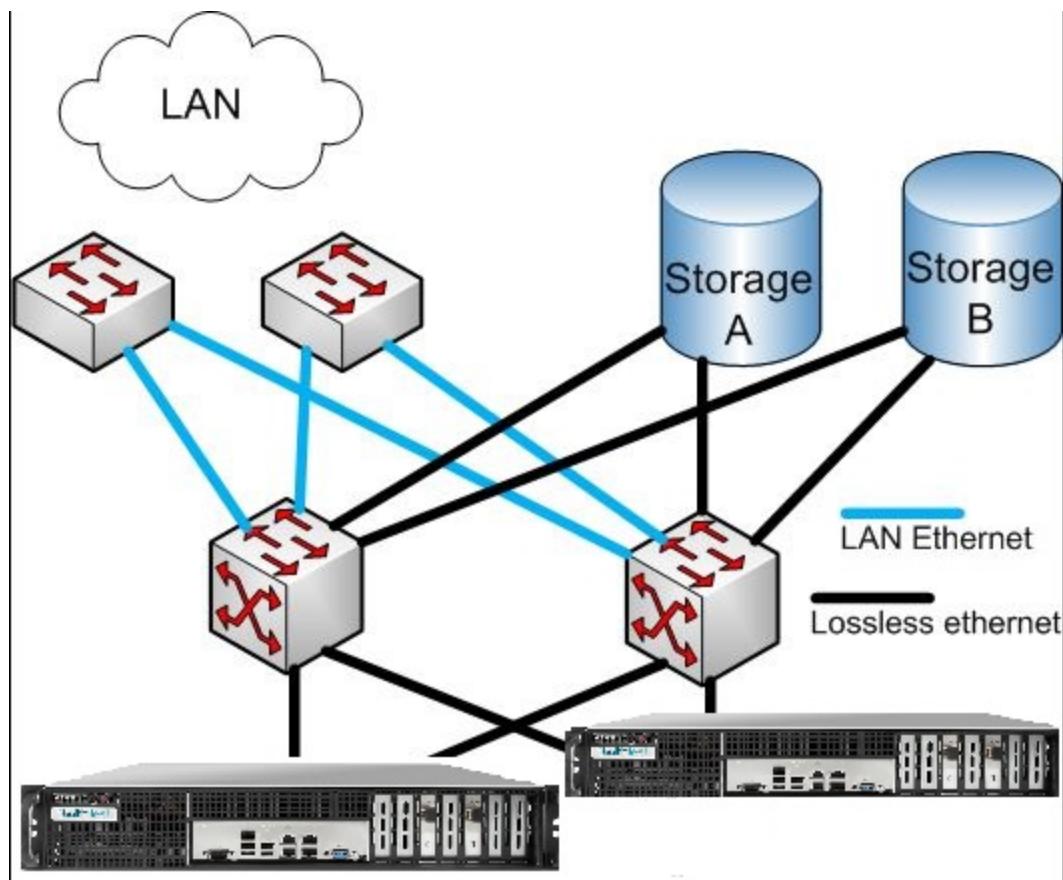
LUN is IMPLICITLY and EXPLICITLY managed

ALUA-related information captured in the Client log file from a Project run showing Target Port Group information (TPGID), ALUA management state and WWID (a globally unique identifier for the target LUN).

10/14/2014 10:24:43 PM	=====
10/14/2014 10:24:43 PM	Initiator: 21:00:00:0e:1e:16:34:90
10/14/2014 10:24:43 PM	Target: 21:00:00:0e:1e:15:e3:41 (LUN3)
10/14/2014 10:24:43 PM	-----
10/14/2014 10:24:43 PM	Tx Frames: 2
10/14/2014 10:24:43 PM	Rx Frames: 4
10/14/2014 10:24:43 PM	Tx Bytes: 112
10/14/2014 10:24:43 PM	Rx Bytes: 582
10/14/2014 10:24:43 PM	ALUA : implicit/explicit
10/14/2014 10:24:43 PM	TPGID: 5
10/14/2014 10:24:43 PM	WWID : \001\0009LIO-ORG\000RAMDISK-MCP:436feb95-b852-47a9-ab1d-3a371433e7

Fibre Channel over Ethernet

Fibre Channel frames can be encapsulated and delivered over an Ethernet network using the Fibre Channel over Ethernet (FCoE) protocol preserving the Fibre Channel lossless network semantics while achieving Ethernet performance. In the Load DynamiX implementation of FCoE, the Fibre Channel NIC with a special SFP+ transceiver (shipped with the Load DynamiX Appliance) and compatible active or passive Direct Attach cables, can be used to send and receive Fibre Channel frames over an Ethernet network to a FCoE SAN device (switch, server, etc.).



FCoE Ports will appear in the Ports & Appliances > Appliances tab with the following icon .

Ports & Appliances

Ports Appliances

Ping Stop Port Ports Status Link Status Clear Content Update Reboot

Port Details Rediscover Targets Clear NPIV Licenses

Port Id	6
Port Type	SwiftTest 16Gb Fibre Channel
WWNN	20:00:00:0e:1e:16:39:81
WWPN	21:00:00:0e:1e:16:39:81
FCID	0x12300
Link Speed	16 Gbps
BB Credit	5
Initiators	1
Initiator WWPN	21:00:00:0e:1e:16:39:81
FCID	0x12300
Targets	8
Target WWPN	20:01:00:11:0d:d2:b2:00
FCID	0x12c00
Target WWPN	21:00:00:0e:1e:09:11:0c
Target WWPN	21:00:00:0e:1e:09:b3:04
Target WWPN	21:00:00:0e:1e:09:b3:05
Target WWPN	21:00:00:0e:1e:15:e3:40
Target WWPN	21:00:00:0e:1e:15:e3:41
Target WWPN	25:00:00:21:88:00:54:4b
Target WWPN	25:10:00:21:88:00:54:4b

OK Cancel

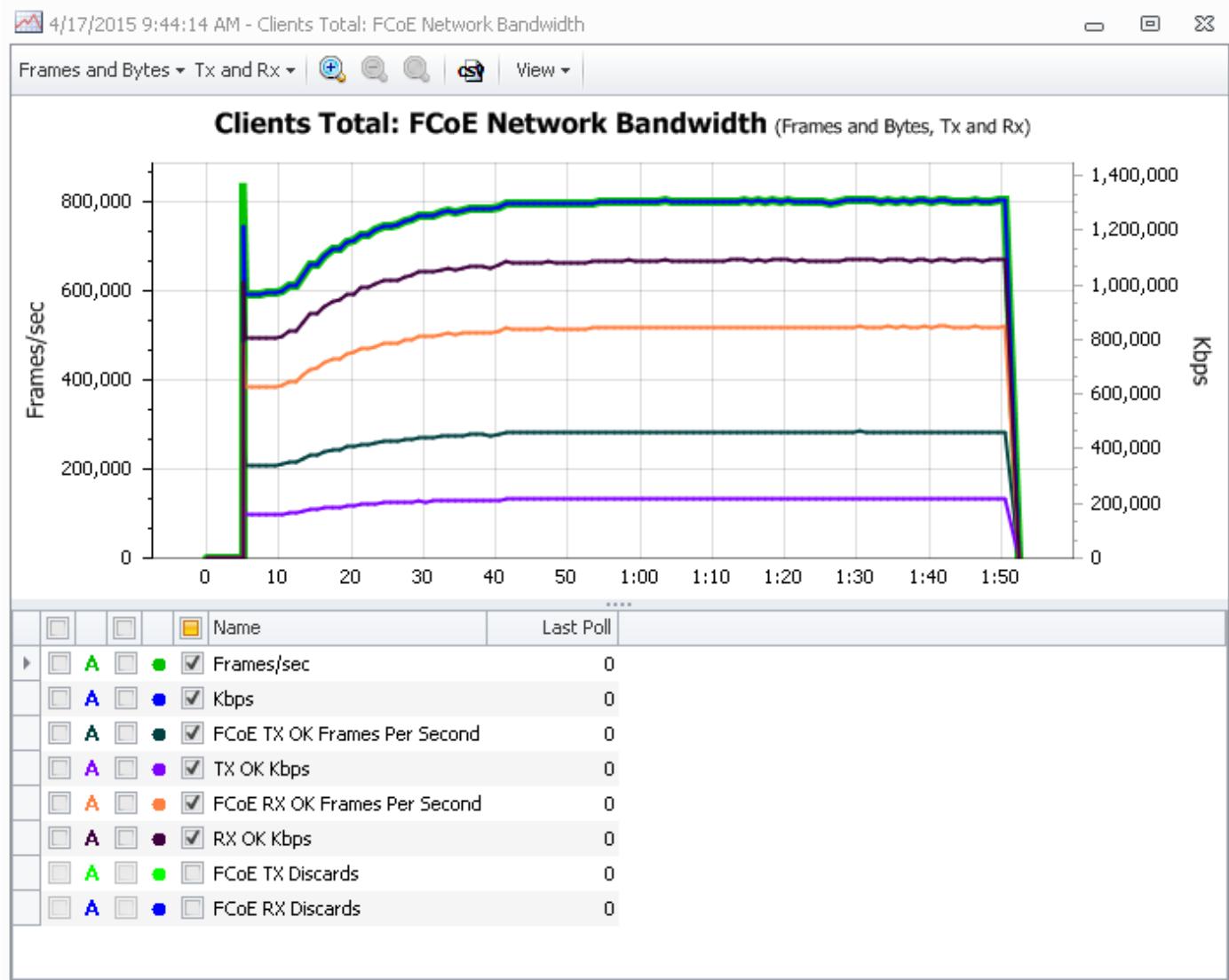
A Load DynamiX FCoE Project uses the same **Open FC Connection** and SCSI Actions as a non-FCoE Fibre Channel Project. The Load DynamiX FCoE interface handles the Ethernet encapsulation and send/receive logic. Assigning an FCoE Physical Port to a Project's Logical Port is all that is necessary to execute an existing Fibre Channel Project over an FCoE transport.

Client FCOE asynch.client_scenario*

#	Protocol	Name
1	FC	Open FC Connection
2	SCSI	Read Capacity(10)
3	#	MPIO Reads and Writes
4	SWT	Begin Loop
5	SWT	Begin Async
6	SCSI	LUN Write
7	SCSI	LUN Read
8	SWT	End Async
9	SWT	End Loop
10	FC	Close FC Connection

Name	Value
Input	
Source WWPN Defined	False
Target WWPN	21:00:00:E0:8B:91:10:B9
Connect Timeout (ms)	30,000
Reconnect	
Maximum Attempts	0
Delay Interval (ms)	1,000
Output	
Output Handle	1: SCSIConnectionHandle

The Load DynamiX FCoE implementation replaces the FC Network Bandwidth statistics with the FCoE Network Bandwidth statistics.



The Client Log file for an FCoE project contains the following statistics

```
=====
FCoE: Port 4
-----
Tx Frames:      29767880
Rx Frames:      54574480
Tx Bytes:       2857708480
Rx Bytes:       14367880480
Tx Frame Discards: 0
Rx Frame Discards: 0
=====
```

All other aspects of an FCoE Project (statistics, PCAP, supported SCSI Actions, FC-specific features such as MPIO/ALUA), remain the same as any Load DynamiX Fibre Channel Project.

BB Credits (Buffer to Buffer Credits)

BB Credits is a Fibre Channel flow control mechanism. It specifies to a device (switch or target) on the other end of the Fibre Channel connection how many full frame buffers this device has. This

value is currently fixed and is set to 5.

iSCSI/Fibre Channel/FCoE Caveats

Currently, the information returned by a Fibre Channel or iSCSI device in response to the SCSI Read Capacity command (example LUN numbers, LUN size and Block Size) is not directly accessible by a Scenario so it is up to the Tester to know this information in advance and use it in Scenarios via User Parameter files or hard coding into Action input.

SCSI Read and Write Actions (and most other SCSI Actions) do not maintain state (example: SCSI Read Actions do not remember where the last read completed so that the next one can start there). It is up to the Tester to maintain this state information in Scenarios.

SCSI LUN Read and LUN Write are not defined in the SCSI command specification - they are aggregate commands that allow the Tester to specify if the Read or Write are a 6, 10, 12 or 16.

Fibre Channel processing of invalid requests. The lower level Fibre Channel interfaces on the Load DynamiX 6202/6204/6208/6202E

Appliances will not send SCSI requests to FC devices that are determined by the low level interfaces to contain invalid information. A simple example is LUN number. If the LUN field of a SCSI Read Capacity or Test Unit Ready command (for example) contains 10 and the targeted FC device does not have LUN 10, the Fibre Channel interfaces will not send the command. The command will be marked with error reason "Transmission Failed" even though transmission was never really attempted. The Transmission Failed error condition cannot be handled by the SCSI Response Handling code so the scenario will always fail in this condition.

Fibre Channel Read and Write Chunk Size limits vary depending on the target FC device. Observed maximum is 128MB. Using Chunk Size values greater than 128MB will result in failed Read and Write commands.

MPIO Scenarios expect all Paths and LUNs to be reachable when the Scenario starts. However the Fibre Channel Reconnect feature allows Projects to continue working if a connection fails. See the Fibre Channel Reconnect discussion above.

Currently when any MPIO Project is running on an Appliance, it is the ONLY Project that can be running on that Appliance, regardless of how many Ports it is using and how many are available on the Appliance.

FCoE does not support:

- Configuring DCB or FIP
- Multi-hop FCoE
- PCAP cannot capture FIP or DCB packets
- FCoE traffic only on a Load DynamiX FCoE Port

Fibre Channel Sample Project

Client FC All Commands.client_scenario*

#	Protocol	Name
1	FC	Open FC Connection
2	SCSI	Test Unit Ready
3	SCSI	Report LUNs
4	SCSI	Read Capacity(10)
5	SCSI	Read Capacity(16)
6	SCSI	Inquiry
7	SCSI	Mode Sense(6)
8	SCSI	Mode Sense(6)
9	SCSI	Mode Sense(6)
10	SCSI	Mode Sense(6)
11	SCSI	Mode Sense(10)
12	SCSI	Mode Sense(10)
13	SCSI	Mode Sense(10)
14	SCSI	Mode Sense(10)
15	SCSI	Mode Select(6)
16	SCSI	Mode Select(10)
17	SCSI	Start/Stop Unit
18	SCSI	Synchronize Cache(...)
19	SCSI	Verify(10)
20	SCSI	LUN Read
21	SCSI	Read(6)
22	SCSI	Read(10)
23	SCSI	Read(12)
24	SCSI	Read(16)
25	SCSI	LUN Write
26	SCSI	Write(6)
27	SCSI	Write(10)
28	SCSI	Write(12)
29	SCSI	Write(16)
30	FC	Close FC Connection

Name

Input

- Connection Handle
- LUN
- Repeat

Response Handlers

- Completion Status
- Scenario Impact

Client FC SSC All Commands.client_scenario*

#	Protocol	Name
1	FC	Open FC Connection
2	SCSI	Test Unit Ready
3	SCSI	Read Block Limits
4	SCSI	Report Density Support
5	SCSI	Prevent Allow Medium Removal
6	SCSI	Sense Block Size
7	SCSI	Select Block Size
8	SCSI	Set Capacity
9	SCSI	Rewind
10	SCSI	Erase(16)
11	SCSI	Erase(6)
12	SCSI	Write Filemarks(16)
13	SWT	Begin Loop
14	SCSI	SSC Write(6)
15	SCSI	SSC Write(16)
16	SCSI	Recover Buffered Data
17	SCSI	Verify(6)
18	SCSI	Verify(16)
19	SCSI	Write Filemarks(6)
20	SWT	End Loop
21	SCSI	Rewind
22	SWT	Begin Loop
23	SCSI	SSC Read(6)
24	SCSI	SSC Read Reverse(6)
25	SCSI	Space(6)
26	SWT	End Loop
27	SCSI	Rewind
28	SCSI	Read Position
29	SCSI	Rewind
30	SCSI	Locate(10)
31	SCSI	Rewind
32	SCSI	Space(16)
33	SCSI	Load Unload
34	SCSI	Load Unload
35	FC	Close FC Connection

Name

Input

- Tape Handle
- Total Transfer Size
- FIXED
- Chunk Size

Data Verification

- Verification Data Source
- Verification Data Offset
- Verification Data Offset Type
- Number of Outstanding Requests

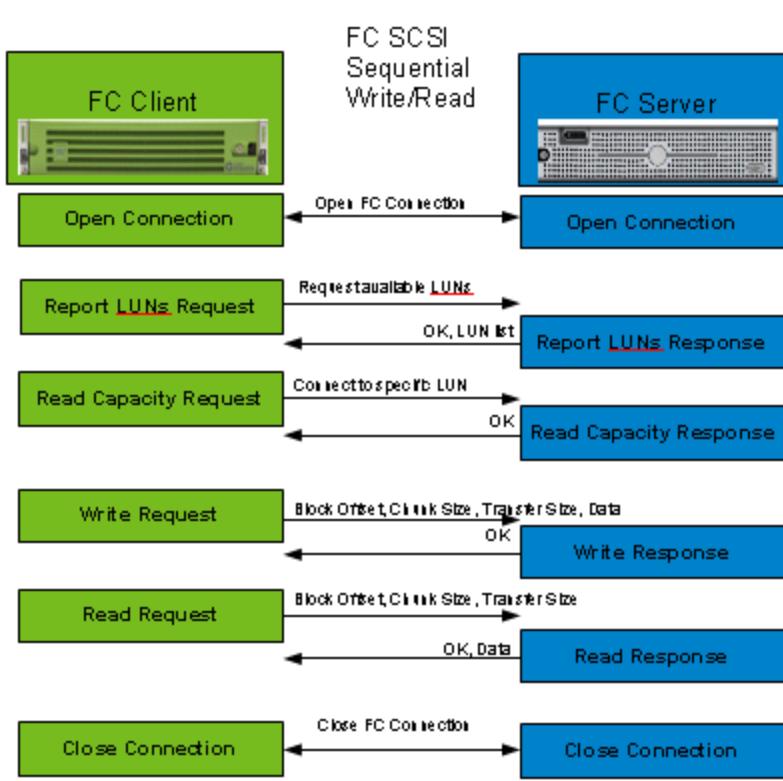
Response Handlers

- Completion Status
- Scenario Impact

OK Cancel

Fibre Channel Project Flow

The following is the Fibre Channel client/server interaction flow for a simple sequential write read Project.



Fibre Channel/FCoE/SCSI/iSCSI Statistics

Fibre Channel and FCoE Response Time statistics

In FC and FCoE, the Response Time statistic measures the total elapsed time from the instant when the first byte of a SCSI IO Request is transmitted from the Load DynamiX SCSI layer, to the instant when the last byte of a Response containing a Status Code to the corresponding SCSI IO Request is received by the Load DynamiX SCSI layer.

It is important to note that the Response Time measurement includes the intrinsic latency introduced by several lower layers, and potentially multiple networked devices, in addition to the Target's latency from processing the Request and generating the Response. These lower layers include the underlying operating system software, OS, HBA / CNA, fabric switches, as well as cabling. While the contribution from these lower layers and networked devices is usually small, in highly congested scenarios, the Response Time may seem larger than expected from the Target's perspective. However, the Response Time reported is accurate, from the vantage point of the Load DynamiX SCSI layer.

Response Time and other performance-related statistics can be impacted in a positive or negative way by the Port Queue Depth setting which controls how many SCSI Actions from a Logical Port can be pending on a given Target. An overabundance of pending Actions on a Target could negatively impact the Response Time statistics for that Target. See Fibre Channel Logical Port Resource discussion earlier in this section for additional details on Port Queue Depth.

See the [Advanced Concepts: Test Execution Rules](#) for a complete list of the statistics generated when Fibre Channel, SCSI or iSCSI Actions are used in a Scenario.

Reference: TCP/UDP Echo and Discard Protocols Command List**Reference: Load DynamiX TCP/UDP Echo and Discard Protocols Command List****TCP/UDP Echo Protocol**

ACTION	
Close TCP Echo Connection	
Close UDP Echo Connection	
Open TCP Echo Connection	
Open UDP Echo Connection	
TCP Echo Transmit	
SERVER ACTION	
Start TCP Echo Server	
Start UDP Echo Server	

Discard Protocol

SERVER ACTION	
Start discard server	

Reference: Action Input Shorthand

Reference: Action Input Shorthand

The TDE supports the following shorthand for Multipliers in Action input fields that require an Integer as input. The shorthand for Multipliers for Kilobytes, Megabytes, Gigabytes, etc are shown below.

Shorthand	Value	Example
KB, kb, Kb, kB	1024	20kB = 20 * 1024
mb, MB, Mb, mB	1048576	10MB = 10 * 1048576
GB, gb, Gb, gB	1073741824	5gb = 5 * 1073741824
TB, tb, Tb, tB	1099511627776	700Tb = 700 * 1099511627776
K, k	1000	3K = 3 * 1000
M, m	10^6 (1000000)	13M = 13 * 1000000
G, g	10^9 (1000000000)	4G = 4 * 1000000000
T, t	10^12 (1000000000000)	9t = 9 * 1000000000000

Examples

SCSI LUN Read

SMB2 Write

HTTP GET

Shorthand Errors

If shorthand notation is not accepted in an input field, the following error indication will be presented .

Decimal Number Support

The TDE and Appliance support the use of Decimal Numbers wherever Integers are expected as input (Protocol Action input fields and Scenario Control Action input fields). A Decimal Number is a number of the form $X.Y$ where X and $Y \geq 0$. Examples of Decimal Numbers are 1.4, 3.3, 10.27. When Decimal Numbers appear as input, the Appliance will process them according to the following rule: The number as specified (e.g. 1.3) will be multiplied by the specified Shorthand (if present). The result of that multiplication will have the mathematical Floor() function applied to it. Decimal Numbers that have no Shorthand multiplier have the Floor function applied as is. The output of the Floor function is the final result. The Floor function is defined to be: $\text{Floor}(X) == \text{the largest integer less than or equal to } X$.

So, for example:

- 1.3 $\rightarrow \text{Floor}(1.3) == 1$
- 1.3KB $\rightarrow \text{Floor}(1.3 * 1024) == \text{Floor}(1331.2) == 1331$
- 1.3K $\rightarrow \text{Floor}(1.3 * 1000) == \text{Floor}(1300.0) == 1300$
- 0.4 $\rightarrow \text{Floor}(0.4) == 0$

Being allowed to use Decimal Numbers is useful when specifying byte counts or blocks where the desired number of bytes or blocks is not a whole integer times a multiplier such as 3mb or 1K but is some fraction in addition such as 3.7mb or 1.9K.

Reference: HTTP Storage Commands and Behaviors

Reference: HTTP Storage Commands and Behaviors

Links to Amazon S3, CDMI, OpenStack protocol reference materials are provided in the [References and Terminology section](#).

HTTP Storage Commands (Actions)

CDMI

Container Operations
Create Container
Read Container
Update Container
Delete Container
Data Object Operations
Create Data Object
Read Data Object
Update Data Object
Delete Data Object
Domain Object Operations
Create Domain Object
Read Domain Object
Update Domain Object
Delete Domain Object
Queue Object Container
Create Queue Object
Read Queue Object
Update Queue Object
Delete Queue Object
Capabilities Object Operations
Read Capabilities Object

OpenStack**Swift Object Storage**

Container Operations
Create Container
Delete Container
List Container Contents
Retrieve Container Metadata
Create/Update Container Metadata
Delete Container Metadata
Data Object Operations
Create/Update Object
Copy Object
Delete Object
Retrieve Object
Retrieve Object Metadata
Update Object Metadata
Service Operations
Authentication
Retrieve Account Metadata
Create/Update Account Metadata
Delete Account Metadata
List Containers

Cinder Block Storage

API Versions
API Version List
API Version Details Show
API Extensions
API Extensions List
Volumes
Volume Create
Volume Delete
Volume List
Volume List Details
Volume Show
Volume Update
Volume Types
Volume Type List
Volume Type Create
Volume Type Show
Volume Type Delete
Snapshots
Snapshot Create
Snapshot List
Snapshot List Detailed
Snapshot Show
Snapshot Update
Snapshot Delete
Snapshot Metadata Show
Snapshot Metadata Update
Quality of service (QOS) specifications
QoS Create
QoS List
QoS Show
QoS Delete
QoS Associate
QoS Disassociate
QoS Disassociate All
QoS Get Association
Quota sets extension
Quota Show
Quota Update
Quota Delete
Quota Default
Quota Show for User
Quota Update for User
Quota Delete for User
Quota Details Show for User
Limits extension
Absolute Limits Show
Backups
Backup Create
Backup List
Backup List Details
Backup Show
Backup Delete
Backup Restore
Volume manage extension
Volume Manage

Keystone Identity Service

Authentication

Amazon S3

Service Operations
GET Service
Bucket Operations
GET Bucket
PUT Bucket
DELETE Bucket
PUT Bucket ACL
GET Bucket ACL
HEAD Bucket
GET Bucket cors
PUT Bucket cors
DELETE Bucket cors
GET Bucket policy
PUT Bucket policy
DELETE Bucket policy
GET Bucket tagging
PUT Bucket tagging
DELETE Bucket tagging
List Multipart Uploads
Object Operations
PUT Object - Copy
PUT Object
GET Object
POST Object
DELETE Object
GET Object ACL
PUT Object ACL
HEAD Object
Initiate Multipart Upload
List Parts
Upload Part
Upload Part - Copy
Abort Multipart Upload
Delete Multiple Objects
Complete Multipart Upload

HTTP Storage Action Behaviors, Authentication Notes

HTTP Storage should be thought of as on demand access of networked storage using FQDN URLs.

Load DynamiX HTTP Storage support includes four storage-oriented protocols: Cloud Data Management Interface (CDMI), OpenStack Swift, OpenStack Cinder and Amazon S3. All Load DynamiX HTTP Storage protocols support the use of the FQDN via the HTTP and HTTPS Open Connection Actions.

The CDMI commands use the CRUD nomenclature: Create, Read, Update and Delete for all supported object types (Container, Data, Domain and Queue).

The OpenStack Swift commands follow a similar CRUD nomenclature (Create and Create/Update, Retrieve, and Delete and add a List Command and also MetaData Commands for each supported object type: Container and Data. Additionally, Service Operations Commands are supported to handle Authentication and Account management.

The OpenStack Cinder provides block filesystem behaviors for OpenStack compute servers. Cinder provides filesystem building blocks such as Volumes, Volume Types, Backups, QoS, Quotas, Limits

and Snapshots, managed with commands that Create, Delete, List, Update, and Show. Multiple API versions with extensions are supported and there are additional API-related commands to query the supported APIs and extensions. Key Cinder concepts:

- Volume: A volume is a detachable block storage device
- Volume Type: SSD (solid state), SATA (rotating)
- Quota: Limits on key elements of a Block Filesystem such as the maximum # of CPU cores that service the filesystem, the maximum number of bytes in a file or file path, the number of files allowed, etc.
- Snapshot: A point in time copy of the data that a Volume contains
- Limit: Show absolute limits for a tenant for all filesystems that belong to that tenant such as maximum gigabytes that can be used, maximum number of snapshots used, maximum number of volumes, etc.
- Backup: A full copy of a Volume stored in an external service
- QoS: Basic reliability parameters
- Retry Mechanism: This mechanism has been added to certain longer-operating Cinder Actions to allow for these Actions to complete before moving to the next Cinder operation. The mechanism behave differently for two sets of Cinder Actions:
 - Volume Show, Backup Show and Snapshot Show: retry mechanism resends the request up to Maximum Retries times while the server returns the state in Retry on Status.
 - Volume List, Backup List and Snapshot List: retry mechanism resends the request up to Maximum Retries times while the server returns the ID specified.

Amazon S3 follows a more HTTP-like nomenclature for its commands: PUT, GET, DELETE, HEAD and POST. Amazon Simple Storage Service (Amazon S3) is storage for the Internet. Amazon S3 is an object storage service like OpenStack Swift and CDMI. Amazon S3 operates with entities similar to folders (Buckets) and files (Objects). Amazon S3 stores data as Objects within Buckets. Buckets are containers for Objects and have access controls (ACL - who can create, delete, and list objects in the bucket). Access logs can be viewed for Buckets and their Objects and the geographical region where the Bucket and its contents are stored can be chosen. An Object is comprised of a file and optionally metadata that describes that file. To store an Object in Amazon S3, upload a file to a Bucket. When a file is uploaded permissions (ACL) can be set on the file and optional metadata.

Key HTTP/HTTPS concepts like Authentication, Transfer Encoding, Header and Body Parsing, Dynamic Content, etc described in [Advanced Concepts: HTTP/HTTPS](#) apply variously to the HTTP-based HTTP Storage protocols:

- Transfer Encoding is only supported on the receive side of of HTTP Storage Protocols and on the Put-style (send side) commands such as (Amazon S3) Put Object, (CDMI) Create Data Object and OpenStack Swift) Create/Update Object when Content-Encoding is enabled and the payload is > 1Mb.
- Header and Body Parsing, MD5 encoding (receive commands) are supported on all Amazon S3, Keystone Identity Service, CDMI and OpenStack Swift commands.
- Dynamic Content is supported only on the Put-style commands such as (Amazon S3) Put Object, (CDMI) Create Data Object and OpenStack Swift) Create/Update Object
- Authentication support (AWS2, AWS4 only) Amazon S3, (all except AWS4) CDMI and (Keystone Identity Service v2/v3, Tempauth [OpenStack Swift Authentication Action]) OpenStack Swift.

Data Verification

When more than one HTTP-based protocol is used in a Project and Data Verification is used, there will be a statistics chart for each protocol that does Data Verification but only a single Data Verification failures file (.csv) generated. This Data Verification failures file contains the Data Verification failures for all of the HTTP-based protocols. Failure information will consist of the "Path" (URI, Bucket/Object,

Container/Object), expected values (Expected Byte), returned values (Invalid Byte), time offset in microseconds, etc.

For more details on Data Verification see [Advanced Concepts: Data File Systems & Data Verification](#).

Threads and Asynchronous I/O (Pipelining)

HTTP Storage protocol Actions are supported in Threads and Async Operations. See [Advanced Concepts: Threads and Async Operations](#) for more details.

OpenStack Cinder

Cinder provides persistent block level storage for OpenStack compute engines which makes it useful for Database applications, raw block storage or extensible filesystems. The example Scenario below shows the use of the Cinder Volume-related Actions.

OpenStack Keystone Identity Service

The OpenStack Keystone Identity Service Action Authentication, allows the Tester to authenticate a Scenario with a Keystone Identity Service. Using the Keystone Identity Service, the Tester can specify either JSON or XML (Request Format) as well as Tenant, User and Password. The Keystone **Authentication** Action supports the use of FQDN. See the Keystone **Authentication** Action v2 below.

OR Keystone Authentication Action v3

Subsequent OpenStack Swift Actions must be configured to use Keystone as the Authentication method.

Amazon S3 Delete Multiple Objects

The Load Dynamix S3 software includes support for Amazon S3 **Delete Multiple Objects** Action which can be used to delete multiple Objects in a single Action. 2 - 5 Tester-specified Objects can be deleted per use of the **Delete Multiple Objects** Action.

Amazon S3 Multipart Upload

The Multipart upload enables the Tester to upload large objects in parts. This mechanism can be used to upload new large objects or make a copy of an existing object. Multipart uploading is a three-step process:

- Initiate the upload,
- Upload the object parts,
- Complete the multipart upload.

Upon receiving the complete multipart upload request, Amazon S3 constructs the object from the uploaded parts, and you can then access the object just as you would any other object in your bucket.

Amazon S3 AWS2/AWS4 Authentication

Amazon S3 Actions can be executed with no authentication. Amazon S3 Actions can also be authenticated using the AWS2/AWS4 mechanism. However, when Amazon S3 Post Object is authenticated it will be authenticated using AWS2 or AWS4 Authentication Policies. Use of AWS4 Authentication requires that the body of messages are signed. When the signed body of a **PUT** Action is greater than 1MB, the Load Dynamix S3 software forces the use of the S3 chunking mechanism (aws-chunk) to deliver the body of the PUT in chunks.

The following Policy discussion applies only to the Amazon S3 Post Object Action.

The Amazon S3 Actions support the use of either Amazon S3 AWS2 or AWS4 authentication Policies. AWS2 or AWS4 authentication use is enabled by setting the input field Form Data > Policy field == TRUE. Once this field is == TRUE the Tester can select either AWS2 or AWS 4 policy to use. AWS2 and AWS4 Authentication Policies vary only by the Signature used (AWS2 or AWS4) and when AWS4 is selected the Tester must provide the AWS Region (default == US Standard).

An Amazon S3 Authentication Policy enables the Tester to set and reference Access Key and Secret Access Key for all Amazon S3 Actions and for the Amazon S3 Post Object Action, the Tester can define when the Policy will expire and the Policy conditions (Match Exact, Match Any, or Custom Match that allows the Tester to define a custom set of conditions to match). Whereas with Amazon S3 Form Data > Policy == False, the Amazon S3 Actions have limited security mechanisms.

See the example Scenarios below that demonstrate the use of AWS2 and AWS4 Policies.

Amazon S3 AWS2

#	Protocol	Name
1	HTTP	Open HTTP Connection
2	Amazon S3	GET Service
3	Amazon S3	PUT Bucket
4	Amazon S3	PUT Bucket ACL
5	Amazon S3	GET Bucket ACL
6	Amazon S3	HEAD Bucket
7	Amazon S3	GET Bucket
8	Amazon S3	PUT Object
9	Amazon S3	PUT Object ACL
10	Amazon S3	HEAD Object
11	Amazon S3	GET Object ACL
12	Amazon S3	POST Object
13	Amazon S3	PUT Object - Copy
14	#	Gets the object added by "PUT Object" command.
15	Amazon S3	GET Object

Policy Settings:

Name	Value
Input	
Connection Handle	Default
Request HTTP version	1.1
Bucket Name	=@STRING(bucket_) + @SCENARIO COUNTER()
Method for accessing buckets	path-style
Request Headers	
Form Data	
> Standard Fields	
> Amazon Fields	
Policy	True
Signature Version	AWS2
Access Key ID	=@UP(0,A)
Secret Access Key	=@UP(0,B)
Policy expiration	2030-01-01T12:00:00.000Z
Policy conditions	Match Exact

Amazon S3 AWS4

#	Protocol	Name
1	HTTP	Open HTTP Connection
2	Amazon S3	GET Service
3	Amazon S3	PUT Bucket
4	Amazon S3	PUT Bucket ACL
5	Amazon S3	GET Bucket ACL
6	Amazon S3	HEAD Bucket
7	Amazon S3	GET Bucket
8	Amazon S3	PUT Object
9	Amazon S3	PUT Object ACL
10	Amazon S3	HEAD Object
11	Amazon S3	GET Object ACL
12	Amazon S3	POST Object
13	Amazon S3	PUT Object - Copy
14	#	Gets the object added by "PUT Object" command.
15	Amazon S3	GET Object
16	#	Gets the object added by "POST Object" command.

Policy Settings:

Name	Value
Input	
Connection Handle	Default
Request HTTP version	1.1
Bucket Name	=@STRING(bucket_) + @SCENARIO COUNTER()
Method for accessing buckets	path-style
Request Headers	
Form Data	
> Standard Fields	
> Amazon Fields	
Policy	True
Signature Version	AWS4
AWS Region	us-east-1 - US Standard
Access Key ID	=@UP(0,A)
Secret Access Key	=@UP(0,B)
Policy expiration	2030-01-01T12:00:00.000Z
Policy conditions	Match Exact

S3 Put Object and Post Object Comparison

Put Object	Post Object
Object content is sent in the full request body	Request body is always sent as multipart/form data Content of the Object is sent in the last part of the multipart request body The Policy is sent in a separate part of the message
When AWS2/AWS4 is enabled, the signature is calculated against the full body	When AWS2/AWS4 is enabled, the signature is calculated against the policy

When AWS2/AWS4 is enabled, the signature is sent in the request headers	When AWS2/AWS4 is enabled, the signature is sent in separate part of the multipart body
When AWS2/AWS4 is enabled, aws-chunked Content-Encoding can be applied (is automatically applied when content size is greater than 1MB)	aws-chunked can never be applied
Chunked Transfer-Encoding can only be applied when AWS2/AWS4 is disabled or aws-chunked is applied	aws-chunked Transfer-Encoding can always be applied
gzip or deflate Content-Encoding are applied to the full request body	gzip or deflate Content-Encoding would be applied to the last part of the multipart body (NOT currently implemented in Load DynamiX S3 support)

Amazon S3 Objects, Policies, Tagging and Cors Notes:

Amazon S3 **Delete Multiple Objects**, **Put Bucket Policy**, **Put Bucket Cors**, **Put Bucket Tagging** Actions have a Tester-defined number (up to 5) of Objects to Delete or Policies (Statements) or Cors or Tags that will be provided when the Action is executed. The TDE limits the number of these items to a maximum of 5. Additionally, in the **Put Bucket Policy** Action, each Statement can contain Tester-defined Resources (up to 5).

Amazon S3 Hostname Notes:

Load DynamiX Amazon S3 Actions contain an input field named "Method of accessing buckets" which determines how the path to the Bucket is constructed and has two options: "virtual hosted-style" and "path-style". When "virtual hosted-style" is selected and the **Open HTTP Connection** Action in a Load DynamiX Amazon S3 Scenario contains an FQDN, that name must be a Virtual Endpoint.

Name	Value
Input	
Connection Handle	Default
Request HTTP version	1.1
Bucket Name	=@UP{0,E}
Object Name	object2.test
Method for accessing buckets	virtual hosted-style
Endpoint	virtual hosted-style path-style
Endpoint Override Rule	

The Amazon S3 "Endpoint" input field appears when "Virtual hosted-style" is selected. The "Endpoint override rule" is used if a "Host" header cannot be constructed using the contents of the **Open HTTP/S Connection** Action Address field. If "Endpoint override rule" is set to "Always", the "Host" header is constructed based on the "Endpoint" input field, even if the **Open HTTP/S Connection** Action Address input field contains a valid FQDN. If the "Endpoint override rule" is set to "If Unavailable from Open Connection", the "Endpoint" input field will be used only if there is no FQDN in the Address input field of the **Open HTTP/S Connection** Action.

Name	Value
Input	
Connection Handle	Default
Request HTTP version	1.1
Bucket Name	=@UP{0,E}
Object Name	object2.test
Method for accessing buckets	virtual hosted-style
Endpoint	s3.amazonaws.com
Endpoint Override Rule	Always
Request Headers	If Unavailable from Open Connection
Authentication	Always

If a "Host" header is defined in the Request Headers list, two "Host" headers will be sent: the one defined by the Tester and one that is constructed dynamically using the rules described above in this section.

Amazon S3 Sample Project

CDMI Request Body

The CDMI Request Body input field is present in the CDMI Actions Create/Update {Container, Object, Domain and Queue}. The Request Body contains the Value to be stored in the entity being Created or Updated.

CDMI, OpenStack Swift/Cinder and Amazon S3 Encapsulation

CDMI and OpenStack Swift Actions are represented in the toolbox and graphs and client log files as a separate command set but these commands are delivered to CDMI and OpenStack Swift enabled servers encapsulated in the HTTP protocol. The following CDMI scenario generates HTTP statistics (Actions, Commands, Details, Response Time, Data Verification, Authentication, Authentication Time, Transfer-Encoding, Content-Encoding, and Throughput) Graphs and client log file data as well as CDMI statistics (Actions, Commands, Details, Response Time, Data Verification and Throughput) Graphs. This statistics behavior is true for all HTTP-based HTTP Storage protocols (they generate HTTP and HTTP Storage-specific statistics).

CDMI Client.client_scenario

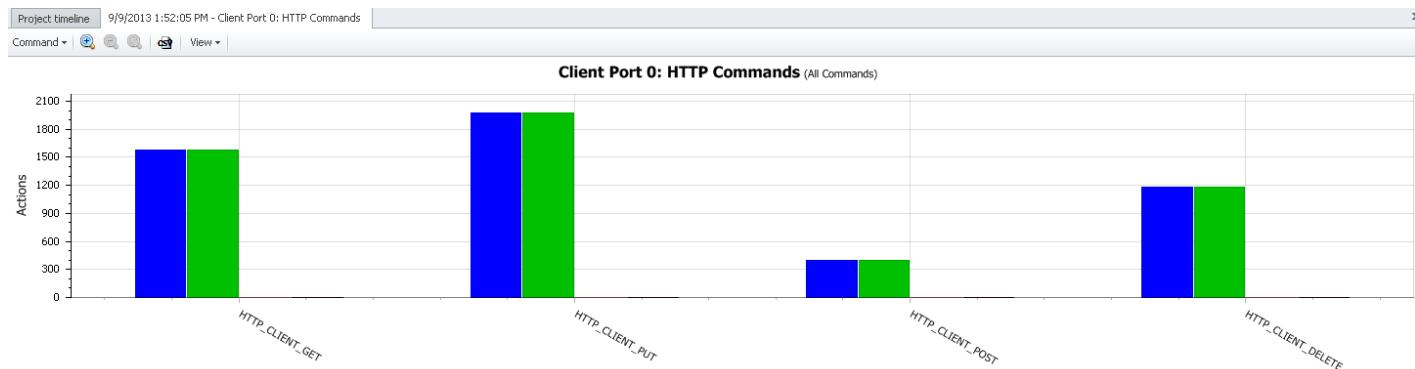
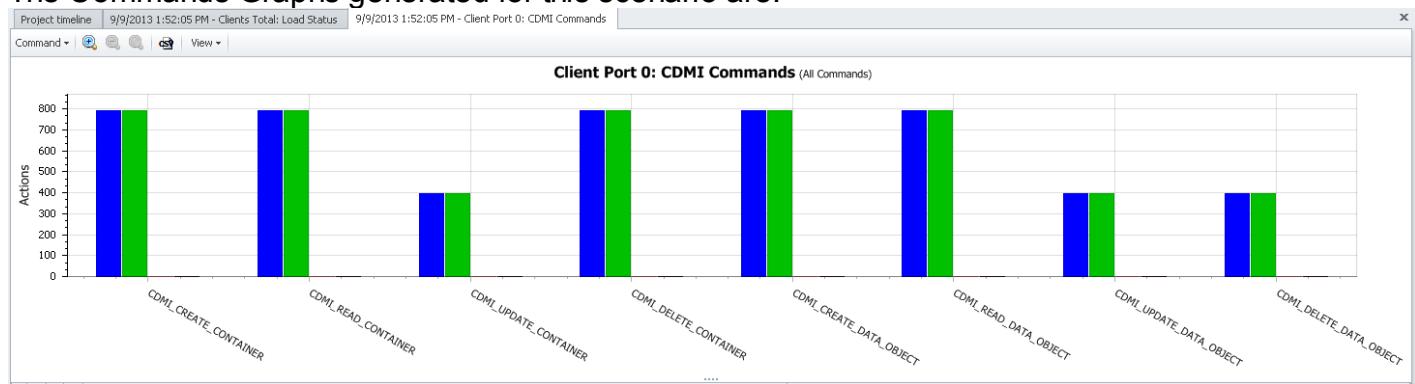
#	Protocol	Name
1	HTTP	Open HTTP Connection
2	#	<i>Container Operations</i>
3	CDMI	Create Container
4	CDMI	Read Capabilities Object
5	CDMI	Create Container
6	CDMI	Read Container
7	CDMI	Update Container
8	CDMI	Read Container
9	#	<i>Data Object Operations</i>
10	CDMI	Create Data Object
11	CDMI	Read Data Object
12	CDMI	Update Data Object
13	CDMI	Read Data Object
14	#	<i>Deleting all objects/Cleanup</i>
15	CDMI	Delete Data Object
16	CDMI	Delete Container
17	CDMI	Delete Container
18	HTTP	Close HTTP Connection

Properties

Name	Value
Input	
Connection Handle	Default
Request URI	= @STRING(/cdmi-server/123_) + @SCENARIOCOUNT() + @STRING(/)
Request Headers	
CDMI Request Headers	
Content-Type	application/cdmi-container
X-CDMI-Specification-Version	1.0.2
Accept	application/cdmi-container
Transfer Encoding	None
Authentication	
Preemptive Authorization	None
> Passive Authorization	
Credentials	
Username	
Password	
CDMI Request Body	
metadata	{}
domainURI	
exports	
deserialize	
copy	
move	
reference	
deserializemode	
Include Content-MDS	False
Output	
Extract Headers	
Extract Body	

OK Cancel

The Commands Graphs generated for this scenario are:



And the PCAP looks like

No.	Time	Source	Destination	Protocol	Length	Info
12	2.001903	172.16.160.0	172.16.0.8	HTTP	306	PUT /cdmi-server/123_1/ HTTP/1.1 (application/cdmi-container)
19	2.001945	172.16.160.2	172.16.0.8	HTTP	306	PUT /cdmi-server/123_3/ HTTP/1.1 (application/cdmi-container)
20	2.001962	172.16.160.4	172.16.0.8	HTTP	306	PUT /cdmi-server/123_5/ HTTP/1.1 (application/cdmi-container)
26	2.002904	172.16.160.1	172.16.0.8	HTTP	306	PUT /cdmi-server/123_2/ HTTP/1.1 (application/cdmi-container)
27	2.002920	172.16.160.3	172.16.0.8	HTTP	306	PUT /cdmi-server/123_4/ HTTP/1.1 (application/cdmi-container)
30	2.007056	172.16.0.8	172.16.160.0	HTTP	785	HTTP/1.1 201 created (application/cdmi-container)
31	2.007110	172.16.160.0	172.16.0.8	HTTP	310	PUT /cdmi-server/123_1/456/ HTTP/1.1 (application/cdmi-container)
32	2.010969	172.16.0.8	172.16.160.4	HTTP	785	HTTP/1.1 201 created (application/cdmi-container)
33	2.010980	172.16.160.4	172.16.0.8	HTTP	310	PUT /cdmi-server/123_5/456/ HTTP/1.1 (application/cdmi-container)
34	2.015063	172.16.0.8	172.16.160.4	HTTP	795	HTTP/1.1 201 created (application/cdmi-container)
35	2.015080	172.16.160.4	172.16.0.8	HTTP	187	GET /cdmi-server/123_5/456/ HTTP/1.1
36	2.018430	172.16.0.8	172.16.160.3	HTTP	785	HTTP/1.1 201 created (application/cdmi-container)
37	2.018441	172.16.160.3	172.16.0.8	HTTP	310	PUT /cdmi-server/123_4/456/ HTTP/1.1 (application/cdmi-container)
38	2.019810	172.16.0.8	172.16.160.1	HTTP	785	HTTP/1.1 201 created (application/cdmi-container)
39	2.019825	172.16.160.1	172.16.0.8	HTTP	310	PUT /cdmi-server/123_2/456/ HTTP/1.1 (application/cdmi-container)
40	2.023853	172.16.0.8	172.16.160.1	HTTP	795	HTTP/1.1 201 created (application/cdmi-container)
41	2.023862	172.16.160.1	172.16.0.8	HTTP	187	GET /cdmi-server/123_2/456/ HTTP/1.1
42	2.026442	172.16.0.8	172.16.160.1	HTTP	698	HTTP/1.1 200 OK (application/cdmi-container)
43	2.026459	172.16.160.1	172.16.0.8	HTTP	275	PUT /cdmi-server/123_2/456/ HTTP/1.1 (application/cdmi-container)
44	2.028660	172.16.0.8	172.16.160.4	HTTP	698	HTTP/1.1 200 OK (application/cdmi-container)
45	2.028670	172.16.160.4	172.16.0.8	HTTP	275	PUT /cdmi-server/123_5/456/ HTTP/1.1 (application/cdmi-container)
46	2.035059	172.16.0.8	172.16.160.2	HTTP	785	HTTP/1.1 201 created (application/cdmi-container)
47	2.035076	172.16.160.2	172.16.0.8	HTTP	310	PUT /cdmi-server/123_3/456/ HTTP/1.1 (application/cdmi-container)
50	2.043866	172.16.0.8	172.16.160.2	HTTP	795	HTTP/1.1 201 created (application/cdmi-container)
51	2.043876	172.16.160.2	172.16.0.8	HTTP	187	GET /cdmi-server/123_3/456/ HTTP/1.1

OpenStack Swift scenario execution generates:

HTTP statistics (Actions, Commands, Details, Response Time, Data Verification, Authentications, Authentications Time, Transfer-Encoding, Content-Encoding, and Throughput) Graphs.

OpenStack Swift statistics (Actions, Commands, Details, Response Time, Data Verification and Throughput) Graphs.

Amazon S3 scenario execution generates

HTTP statistics (Actions, Commands, Details, Response Time, Data Verification, Authentications, Authentications Time, Transfer-Encoding, Content-Encoding, and Throughput) Graphs.

Amazon S3 statistics (Actions, Commands, Details, Response Time, Data Verification and Throughput) Graphs.

CDMI scenario execution generates

HTTP statistics (Actions, Commands, Details, Response Time, Data Verification, Authentications, Authentications Time, Transfer-Encoding, Content-Encoding, and Throughput) Graphs.

CDMI statistics (Actions, Commands, Details, Response Time, Data Verification and Throughput) Graphs.

OpenStack Cinder scenario execution generates

HTTP statistics (Actions, Commands, Details, Response Time, Data Verification, Authentications, Authentications Time, Transfer-Encoding, Content-Encoding, and Throughput) Graphs.

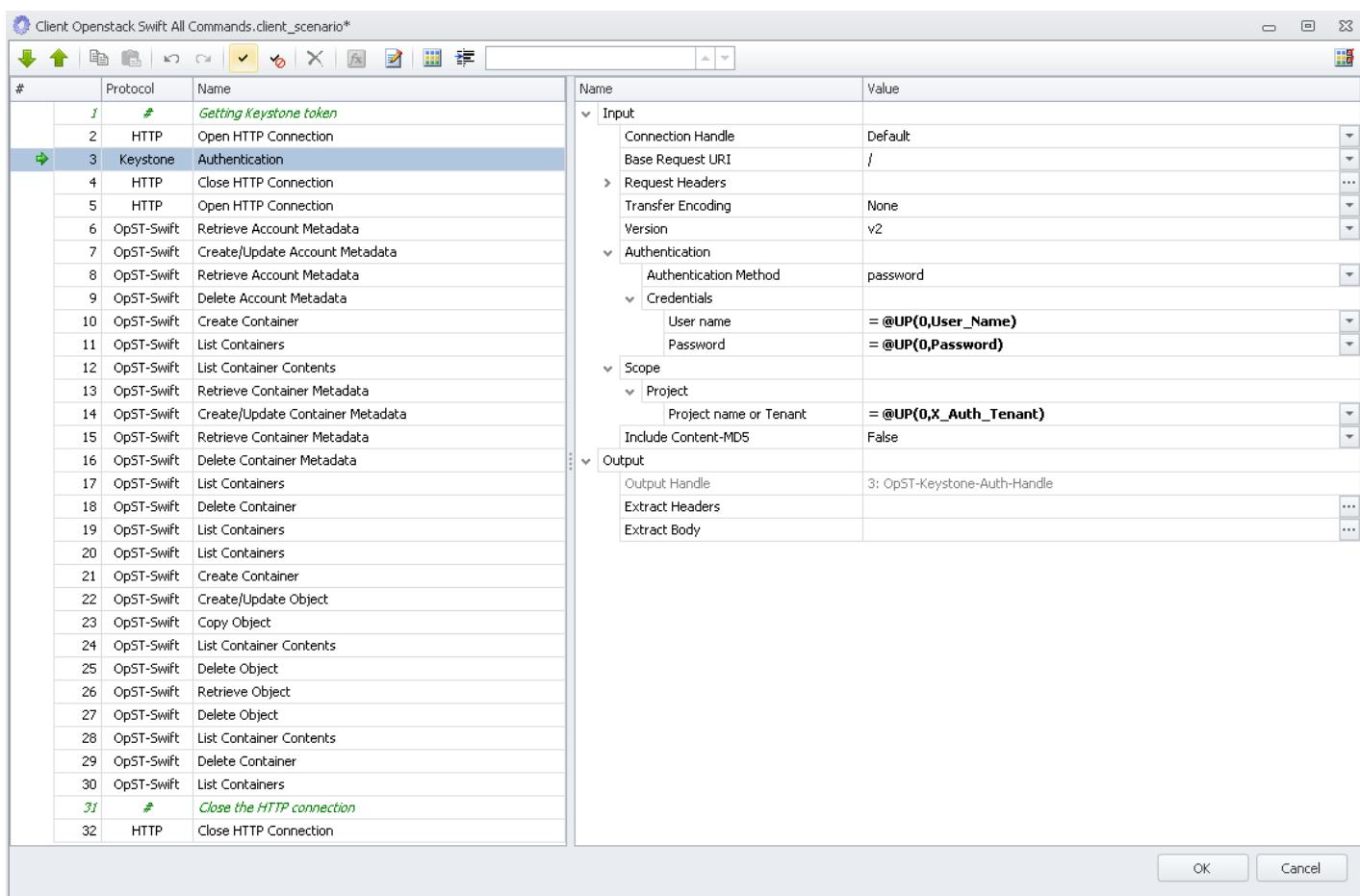
OpenStack Cinder statistics (Actions, Commands, Details, Response Time, Data Verification and Throughput) Graphs.

See the [Advanced Concepts: Test Execution Rules](#) for a complete list of the statistics generated when HTTP Storage Actions are used in a Scenario.

CDMI Sample Project

OpenStack Cinder Sample Project

OpenStack Swift Sample Project



HTTP Storage Caveats/Notes:

- Amazon S3 protocol supports a subset of the REST API, no SOAP API support provided.
- If the HTTP Open Connection Action in a Load DynamiX Amazon S3 Scenario contains an FQDN, that name must be a Virtual Endpoint if the "virtual hosted style" is selected in the Amazon S3 Actions.
- Amazon S3 Headers do not support the "Override Host header" option in the Header Editor.
- Amazon S3 Actions do support the "Endpoint Override rule" that allows the Amazon S3 Scenario to change the behavior of the construction of the Host Name.
- Cinder: Use of the Retry Conditions > Retry On Status input "deleting" in the Volume Show, Backup Show, Snapshot Show Actions will cause the Scenario to fail because once the Volume, Backup or Snapshot is deleted, the error code returned by the service is a 404 "Not Found" which will cause the Action to Fail.
- CDMI: Create/Update Data Object
 - Entity Data section exists not only for Non-CDMI Content-Type (Content-Type != application/cdmi-object) but also for CDMI Content-Type (Content-Type == application/cdmi-object)
 - The value transfer encoding combobox with values utf8/base64 is applied to Entity Data when CDMI Content-Type is chosen.
- Cinder: Quota Update for User is implemented using a PUT method instead of the POST method called for in the specification.
- Cinder: Quota Update Action, Quotas Set Parameters are not supported by the native Cinder implementation:
 - cores
 - fixed_ips
 - floating_ips
 - injected file content bytes

- injected_file_path_bytes
- injected_files
- instances
- key_pairs
- metadata_items
- ram
- security_group_rules
- security_groups
- Cinder: The information provided in the Ref field of the Volume Manage Action is target HTTP Storage provider dependent.
 - Volume Manage supports only those drivers that support the following driver-specific ref format:

```
ref: {
    "source-name": "name",
    "source-id": "id"
}
```
- HTTP Storage Pipelining (Async Operations) and Threads
 - See [Advanced Concepts: Threads and Async Operations](#) for details.
 - The following two HTTP/HTTPS commands are not supported in Async Blocks or Threads: **POST, CONNECT**.
 - The following five Amazon S3 Actions are not supported in Async Blocks or Threads: **Post Object, Delete Multiple Objects, Initiate Multipart Upload, Complete Multipart Upload and Upload Part**.
 - The following two CDMI Actions are not supported in Async Blocks or Threads: **Create Data Object and Create Queue Object**.
 - The following OpenStack Keystone Identity Service Action is not supported in Async Blocks: **Authentication**.
 - The following twelve OpenStack Cinder Actions are not supported in Async Blocks or Threads: **Volume List, Snapshot List, Backup List, Open HTTP Connection and Open HTTPS Connection Actions, Snapshot Show, Backup Show, Volume Create, Volume Type Create, Snapshot Create, QOS Create, Backup Create, Backup Restore and Volume Manage**.
 - The following five OpenStack Swift Actions are not supported in Async Blocks or Threads: **Create/Update Container Metadata, Delete Container Metadata, Update Object Metadata, Create/Update Account Metadata, Delete Account Metadata**.

Reference: End of Life Statement

Reference: End of Life Statement TDE Release v4.0 and earlier

June 9, 2016

To: All Load DynamiX Customers

Re: End of Life Announcement – TDE Software Release v4.0
Effective July 1, 2016.

To our valued customers,

Load DynamiX is announcing the end of life for TDE software release version v4.0 and all earlier releases effective June 9, 2016. This notice allows our customers a six month time frame to migrate to a current version of our software after which period, support will no longer be available for TDE v4.0 and earlier. We encourage customers to migrate as early as convenient to take advantage of the significant new features, capabilities, as well as fixes included in the current release. All customers who are covered under a maintenance agreement can call (+1 (408) 477-8946) or email our Technical Support staff Support@LoadDynamix.com for information on how to get a downloadable copy of the latest software. For customers who are not covered by a maintenance agreement, please contact your Load DynamiX sales representative at 1-877-652-5733 or via email at sales@LoadDynamix.com.

Regards,

Load DynamiX, Inc.
Technical Support
Email: Support@LoadDynamix.com
Phone: 1-408-477-8946

Index

:

::Compressible(Seed,Percentage), 187
::DataContent, 187
::Random, 187
::SeededRandom(Seed), 187
::Sequential, 187

@

@FORMULA, 326
@LOOPINDEX, 326
@LOOPTOTAL, 326
@RANDOM, 326
@RANDOM64(), 39, 326
@SCENARIOCOUNTER, 326
@STRING, 326
@TIME, 39
@UP, 158
@VARIABLE, 210

A

Aborted or Failed Actions, 123
Aborted or Failed Scenarios, 123
Aborting Scenarios, 123
Action Input Shorthand, 575
Advanced Load Profile, 81
Aliases, 158, 210, 326
ALUA, 487
ALUA Explicit management, 487
ALUA Implicit management, 487
Amazon S3, 577
ANY, 158
Appliance Configuration, 253
Appliance Dimensions, 16
Appliance Events, 253
Appliance Firmware Update, 123
Appliance Firmware Version, 123
Appliance URL, 253
Appliance utilization information, 253
Appliances Tab, 123
ARP, 81
Asynchronous Operations, 239
Auto Lookup, 303
AutoFill, 158
Automatic Offset, 187
Automation, 264
Automation on Linux, 264
AWS2, 577
AWS4, 577

B

BB Credits, 487
 Beta website, 4
 Break, 289
 Bulk Project Conversion, 264

C

CDB Action templates, 487
 CDMI, 577
 Chained Commands, 235
 Check Condition Custom extension, 487
 Check Condition Response Handling, 487
 CIFS-SMB, 389
 CIFS-SMB Statistics, 216
 Cinder, 577
 Circular Tacing Parameter buffer, 81
 Compile Errors, 245
 Compound Requests, 235
 Concurrency Control, 289
 Concurrent Threads, 239
 Connectivity Check, 16
 Continue, 289
 Copperating Scenarios, 289
 Create Handle Variable, 303
 Creating CIFS-SMB Directories, 389
 Custom Actions, 487
 Custom CDB Builder, 487
 Custom Graphs, 123
 Custom SCSI Actions, 487

D

Data Source Absolute, 487
 Data Source Relative, 487
 Data Verification, 187
 data verification completion modes, 187
 DCB Statistics, 216
 DCB/DCBX, 360
 Debugging, 245
 Decimal Number support, 575
 Decimal Shorthand, 575
 Default Handle, 81
 Delay, 289
 Delay Port, 39
 Direct Attach (DA) cables, 16
 Discard Protocol, 324
 Distribution, 289
 DNS, 81, 348
 DNS Statistics, 348
 duration, 4

E

Echo Protocol, 324
Else, 289
Else If, 289
Emergency License, 16
End If, 289
Events, 289
Export Project, 39

F

FC Port Info Generate UP, 487
FC/FCoE Response Time, 487
FCoE, 487
Fibre Channel, 487
Fibre Channel Connect timeout, 487
Fibre Channel Inactivity Timer, 487
Fibre Channel IO timeout, 487
Fibre Channel Port Queue Depth, 487
Fibre Channel Ports Info, 487
Fibre Channel Reconnect, 487
Fibre Channel Statistics, 216
Fibre Channel Tracing, 487
Formula, 326
FTP site, 4
Functions, 326

G

GB, 575
Generate Automation Files, 264
Global User Parameter files, 158
Graph Legend, 39

H

Handles, 81
Help version (56.m), 3
Hex editor, 81
Hexadecimal Input, 81
HTTP encapsulation, 577
HTTP Pipelining, 239, 577
HTTP Storage, 239, 577
HTTP Storage Statistics, 216
HTTP/HTTPS, 440
HTTP/S Statistics, 216
HW Setup, 16

I

If, 289
Import Automation Projects, 264
Import Project, 39
Information Required, 4
Installation and setup, 16
IO Manager, 289, 487
IP Address Calculation, 81, 253

IPv4, 81
IPv6, 81, 346
IPv6 Caveats, 346
IPv6 Statistics, 216
iSCSI, 472
iSCSI 2Way Chap, 472
iSCSI 2way chap authentication, 472
iSCSI Asynchronous Read/Write, 472
iSCSI Data Verification, 187
iSCSI Login Session Type, 472
iSCSI MPIO, 487
iSCSI Reconnect, 472, 487
iSCSI Redirect, 487
iSCSI Statistics, 216

J

Jumbo Frames, 199

K

KB, 575
Kerberos, 81, 439
Kerberos Statistics, 216
Keystone v2/v3, 577

L

LDX-V, 379
LDX-VLS, 379
Lease, 394
License Administration, 16
Licenses, 16
Linux, 264
Load Profile, 81
Local User Parameters, 158
Log Message, 289
Logical Port, 39
LUN acceptable values, 487

M

Management Station, 16
Max Open Connections, 338
Max Project Duration, 39, 81, 253
Max Test Duration, 4, 39, 81, 253
MAXIMUM, 326
Maximum Duration, 39, 81, 253
maximum project duration, 4
MB, 575
MINIMUM, 326
Mono, 264
MPIO, 487
MPIO Fail Over, 487
MPIO Least Queue Depth, 487
MPIO Round Robin, 487

MPIO Weighted Paths, 487

MSRPC, 394

MS-RPC, 394

MSS, 199

MTU, 199

Multi Charting, 123

My Resources, 81

N

NDP, 81

Network Profile, 81

NFS, 303, 425, 427, 432

NFS Onwer ID, 303

NFS Open File Confirm, 303

NFS Statistics, 216

NFS UID/GID, 303

NFSv3 Asynchronous Read/Write, 303, 427

NFSv3 Auto Lookup, 303

NFSv4 ACI, 303

NOR, 303, 427, 487

NPIV, 81, 487

NTLM Flags, 365

O

OpenStack Cinder, 577

OpenStack Swift, 577

Oplock, 394

Output log file, 123

Output Window Messages, 39

P

PCAP, 81, 487

Performance, 253

PERL, 264

Physical files, 187

Ping, 245

Port Queue Depth, 487

POWER, 326

Project Configuration, 123

Project Conversion, 39

Project Export, 39

Project flow, 389, 394, 427, 432, 440, 472, 487

Project Import, 39

Project Information, 4

Project log file, 123

Project Summary, 123

ProjectConfig.zip, 123

Protocol Statistics, 216

R

Random files, 187

Reconnect, 303, 472, 487

Report Wizard, 123
Resource Explorer, 39, 81
Response Handling, 204
Restore Test Configuration, 123
Results, 123

S

S3, 577
Sample Projects, 321, 389, 394, 427, 440, 472, 415, 487, 577
SampleProject Examples, 389, 394, 432, 472, 415, 440, 427, 487, 577
Scenario, 81
Scenario Editor, 39
SCSI CDB Action templates, 487
SCSI Check Condition Response Handling, 487
SCSI Per LUN Statistics, 487
SCSI Statistics, 216
Seeded random, 187
Sequential files, 187
SFP+, 16
Simple Reports, 123
SMB Keep Alive, 389
SMB2, 394
SMB2 Credit Charged and Credit Requested, 394
SMB2 Honor Credits, 394
SMB2 IOCTL, 394
SMB2 Keep Alive, 394
SMB2 Negotiate Capabilities, 394
SMB2 Statistics, 216
SMB3, 415
SMB3 Directory Leasing, 415
SMB3 IOCTL, 415
SMB3 Multi-Channel, 394, 415
SMB3 Persistent Handles, 415
SMB3 Server, 415
SMB3 Signing, 415
SSC Fixed Input, 487
Statistics, 216
Status Code Return, 289
Support email address, 16
Swift, 577
SwiftCmd, 264
SwiftCmd Syntax, 264
SwiftTest Resources, 81
Syncronizing Actions, 289
System Update, 4

T

TB, 575
TCL, 264
TDE Log files, 123
TDE Preferences, 39
TDE Software Version, 16

TDE Unhandled Exception log file, 4

Test Configuration, 123

Test Creation, 81

Test Duration, 39, 389

Test Execution, 123

Test Execution Rules, 216

test expired, 123

Thread Controls, 239

Threads, 239

Timeline, 81

Toolbox, 81

Tracing Parameters, 81

Transmission failed error, 487

Troubleshooting, 245

U

UDP, 348, 574

Unique Aliases, 158

UP Aliases, 158

Update Firmware, 123

Update Handle Variable, 303

UPL(), 158, 326

User Parameter Files, 158

UTF8, 81

V

VAAI, 472

Variables, 210

version, 303

Virtual Appliance, 379

Virtual Appliance License Server, 379

Virtual Appliance Licenses, 379

Virtual Circuit, 389

Virtual Tape Library, 487

VTL, 487

W

Windows Authentication of a DUT, 245

WWID, 487