

**TekExpress®**  
**D-PHY Automated Solution**  
**Online Help**





**TekExpress®**  
**D-PHY Automated Solution**  
**Online Help**

Copyright © Tektronix. All rights reserved. Licensed software products are owned by Tektronix or its subsidiaries or suppliers, and are protected by national copyright laws and international treaty provisions.

Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specifications and price change privileges reserved.

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

TekExpress is a registered trademark of Tektronix, Inc.

MIPI is a registered trademark of MIPI Alliance.

TekExpress D-PHY Automated Solution Online Help, 076-0227-00.

### **Contacting Tektronix**

Tektronix, Inc.  
14150 SW Karl Braun Drive  
P.O. Box 500  
Beaverton, OR 97077  
USA

For product information, sales, service, and technical support:

- In North America, call 1-800-833-9200.
- Worldwide, visit [www.tektronix.com](http://www.tektronix.com) to find contacts in your area.

# Table of Contents

## Introduction to the Application

Using Online Help.....	1
Related Documentation .....	1
Conventions .....	2
Technical Support .....	3

## Getting Started

Compatibility.....	5
Minimum System Requirements.....	5
Install the Software .....	6
Application Directories and Usage .....	7
File Name Extensions .....	8
Activate the License.....	8
Before You Click Run.....	10

## Operating Basics

D-PHY Application Overview .....	13
Starting the Application.....	13
Resizing the Application Window .....	14
Exiting the Application .....	15
Global Controls .....	15
Menus .....	15
File Menu .....	15
View Menu .....	16
Tools Menu .....	16
Help Menu.....	18

## How To

Select Test(s) .....	21
Configure and Run the Test(s).....	22
View and Select Connected Instruments .....	27
Use Prerecorded Waveforms for Analysis .....	29
Use Acquire Live Waveform for Analysis.....	31
View the Progress of Analysis .....	32
View the Report.....	34
View Test Related Files .....	35

## Application Examples

Connection Setup .....	38
Configuring Parameter Values .....	38
Group 1 Tests	
Testing Data Lane LP-TX Thevenin Output High Level Voltage (VOH)	
Connections for Data Lane LP-TX Thevenin Output High Level Voltage (VOH) .....	39
Configure and Run Data Lane LP-TX Thevenin Output High Level Voltage (VOH) .....	39
Testing Data Lane LP-TX Thevenin Output Low Level Voltage (VOL)	
Connections for Data Lane LP-TX Thevenin Output Low Level Voltage (VOL) .....	41
Configure and Run Data Lane LP-TX Thevenin Output Low Level Voltage (VOL) .....	41
Testing Data Lane Rise Time	
Connections for Data Lane Rise Time .....	42
Configure and Run Data Lane Rise Time .....	42
Testing Data Lane Fall Time	
Connections for Data Lane Fall Time .....	43
Configure and Run Data Lane Fall Time .....	43
View Report for Group 1 Tests .....	44
Group 2 Tests	
Testing Clock Lane LP-TX Thevenin Output High Level Voltage (VOH)	
Connections for Clock Lane LP-TX Thevenin Output High Level Voltage (VOH) .....	45
Configure and Run Clock Lane LP-TX Thevenin Output High Level Voltage (VOH) .....	45
Testing Clock Lane LP-TX Thevenin Output Low Level Voltage (VOL)	
Connections for Clock Lane LP-TX Thevenin Output Low Level Voltage (VOL) .....	47
Configure and Run Clock Lane LP-TX Thevenin Output Low Level Voltage (VOL) .....	47
Testing Clock Lane Rise Time	
Connections for Clock Lane Rise Time .....	48
Configure and Run Clock Lane Rise Time .....	48
Testing Clock Lane Fall Time	
Connections for Clock Lane Fall Time .....	49
Configure and Run Clock Lane Fall Time .....	49
View Report for Group 2 Tests .....	50
Group 3 Tests	
Testing Data Lane HS Entry: Data Lane TLPX Value	
Connections for Data Lane HS Entry: Data Lane TLPX Value .....	51
Configure and Run Data Lane HS Entry: Data Lane TLPX Value .....	51
Testing Data Lane HS Entry: THS-PREPARE Value	
Connections for Data Lane HS Entry: THS-PREPARE Value .....	52
Configure and Run Data Lane HS Entry: THS-PREPARE Value .....	52
Testing Data Lane HS Entry: THS-PREPARE + THS-ZERO Value	
Connections for Data Lane HS Entry: THS-PREPARE + THS-ZERO Value .....	54
Configure and Run Data Lane HS Entry: THS-PREPARE + THS-ZERO Value .....	54
View Report for Group 3 Tests .....	55

Group 5 Tests	
Testing HS Entry TCLK PREValue	
Connections for HS Entry TCLK PREValue .....	56
Configure and Run HS Entry TCLK PREValue .....	56
Testing HS Exit TCLK POSTValue	
Connections for HS Exit TCLK POSTValue .....	57
Configure and Run HS Entry TCLK POSTValue .....	57
Testing Clock Rising Edge Alignment to First Payload Bit	
Connections for Clock Rising Edge Alignment to First Payload Bit.....	58
Configure and Run Clock Rising Edge Alignment to First Payload Bit .....	58
View Report for Group 5 Tests.....	59

## TekExpress Programmatic Interface

About the Programmatic Interface.....	63
Server and Client Proxy Objects.....	64
Remote Proxy Object.....	64
Client Proxy Object.....	65
Client Programmatic Interface: An Example.....	66
D-PHY Application Command Arguments and Queries.....	69
Handle Error Codes .....	97
Program Example.....	97

## Reference

Algorithms	
Group 1 Tests.....	101
Test 1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH) .....	101
Test 1.1.2 Data Lane LP-TX Thevenin Output Low Level Voltage (VOL) .....	101
Test 1.1.3 Data Lane Rise Time .....	102
Test 1.1.4 Data Lane Fall Time.....	103
Group 2 Tests.....	104
Test 1.2.1 Clock Lane LP-TX Thevenin Output High Level Voltage (VOH).....	104
Test 1.2.2 Clock Lane LP-TX Thevenin Output Low Level Voltage (VOL).....	104
Test 1.2.3 Clock Lane Rise Time.....	105
Test 1.2.4 Clock Lane Fall Time .....	105
Group 3 Tests.....	106
Test 1.3.1 Data Lane HS Entry: Data Lane TLPX Value.....	106
Test 1.3.2 Data Lane HS Entry: THS-PREPARE Value.....	107
Test 1.3.3 Data Lane HS Entry: THS-PREPARE + THS-ZERO Value.....	108
Group 5 Tests.....	110
Test 1.5.1 HS Entry TCLK PREValue.....	110
Test 1.5.2 HS Exit TCLK POSTValue.....	111
Test 1.5.3 HS Clock Rising Edge Alignment to First Payload Bit .....	113

Shortcut Keys ..... 113

## Index



## Using Online Help

The D-PHY online help serves as a reference for using the D-PHY Automated Solution application.

**Contents.** The Contents tab displays books and pages that represent the categories of information in the online Help system. The Search tab enables you to search for keywords in the Help system and to locate topics containing those words.

**Back and Forward.** The Back and Forward buttons allow you to browse through topics.

**Index.** The Index tab displays a multi-level list of keywords and keyword phrases. These terms are associated with topics in the Help system and direct you to specific topics.

**Search.** When the search is completed, a list of topics is displayed so that you can select a specific topic to view.

**To print a topic.** Select the **Print** button from the Help Topics menu bar.

While using the online help, you can print topics and information from the HTML Help viewer.

**To print a single topic.**

1. Find the topic in the Contents tab.
2. Click **Print**.
3. Click **Print** the selected topic and click OK.

**To print all topics in a selected TOC book.**

1. Find the TOC book in the Contents tab.
2. Click **Print**.
3. Click Print the selected heading and all subtopics and click **OK**.

---

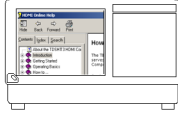


**NOTE.** *If topics include expanding or drop-down hot spot, click the hotspots to display the information before you print. Hotspot is a clickable text that displays additional information below the link. Click on the hotspot text to show or hide the hotspot content.*

---

## Related Documentation

The following manuals are available as part of the TekExpress D-PHY Automated Solution documentation set.

**Table 1: Product documentation**

Item	Purpose	Location
Online Help	In-depth operation and UI help.	
PDF of the Online Help (077-0514-XX)	In-depth operation and UI help.	  <a href="http://www.Tektronix.com">www.Tektronix.com</a>

## Conventions

Online Help uses the following conventions:

- The term “DUT” is an abbreviation for Device Under Test.
- The term “select” is a generic term that applies to the two mechanical methods of choosing an option: using a mouse or using the touch screen.
- A Note identifies important information.

## Technical Support

Tektronix values your feedback on our products. To help us serve you better, please send us your suggestions, ideas, or comments on your application or oscilloscope.

When you contact Tektronix Technical Support, please include the following information (be as specific as possible):

### General Information

- All instrument model numbers.
- Hardware options, if any.
- Probes used.
- Your name, company, mailing address, phone number, FAX number.
- Please indicate if you would like to be contacted by Tektronix about your suggestion or comments.

### Application Specific Information

- Software version number.
- Description of the problem such that technical support can duplicate the problem.
- If possible, save the setup files for all the instruments used and the application.
- If possible, save the TekExpress setup files, log.xml , \*.TekX (session files and folders), and status messages text file.
- If possible, save the waveform on which you are performing the measurement as a .wfm file.

Forward the information to technical support using one of these methods:

- E-mail – [techsupport@tektronix.com](mailto:techsupport@tektronix.com)
- FAX – (503) 627-5695



## Compatibility

The TekExpress D-PHY application runs on the following Tektronix oscilloscopes:

- DPO7254 <sup>1</sup> and DPO7354 Digital Oscilloscope
- DPO/DSA70404, DPO/DSA70604, DPO/DSA70804, DPO/DSA71254, DPO/DSA71604, and DPO/DSA72004 Series Digital Oscilloscopes
- DPO/DSA70404B, DPO/DSA70604B, DPO/DSA70804B, DPO/DSA71254B, DPO/DSA71604B, and DPO/DSA72004B Series Digital Oscilloscopes
- MSO70404, MSO70604, MSO70804, MSO71254, MSO71604, and MSO72004 Series Mixed Signal Oscilloscopes

<sup>1</sup> The DPO7254 oscilloscope may not meet the Rise/Fall time specifications of D-PHY standards. If that is not critical, then you can use the DPO7254 oscilloscope for the D-PHY measurement.

The TekExpress D-PHY application supports the following Tektronix probes:

- TAP1500
- P7240
- P6245
- P7380A

## Minimum System Requirements

The minimum system requirements for a PC to run TekExpress are as follows:

**Table 2: System requirements**

<b>Processor</b>	Pentium 4/M or equivalent processor.
<b>Operating System</b>	Windows XP Service Pack 2.
<b>Memory</b>	512 MB of memory.
<b>Hard Disk</b>	Approximately 2 GB of available hard-disk space for the recommended installation, which includes full TekExpress installation and distributed components.
<b>Drive</b>	DVD drive.
<b>Display</b>	Super VGA resolution or higher video adapter (800x600 minimum video resolution for small fonts or 1024x768 minimum video resolution for large fonts). The application is best viewed at 96 dpi display settings <sup>1</sup> .

**Table 2: System requirements (cont.)**

<b>Software</b>	<ul style="list-style-type: none"> <li>■ Microsoft Excel 2002 or above.</li> <li>■ Microsoft Internet Explorer 6.0 SP1 or later.</li> <li>■ Microsoft Photo Editor 3.0 or equivalent software for viewing image files.</li> <li>■ Adobe Reader 6.0 or equivalent software for viewing portable document format (PDF) files.</li> </ul>
<b>Other Devices</b>	<ul style="list-style-type: none"> <li>■ Microsoft compatible mouse or compatible pointing device.</li> <li>■ Four USB ports (two USB ports minimum).</li> <li>■ PCI-GPIB or equivalent interface for instrument connectivity<sup>2</sup>.</li> </ul>

<sup>1</sup> If TekExpress is running on an instrument having a video resolution lower than 800x600 (for example, sampling oscilloscope), it is recommended to connect a secondary monitor and this must be enabled before launching the application.

<sup>2</sup> If TekExpress is installed on a Tektronix oscilloscope, the virtual GPIB port will be used by TekExpress for communicating with oscilloscope applications. If external GPIB communication devices like USB-GPIB-HS or equivalent are used for instrument connectivity, ensure that the Talker Listener utility is enabled in the DPO/DSA oscilloscope's GPIB menu. For ease of use, connect to an external (secondary) monitor.

## Install the Software

1. Close all applications.
2. Download D-PHYWebinstaller\_<version>.EXE.
3. Double-click the executable to extract the files. After extraction, the installer launches and the software is installed in C:\Program Files\Tektronix\TekExpress\TekExpress D-PHY.

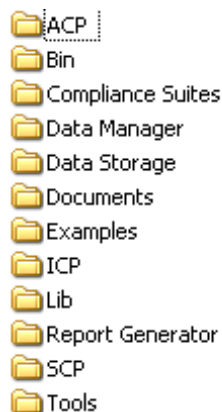
---

**NOTE.** *If you are updating the existing TekExpress D-PHY application with a newer version, make sure that you manually delete the folder "MIPI\_DPHY\_mcr" under C:\Program Files\Tektronix\MeasurementLibs\D-PHY\Algorithms. If you do not do this, some of the measurements may not run properly.*

---

## Application Directories and Usage

The application directory and associated files are organized as follows:



The following table lists the default directory names and their usage:

**Table 3: Application directories and usage**

Directory names	Usage
InstallDir\TekExpress\TekExpress D-PHY	Contains the application and associated files.
TekExpress D-PHY\Compliance Suites	Contains compliance specific files.
TekExpress D-PHY\Bin	Includes the Miscellaneous libraries of the D-PHY application.
TekExpress D-PHY\Lib	Includes utility files specific to the D-PHY application.
TekExpress D-PHY\Tools	Includes instrument application specific files for the D-PHY application.
TekExpress D-PHY\ACP	Includes instrument and application specific interface libraries of the D-PHY application.
TekExpress D-PHY\SCP	
TekExpress D-PHY\ICP	
TekExpress D-PHY\Documents	Includes the technical documentation for the application.
TekExpress D-PHY\Data Manager	Includes the result management specific libraries of the D-PHY application are present in these folders.
TekExpress D-PHY\Data Storage	
TekExpress D-PHY\Report Generator	

## File Name Extensions

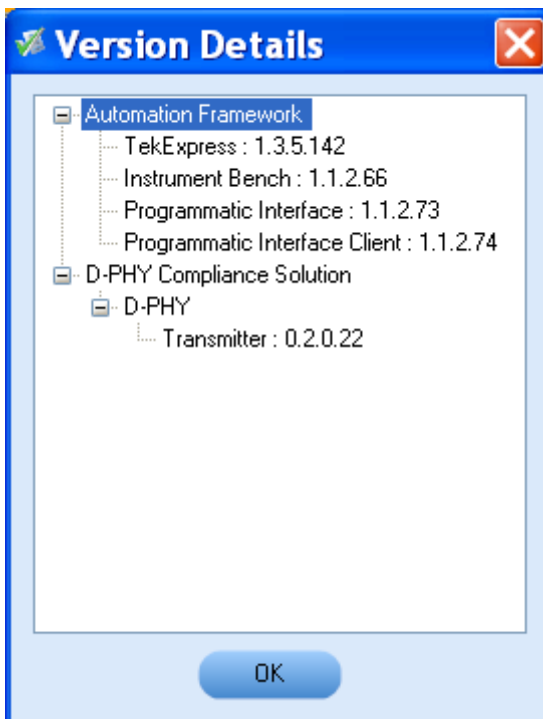
The software uses the following file name extensions:

File name extension	Description
.TekX	The session file will be saved in this format.
.seq	The test sequence file.
.xml	The encrypted XML file that contains the test specific configuration information. The log file extension is also xml.
.wfm	The test waveform file.
.mht	The test result report will be saved in this format.

## Activate the License

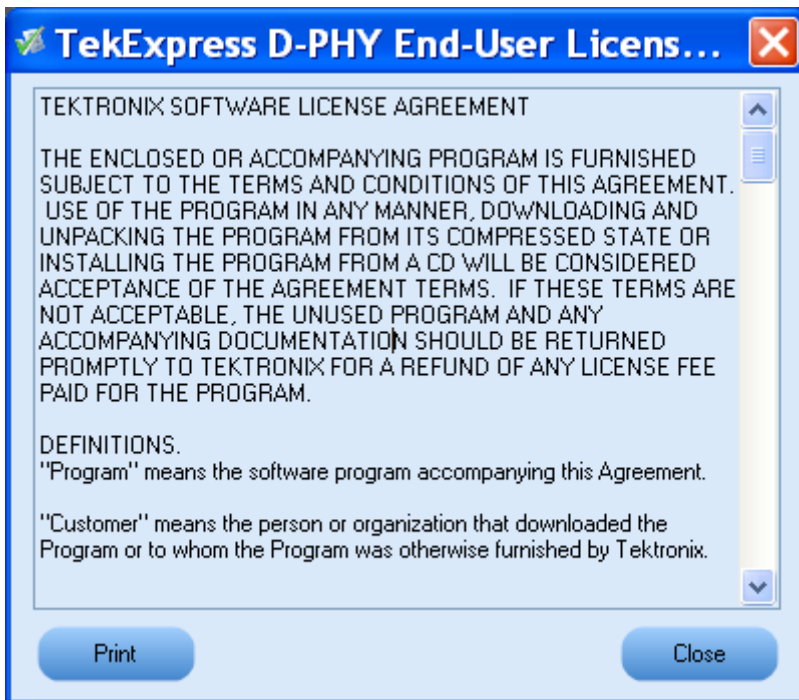
Follow the steps below to activate the license:

1. Click **Help > Activate License** or **Help > About TekExpress > License Info** to view the license information and activate the application.
2. Click the **View Version Details** link to check the version numbers of the installed test suites.

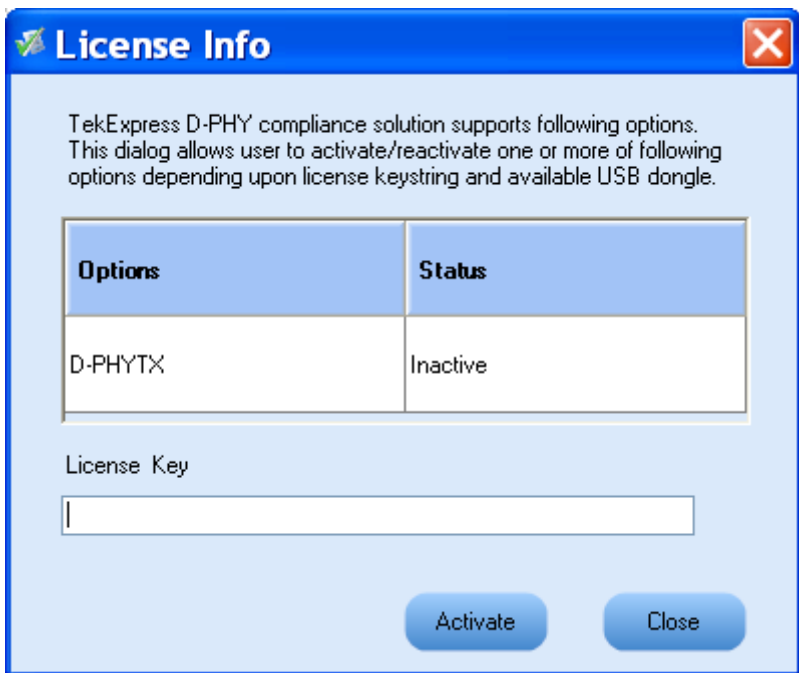




3. Click the **View End-User License Agreement** link to open the following Tektronix Software License Agreement window. Click **Print** to print the License Agreement.



4. Click **License Info** to view the available software options. This window shows the license key and supported options with their status (active or inactive) with the current license key.



5. If you are activating the license for the first time, the license key field will be empty. To activate the license, connect the USB dongle to your computer, enter the license key provided in the license key certificate, and click **Activate**. If the activation is successful, a sign is displayed next to the license key field.
6. If you are reactivating the license, click **Reactivate**, enter the new license key and click **Activate**.

## Before You Click Run

After you install D-PHY, it creates the following folders on your computer:

- \Program Files\Tektronix\TekExpress\TekExpress D-PHY.

---

**NOTE.** *Ensure that the “TekExpress” folder has read and write access.*

---

- \My Documents\My TekExpress\D-PHY.
- \My Documents\My TekExpress\D-PHY\Untitled Session.

Every time the TekExpress D-PHY.exe is launched, an Untitled Session folder is created under D-PHY folder. The Untitled Session folder is deleted when you exit the D-PHY application.



---

**CAUTION.** *Each session has multiple files associated with it. Do not modify any of the session files and/or folders as this may result in loss of data or corrupted session files.*

---

- The My TekExpress folder is created as a shared folder with share name as <domain><user ID>My TekExpress (or if the PC is not connected to domain, then share name is <Computer name><user ID> My TekExpress).
- The above shared folder is mapped as X: (X drive) on to the instrument where D-PHY is running.

---

**NOTE.** *If X drive is mapped to any other shared folder, the application will display a warning message window asking you to disconnect the X: drive manually.*

---

Do the following before you click Run:

---

**NOTE.** *Ensure that the network connectivity is enabled on the instrument running the D-PHY application.*

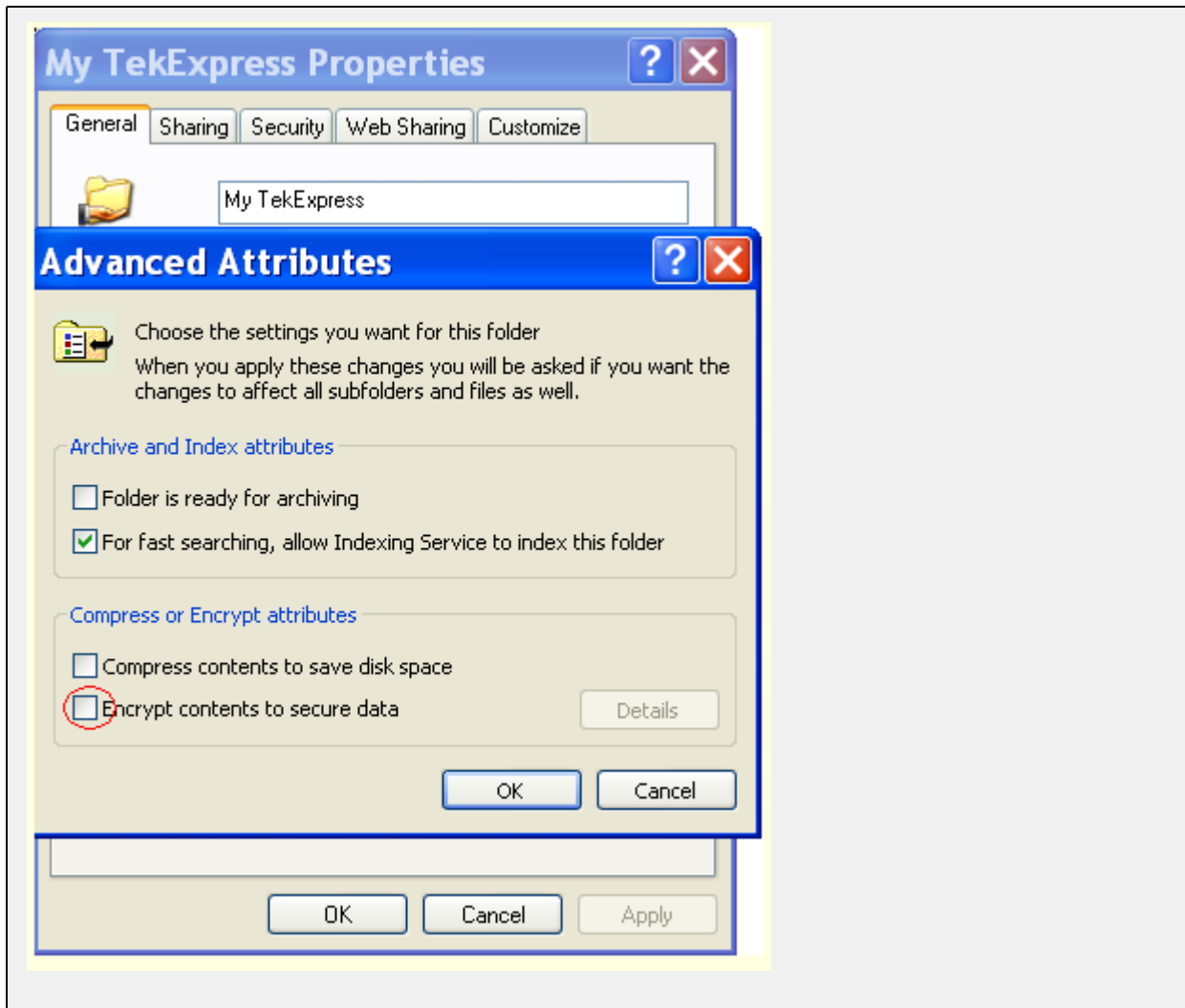
---

1. [Map \(see page 11\)](#) the shared My TekExpress folder as X: (X drive) on all the instruments used in test setup running Microsoft Windows Operating System. This shared folder is used to save the waveform files or used during any other file transfer operations.
2. Right-click on the My TekExpress folder and open the **Properties** dialog box. Select the **General** tab and then **Advanced**. In the **Advanced Attributes** window, ensure that the option **Encrypt contents to secure data** is NOT selected. Click [here \(see page 12\)](#) to view the picture.
3. Ensure that all the required instruments are properly warmed up, [Signal Path Compensation \(SPC\) \(see page 12\)](#) performed, followed by cable deskew.

### Mapping My TekExpress Folder

To map the My TekExpress folder on the instruments, follow the steps below:

1. Open Windows Explorer.
2. From the Windows Explorer menu, select **Tools > Map Network drive**.
3. Select the Drive letter as X: (if there is any previous connection on X:, disconnect it first through **Tools > Disconnect Network drive** menu of Windows Explorer).
4. In the Folder field, enter the remote My TekExpress folder path (for example, \\192.158.97.65\John's My TekExpress)
5. To determine the IP address of the PC where "My TekExpress" folder exists, do the following:
  - Select **Start > Run** menu on the PC where the My TekExpress folder exists.
  - Enter cmd and press **Enter**.
  - At the command prompt, type ipconfig.



To find SPC, do the following:

1. On the oscilloscope main menu, select the **Utilities** menu.
2. Click the **Instrument Calibration** option.

## D-PHY Application Overview

TekExpress is the Tektronix Test Automation Framework, developed to support your current and future test automation needs. TekExpress uses a highly modular architecture that lets you deploy automated test solutions for various standards in a relatively short time.

### Key Features

The key features of the D-PHY application are:

- Allows conformance testing to the latest Compliance Test Specification (CTS)
- Automated testing:
  - Eliminates user intervention to conduct time-consuming testing
  - Reduces the time required to conduct testing
  - Enables you to test devices faster
- Customizing the setup:
  - Allows you to modify the test setup as per the DUT configuration
  - Automatically calculates unit intervals based on the DUT data rates
- Margin testing and characterization:
  - Allows custom-limits or limits-editing to perform Margin testing
  - Performs characterization of your design
- Detailed test reporting:
  - Provides a Pass/Fail summary table
  - Provides margin details on each test
  - Provides a consolidated report for all tests
- Selective testing:
  - Performs fully-automated testing for transmitter measurements
  - Allows you to select individual tests or test groups in the tree-structure
  - Avoids repeated testing, through accurate and reliable results from a single run

## Starting the Application


The application uses a USB dongle that contains the license key. This dongle must be present on the instrument hosting the D-PHY application.

The application also checks for a file, called Resources.xml, located in My TekExpress folder. If this file is not found, instrument discovery is performed before launching D-PHY. The Resources.xml file contains information regarding instruments available on network.

When the application starts, it checks for the appropriate license key. If the valid license key is not present, the application switches to the “Evaluation” mode. If the application fails to detect the dongle, it continues to run in Evaluation mode.

Start the application in one of the following ways:

- Click **Start > Programs > Tektronix > TekExpress > TekExpress D-PHY**.

- Double click the icon  on the desktop.
- If you have previously saved a session, double-click the session file stored under My TekExpress\D-PHY.






---

**NOTE.** *If the application was not terminated properly during the last use, a dialog box asks to recall the previously unsaved session.*

---


## Resizing the Application Window

To minimize the application, click  on the application title bar. To restore the application to its previous size, select  in the Windows task bar.

To maximize the application, click . To restore the application to its previous size, click on the application title bar.

## Exiting the Application

To exit the application, do one of the following:

- Click **File > Exit**.
- Click  on the application title bar.

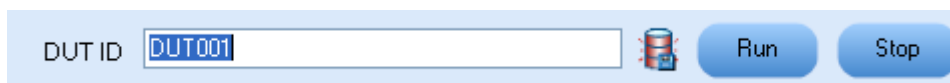
---

**NOTE.** *Using other methods to exit the application results in abnormal termination of the application.*



---

## Global Controls

The menus and controls that appear outside the individual tabs are called “Global Controls”. These are used to specify the devices to be tested.



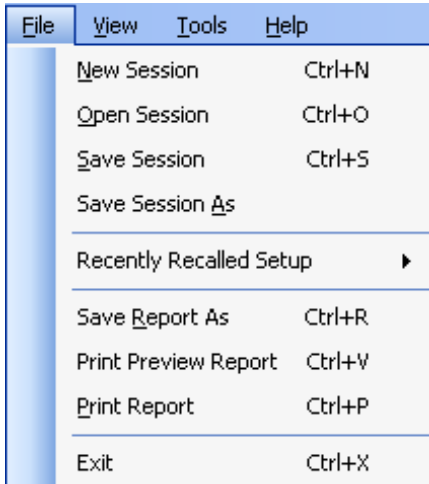
**Table 4: Global controls**

Control name	Function
DUT	The device ID is specified at the global level and the information is stored in the default location for all data files. This field cannot be empty and does not allow these special characters (.,,.....,/,/?"< *"). The maximum length of characters allowed is 32.
	Displays the status of the disk space. When the disk space is low, a warning dialog appears to perform the cleanup and continue working on the application.
	You will be able to run, pause, resume, and stop the tests.

## Menus

### File Menu

Click **File** on the application menu bar.

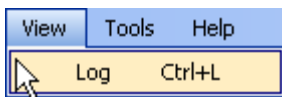


The File menu has the following selections:

Menu	Function
New Session	Starts a default session of D-PHY.
Open Session	Opens a saved session.
Save Session	Saves the session.
Save Session As	Saves a session in a different name.
Recently Recalled Setup	Recalls the recently saved setup.
Save Report As	Saves the report in user specified location.
Print Preview Report	Previews the report before printing.
Print Report	Opens the Windows "Print" dialog box.
Exit	Closes the application.

### View Menu

Click **View** on the application menu bar.



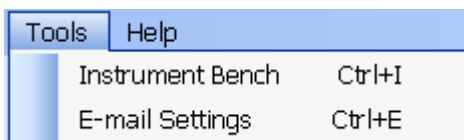
The View menu has the following selections:

Menu	Function
Log	Opens the log (log.xml) file in the default viewer.

### Tools Menu

Click **Tools** on the application menu bar.





The Tools menu has the following selections:

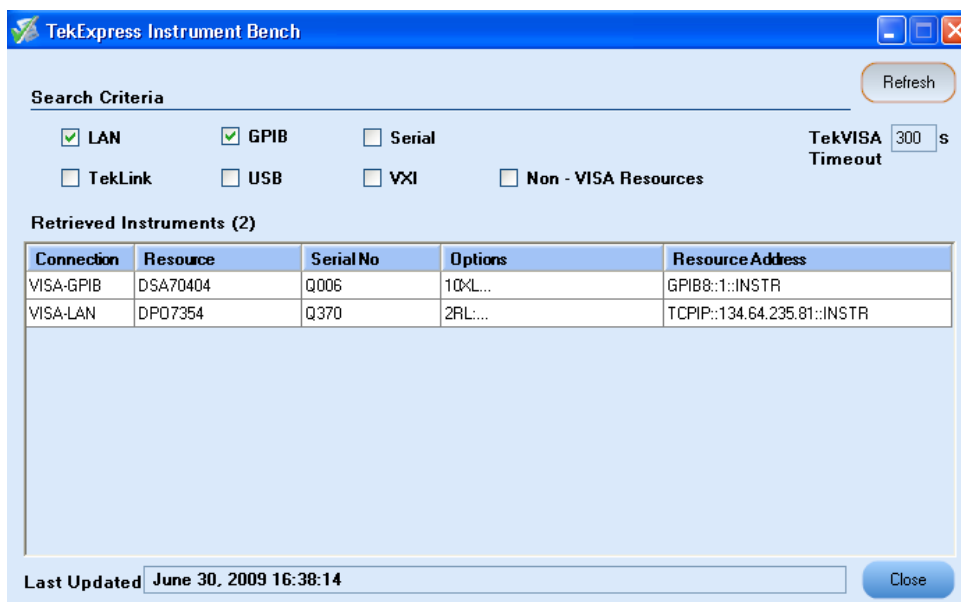
Menu	Function
<a href="#">Instrument Bench (see page 17)</a>	Shows the list of instruments connected to the test setup.
<a href="#">E-mail Setting (see page 18)</a>	Allows you to configure and set the e-mail options.

### Instrument Bench

The Instrument Bench window shows the list of VISA and Non-VISA resources found on different interfaces/connections. It serves two purposes at the launch of TekExpress:

- Discovers the connected instruments.
- Confirms the instrument connection setup.

When you click **Tools > Instrument Bench**, the following dialog box is displayed:

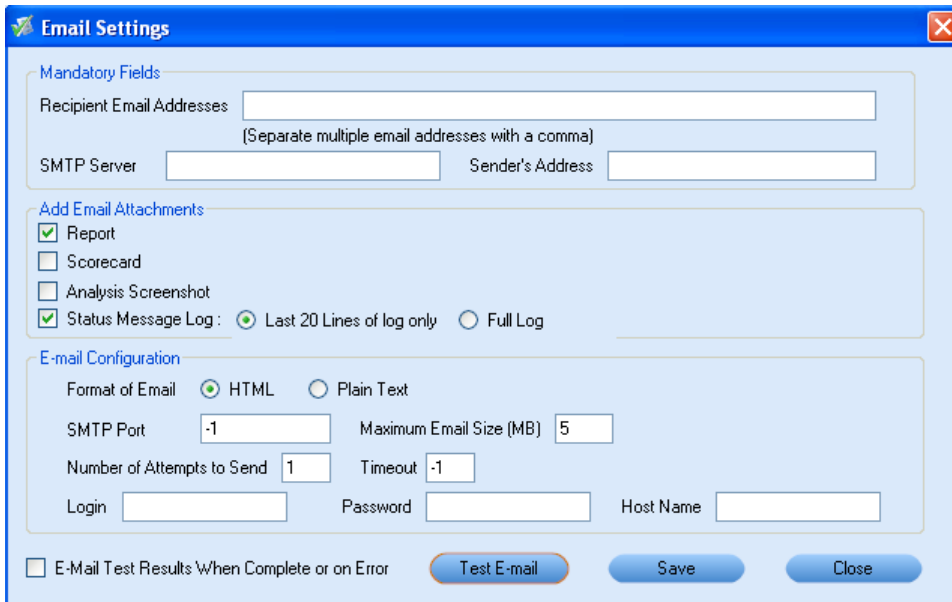


## E-mail Settings

The E-mail Settings utility allows you to configure and set the e-mail options. The following fields are mandatory for receiving e-mail notification from TekExpress:

1. Recipient Email Addresses. For example, User@domain.com
2. Sender’s Address
3. SMTP Server address of the Mail server configured at client location

If any of the above mentioned fields are left blank, the settings will not get saved and e-mail notification will not be sent.



Check the option “E-mail Test Results When Complete or on Error” to receive the e-mail. The attachment list depends on the choice which is made while configuring the email setup.

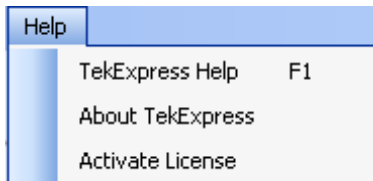
---

**NOTE.** *The Analysis Screenshot option in the Add Email Attachments pane is not yet functional.*

---

## Help Menu

Click **Help** on the application menu bar.



The Help menu has the following selections:

<b>Menu</b>	<b>Function</b>
TekExpress Help	Displays TekExpress Help (F1).
About TekExpress	Displays the TekExpress screen with the application title. Also displays the application details such as software name, version number, and copyright.
<a href="#">Activate License (see page 8)</a>	Displays the details of activating the application.

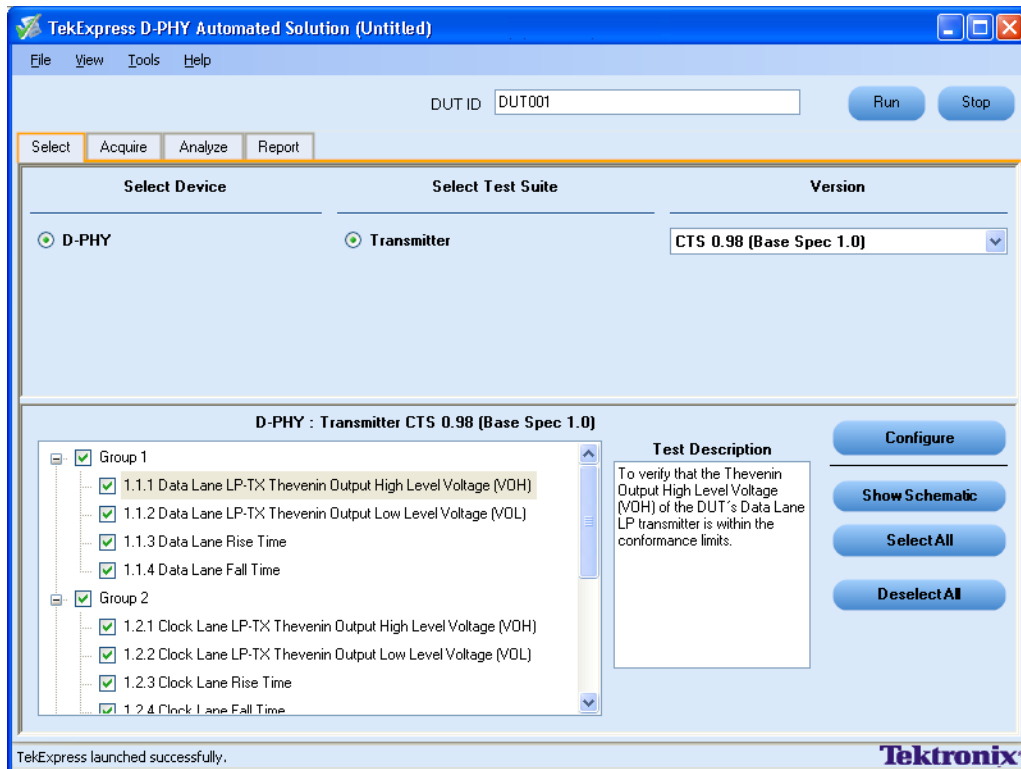


## Select Test(s)

The application tests the following D-PHY, single-ended tests:

- Group 1 tests: Data Lane LP-TX Thevenin Output High Level Voltage (VOH), Data Lane LP-TX Thevenin Output Low Level Voltage (VOL), Data Rate Rise Time, and Data Rate Fall Time.
- Group 2 tests: Clock Lane LP-TX Thevenin Output High Level Voltage (VOH), Clock Lane LP-TX Thevenin Output Low Level Voltage (VOL), Clock Rate Rise Time, and Clock Rate Fall Time.
- Group 3 tests: Data Lane HS Entry: Data Lane TLPX Value, Data Lane HS Entry: THS-PREPARE Value, and Data Lane HS Entry: THS-PREPARE + THS-ZERO Value.
- Group 5 tests: HS Entry TLCK PREValue, HS Exit TLCK POSTValue, and HS Clock Rising Edge Alignment to First Payload Bit.

Use the Select panel to select configure, and run one or more groups of tests.







**Select Device.** Select the device type. The default value is D-PHY. Enter the DUT ID for the device.

**Select Test Suite.** Select the test suite. The default value is Transmitter.

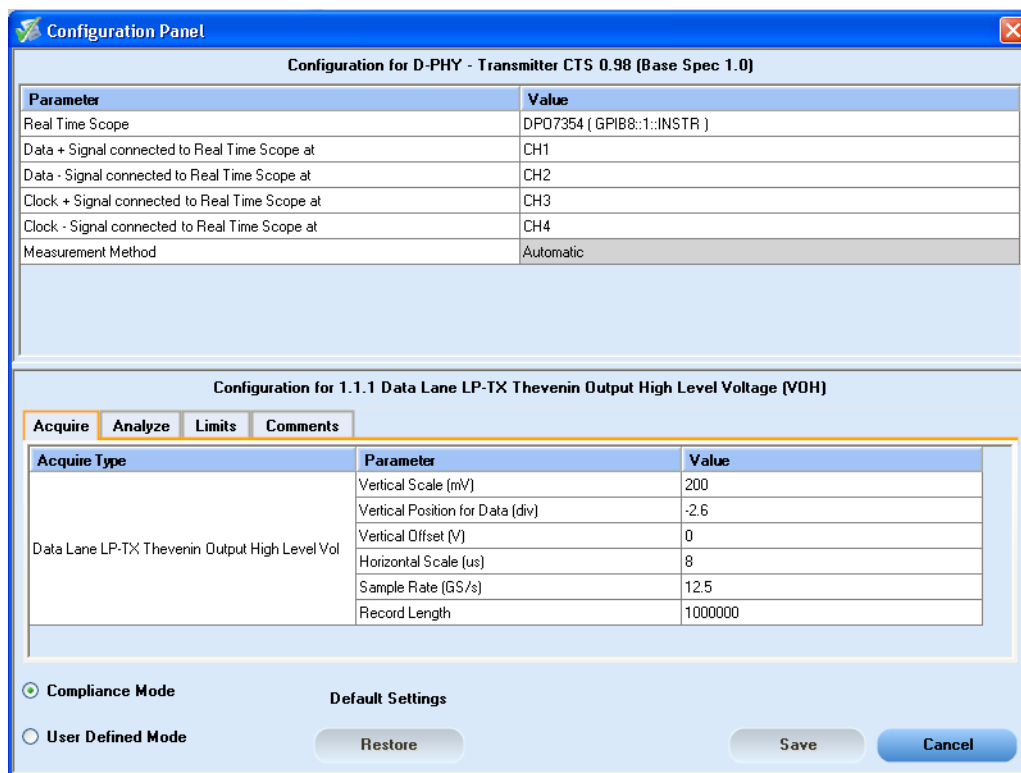
**Version.** Select the Compliance Test Specification version, the default value is CTS 0.98. For the selected CTS version, select any particular test, or one or more test groups.

**Table 5: Select panel buttons**

Button	Description
	Opens the configuration panel for the selected test.
	Opens the schematic for the selected test. This is useful if you want to verify the test setup before running the test.
	Selects all tests in the table.
	Deselects all tests in the table.

## Configure and Run the Test(s)

The configuration panel is used to view and edit the parameters associated with the acquisition and the analysis of the selected test. [Click here](#) for the options.



**NOTE.** *If any of the test parameters are grayed, it means that these parameters are not editable.*

The upper part of the Configuration Panel shows parameters and their values. These parameters are common for a group of tests. Parameters specific to a select test are shown in the lower part of the Configuration Panel. Test-specific parameters include acquisition, analysis, and limit parameters.

The following table lists the common parameters and their values:

**Table 6: Common parameters and values**

Parameter	Value
Real Time Scope	Select the oscilloscope to which to connect.
Data+ Signal connected to Real Time Scope at	Select the channel (CH1–CH4) to which the Data+ signal is connected.
Data– Signal connected to Real Time Scope at	Select the channel (CH1–CH4) to which the Data– signal is connected.
Clock+ Signal connected to Real Time Scope at	Select the channel (CH1–CH4) to which the Clock+ signal is connected.
Clock– Signal connected to Real Time Scope at	Select the channel (CH1–CH4) to which the Clock– signal is connected.
Measurement Method	Select the measurement method: Automatic or Manual. In the Automatic mode, you perform a test without any intervention. In the Manual mode, you will be prompted to place the cursors on the acquired waveform before the application proceeds with the analysis.  <b>NOTE.</b> <i>The Manual method is available only in the User Defined mode.</i>

**NOTE.** *If you change the channel selection for Data+ (DP), Data– (DN), Clock+ (CP), and Clock– (CN) signals, make sure that you change the trigger source correspondingly in the Analyze tab for each of the measurements. If you do not do this, the waveform might not trigger and the measurements might not be completed.*

The following table lists the tabs and button controls for the tests:

**Table 7: Configure panel buttons**

Configure parameters	Description
<a href="#">Acquire (see page 25)</a>	Shows the various parameters related to acquisition of a selected test.
<a href="#">Analyze (see page 27)</a>	Shows the various parameters related to analysis of a selected test. These parameters can vary from one test to another test. For example, Vertical scale, Vertical offset, Horizontal scale, and others.
<a href="#">Limits (see page 26)</a>	Applies to a specific version. It shows the measurement lower and upper limits using different types of comparisons like EQ(==), NE(!=), GT(>), LT(<), GE(>=), LE(<=), GTLT(> <), GELT(>= <), GTLE(> <=), LTGT(< >), LEGE(<= >=), LEGT(<= >), LTGE(< G=), GELE(>= <=). Some use unary operands and some use binary operands. For unary operands, only value1 (on the left) is active.
Comments	Specify a comment up to 256 characters long for selected test.
Restore	Restores the default values.

**Table 7: Configure panel buttons (cont.)**

Configure parameters	Description
Apply	Applies the configuration parameters that you have set.
Close	Dismisses the dialog box.

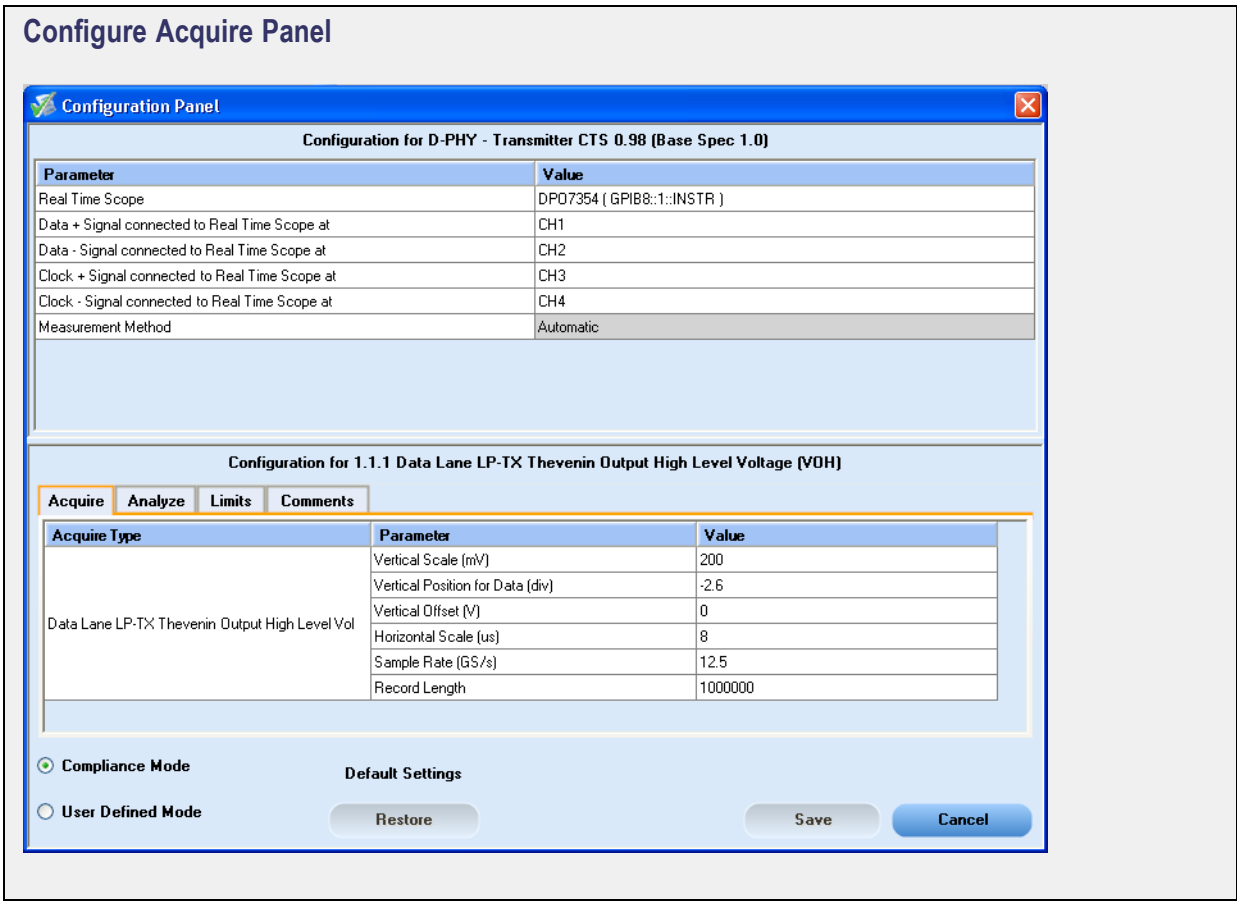
Click **Run** in the Select panel to run the selected tests.

Refer to the following table for various test limit comparisons:

**Table 8: Test limit comparisons**

Comparison string	Description
EQ(==)	Equal to
NE(!=)	Not equal to
GT(>)	Greater than
LT(<)	Less than
GE(>=)	Greater than or Equal to
LE(<=)	Less than or Equal to
GTLT(><)	Greater than and Less than
GELT(>=<)	Greater than or equal to and Less than
GTLE(><=)	Greater than and Less or equal to
LTGT(<>)	Less than and Greater than
LEGE(<= >=)	Less than or equal to and Greater than or equal to
LEGT(<= >)	Less than or equal to and Greater than
LTGE(< >=)	Less than and Greater than or equal to
GELE(>= =<)	Greater than or equal to and Less than or equal to





- You have the following options:
- Restore compliance mode values.
  - Change the parameters associated with analysis configuration.
  - Change the test limits in the User Defined mode.
  - Add comments for the selected test that would appear in the Report panel.

### Configure Limits Panel

**Configuration Panel**

Configuration for D-PHY - Transmitter CTS 0.98 (Base Spec 1.0)

Parameter	Value
Real Time Scope	DPD7354 ( GPIB8::1::INSTR )
Data + Signal connected to Real Time Scope at	CH1
Data - Signal connected to Real Time Scope at	CH2
Clock + Signal connected to Real Time Scope at	CH3
Clock - Signal connected to Real Time Scope at	CH4
Measurement Method	Automatic

Configuration for 1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)

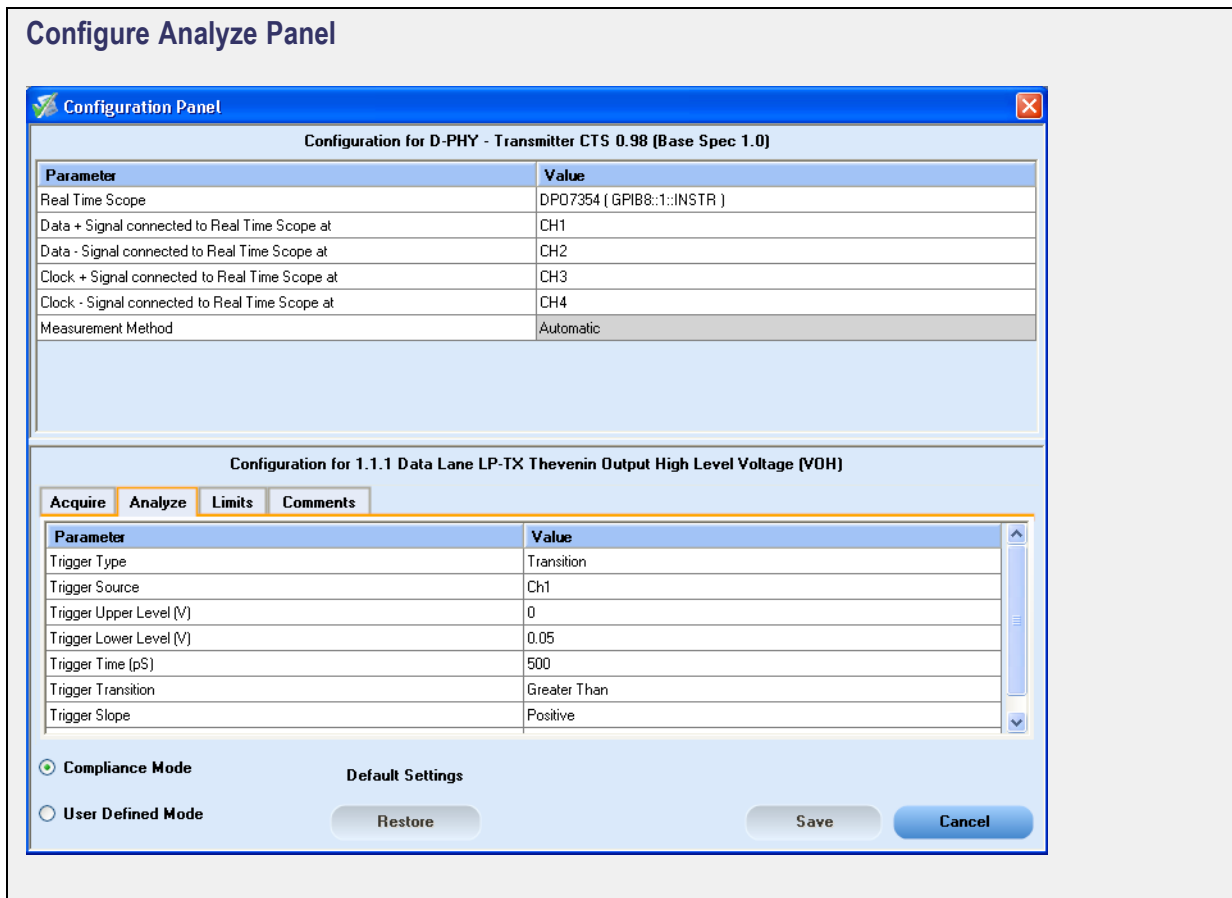
Acquire Analyze **Limits** Comments

Details	Value1	Compare String	Value2
Data Lane LP-TX Thevenin Output High Level Voltage DP (V)	1.1	Greater Than Or Equal To Less Than Or Eq	1.3
Data Lane LP-TX Thevenin Output High Level Voltage DN (V)	1.1	Greater Than Or Equal To Less Than Or Eq	1.3

Compliance Mode      Default Settings

User Defined Mode

Restore      Save      Cancel



## View and Select Connected Instruments

### Viewing Connected Instruments

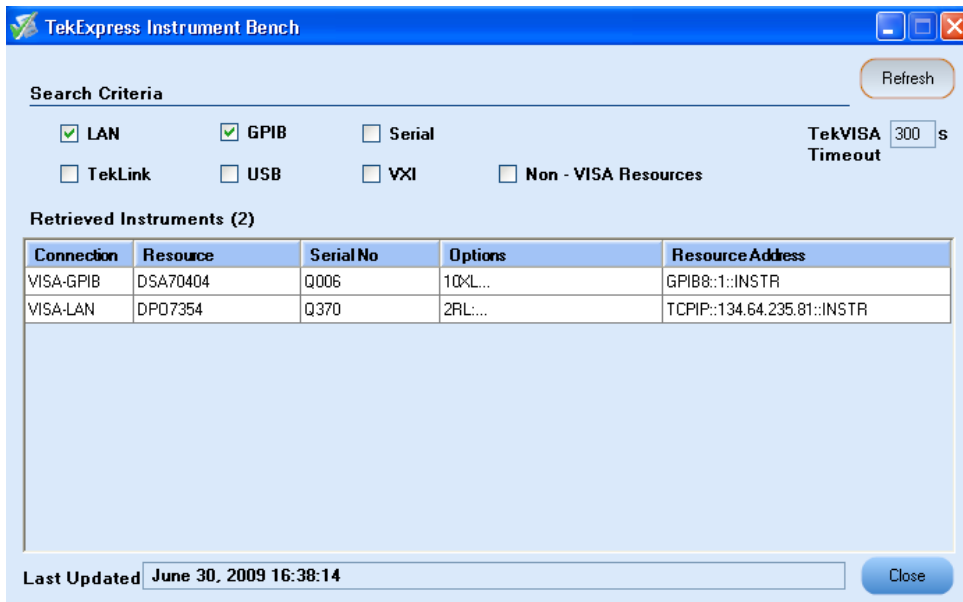
The **Tools > Instrument Bench** menu item is used to discover connected instruments required for the tests. The application uses TekVISA to discover the connected instruments. Once the operation is done, the Instrument Bench dialog box resumes operation and lists the instrument-related details based on the selected search criteria.

---

**NOTE.** When the TekVISA Instrument Manager checks for connected Instruments, the Instrument Bench dialog box does not respond.

---

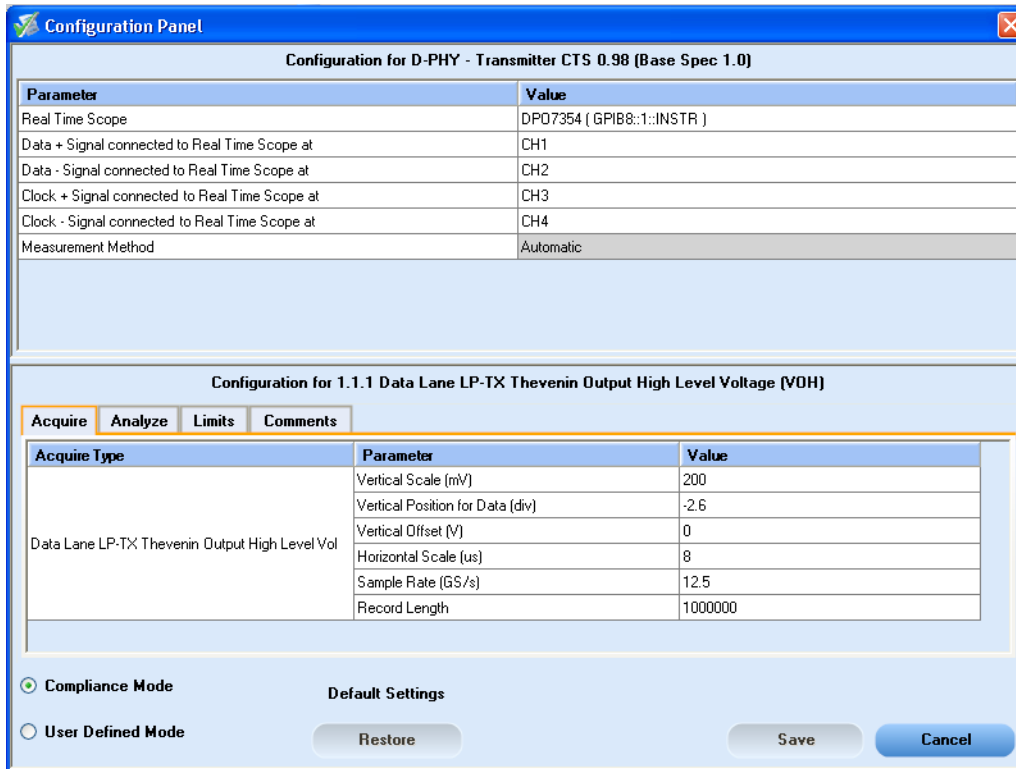
For example, if you select LAN and GPIB as the search criteria in the Instrument Bench dialog box and click Refresh, the TekVISA Instrument Manager checks for the availability of instruments over LAN and the details of the instruments are displayed under Retrieved Instruments table.



Provide the time in the TekVISA Refresh Timeout (Seconds) field, within which if the TekVISA Instrument Manager does not find the instruments, the TekExpress application resumes the operation. If you choose Non-VISA resources, all the instruments supported by TekExpress but not communicating over the VISA layer can be searched.

## Selecting Connected Instruments

View the instruments connected in the Configuration panel. The upper half of the panel displays the general parameters for the tests under the selected test suite. Select the Real Time Scope to connect to from the drop-down list.




---

**NOTE.** The list of instruments displayed is specific to the selected test suite. It does not show all the connected instruments.

---

## Use Prerecorded Waveforms for Analysis

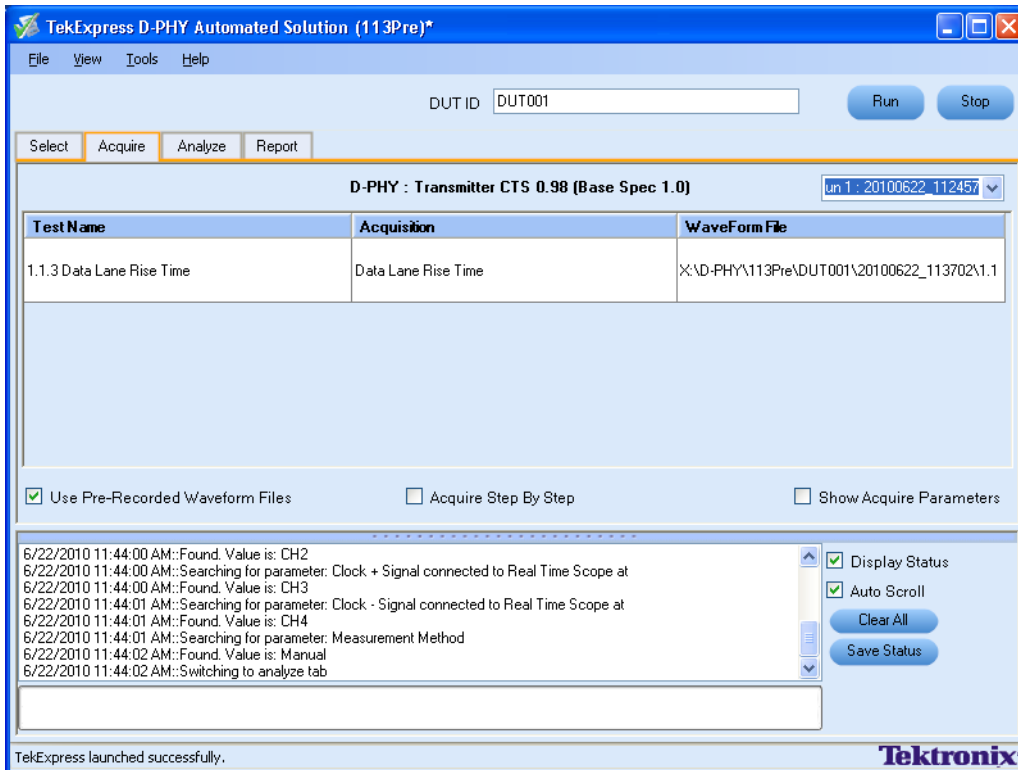
Select the Acquire tab to see unique acquisitions, acquisition parameters, acquisition status, and prerecorded waveform files of the selected test suite. The Acquire panel is specific to a group and gets updated every time the selected test group is changed. This panel shows the acquisition details for the tests in the currently selected group. The tests with the common acquisition parameters are grouped together and shown as a single acquisition.

---

**NOTE.** You can only use saved sessions in which the tests have been run in Automatic mode for prerecorded testing. You cannot use the sessions in which tests have been run in Manual mode for prerecorded testing.

---

**NOTE.** You cannot use the prerecorded waveform option for the 1.5.3 HS Clock Rising Edge Alignment to First Payload Bit test.



Column name	Function
Test Name	Name of the tests performing acquisitions. One or more tests can perform the same acquisitions.
Acquisition	Test acquisition status of the running test passed at intervals.
Waveform File	Prerecorded waveform files of unique acquisitions. You can select waveform files by clicking the hyperlink. This allows you to select any waveform file using the standard file open window.

**NOTE.** To use the prerecorded option, it is recommended that you use a waveform file (.wfm) which has been captured from a Tektronix oscilloscope. This eliminates the need to use an oscilloscope. You can manually select waveforms and perform the tests with a click of the Run button.

The following Acquire source options are available:

- **Use Pre-recorded Waveform Files:** Enabling or disabling the option shows or hides the waveform file column in the acquisition table.

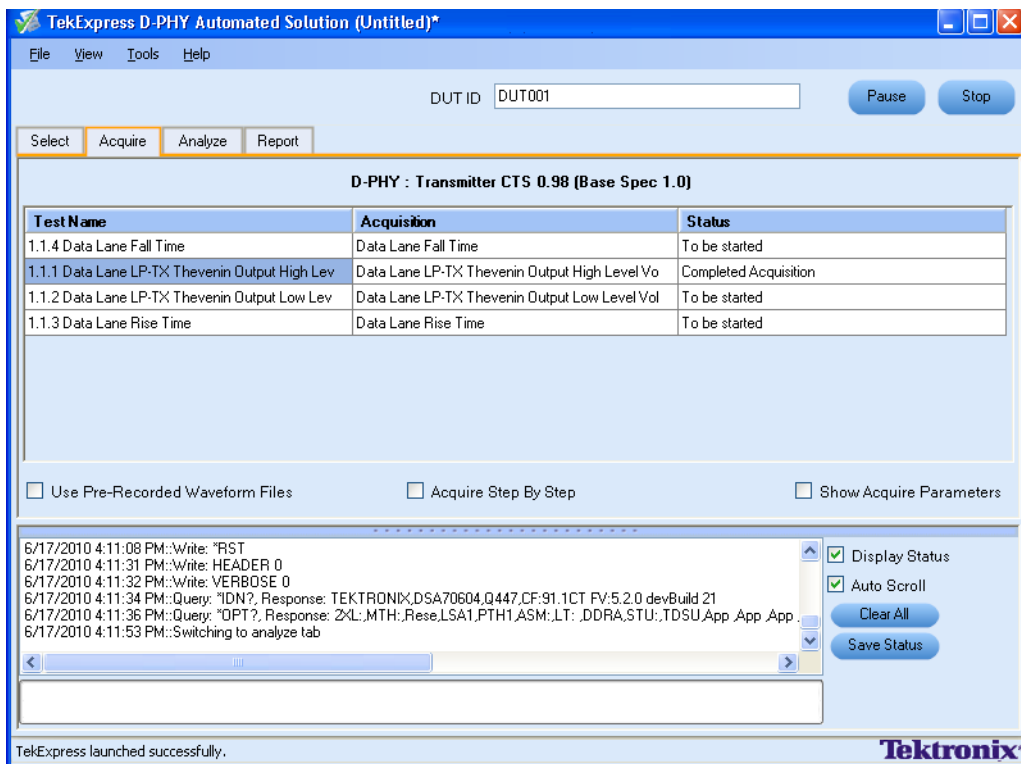
To know how to use the prerecorded waveform files for performing a test or the selected test group, [click here \(see page 60\)](#).

- **Acquire Step by Step:** Selecting this prompts you at the end of each acquisition before proceeding to the next one.
- **Show Acquire Parameters:** Selecting this shows the acquisition parameters.

When you select “Use Pre-recorded”, the first column shows the Acquisition, the second column shows the Test, and the next column shows the Waveform File for analysis.

## Use Acquire Live Waveform for Analysis

Click the Acquire tab to see acquisition parameters, and acquisition status of the selected test suite. The Acquire panel is specific to a suite and gets updated every time the selected test suite is changed. This panel shows the acquisition details for the tests in the currently selected suite. The tests with common acquisition parameters are grouped together and shown as a single acquisition.

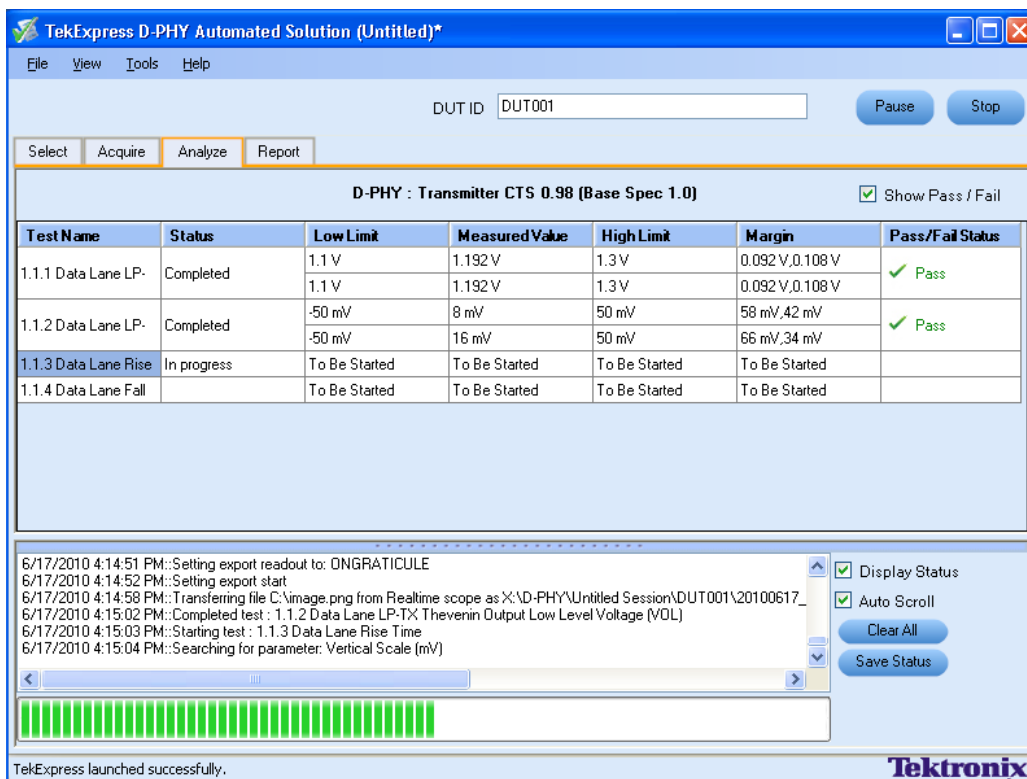


**Table 9: Controls on the Acquire panel**

Column name	Function
Test	Displays the name of the selected test for performing acquisitions. One or more tests can perform the same acquisitions.
Status	Acquisition status of the running test at intervals. The messages are passed only in live acquisition.
Acquisition	Updates the location of the acquisition name.
<b>Acquire Options</b>	
Use Pre-recorded Waveform Files	When enabled, uses pre-recorded waveforms for the selected test.
Acquire Step By Step	When enabled, displays the reference input waveform of the selected test. This helps you to compare the input waveform with the reference waveform, allows you to change the setup before acquiring the waveforms, and then proceeds with the next selected test.

## View the Progress of Analysis

Click the Analyze tab to view the progress of the analysis. As the analysis of each test is complete, the result value is updated.





## Analyze Table

The Analyze table contains the following:

- The test name.
- The status of the tests that are being run.
- The low and high limit values, and margins, for the tests.
- The measured values for the tests.
- The pass/fail status of the tests.

The tests that are not yet started are shown with a “To be Started” status. A summarized status of the currently running test is shown in the Status Messages panel.

The **Status Messages** window time-stamps all run time messages and displays them. You can do the following:

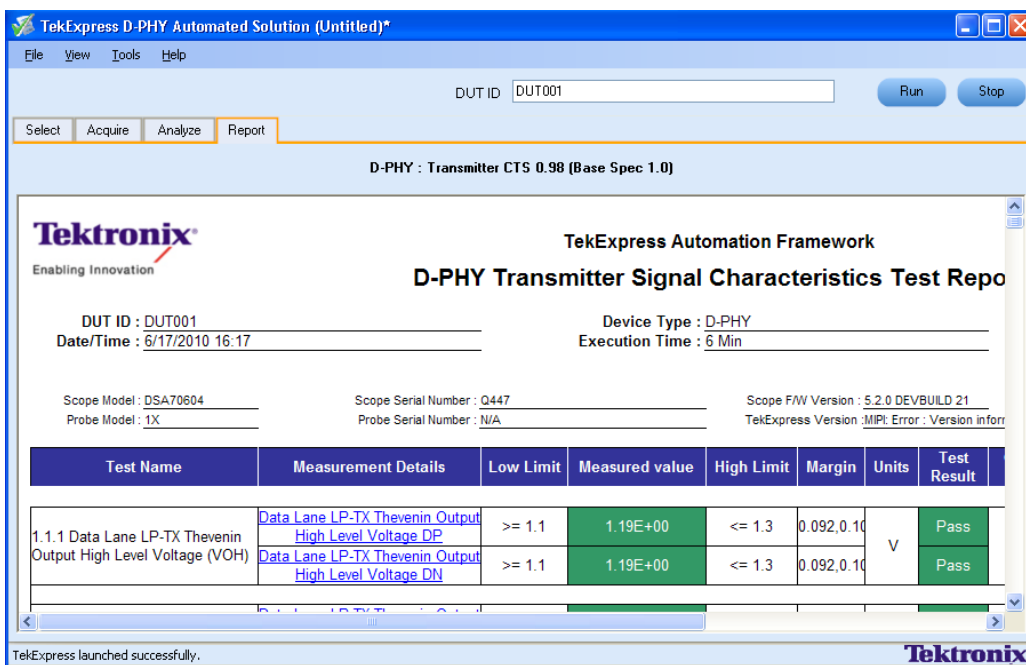
- **Display Status:** Enable/Disable status messages.
- **Auto Scroll:** Scroll the status messages automatically.
- **Clear All:** Clear all status messages in Status Window.
- **Save Status:** Save all status messages in text file. Displays a standard save file window and saves the status messages in the user specified file.

## View the Report

After the analysis, a report is automatically generated and displayed in the report panel. The device information such as oscilloscope model, serial number, firmware version of the oscilloscope, version information of the TekExpress and the application, start time, and the execution time are displayed.

A table shows the test name, measurement details, test result (pass/fail), compliance mode, and analysis time. Save the report as an MHT file using the **File > Save As** option.

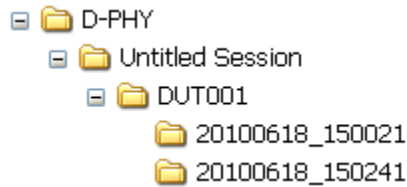
**NOTE.** *If you use a secondary monitor, verify that the screen resolution of the secondary monitor is the same as that of the primary display. This ensures that captured images in the reports are of the same size.*



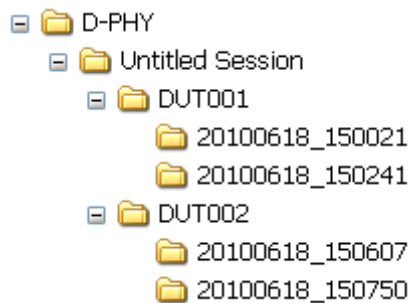
## View Test Related Files

All the test related files for currently selected tests are always saved under: My Documents\My TekExpress\D-PHY\SessionName.

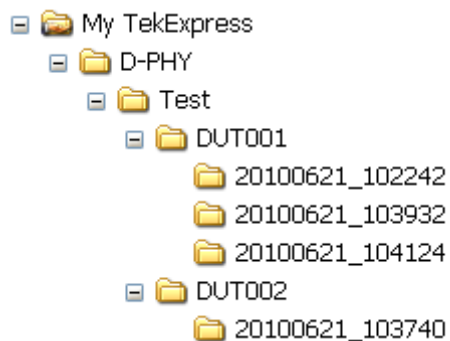
**Case 1.** An example image of data when the session is still not saved and is in the My TekExpress/D-PHY path as “Untitled Session”, the DUT ID is set to DUT001, and the tests are performed twice (two folders are created with the file naming convention YYYYMMDD\_HHMMSS (Date\_Time)).



**Case 2.** An example image of data when the session is still not saved and is in the My TekExpress/D-PHY path as “Untitled Session”, the DUT001 folder already present (tests are performed twice), and the DUT ID is now set to DUT002 (here the tests are performed once).



**Case 3.** An example image of data when the session is saved and is in the My TekExpress/D-PHY path as “Test”.





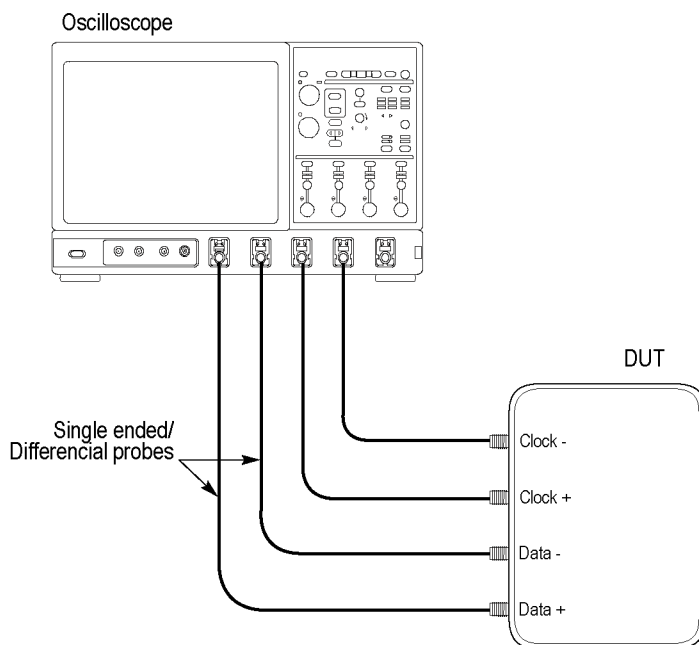


## Connection Setup

You need the following equipment (see [Compatibility \(see page 5\)](#)):

- A supported Tektronix DPO oscilloscope
- Single-ended probe
- Device under test

Make connections as follows:



## Configuring Parameter Values

To configure a test, do the following:

1. In the Select panel click **Configure**.
2. In the Configure panel, set the following general parameters that are common for all tests:

Parameter	Value
Real Time Scope	Select the oscilloscope to which to connect.
Data+ Signal connected to Real Time Scope at	Select the channel (CH1-CH4) to which the Data+ signal is connected.

Parameter	Value
Data– Signal connected to Real Time Scope at	Select the channel (CH1–CH4) to which the Data– signal is connected.
Clock+ Signal connected to Real Time Scope at	Select the channel (CH1–CH4) to which the Clock+ signal is connected.
Clock– Signal connected to Real Time Scope at	Select the channel (CH1–CH4) to which the Clock– signal is connected.
Measurement Method	Select the measurement method: Automatic or Manual.  <b>NOTE.</b> <i>The Manual method is available only when in the User Defined mode.</i>

The Automatic measurement method mode, lets you perform a test without any intervention. You must select one or more tests, configure the tests (or use the default values) and click Run to perform the test. The application acquires and analyzes the data, and displays a report when complete.

If you select the Manual mode, select one or more tests, configure the tests (or use the default values) and click Run to perform the test. The application begins to acquire data. A window opens where you are prompted to place the cursors on the acquired waveform and click **OK**. Go to the TekScope application and zoom the waveform around the trigger point. Place the cursors on the waveform to capture the values at those points and click **OK** in the window. The application completes the test and displays the report.

---

**NOTE.** *Other applications will display there windows in the background. The TekScope application takes precedence over other applications. Use Alt+Tab to view the other application windows.*

---

3. Select either **Compliance Mode** or **User Defined Mode**. In the User Defined Mode, you can modify the test parameters but you will no longer be testing against the compliance standards.

## Connections for Data Lane LP-TX Thevenin Output High Level Voltage (VOH)

[Click here](#) for information on connections for the Data Lane LP-TX Thevenin Output High Level Voltage (VOH) test.

## Configure and Run Data Lane LP-TX Thevenin Output High Level Voltage (VOH)

1. In the Select panel, select Device as **D-PHY** and Test Suite as **Transmitter**. The Version default value is **CTS 0.98 (Base Spec 1.0)**.
2. Select a test and click **Configure**.

3. In the Configuration Panel, set the following parameters for the test. The following table lists the compliance mode values and the permitted range of values for the user-defined mode:

**Table 10: Configuration parameters for Data Lane LP-TX Thevenin Output High Level Voltage (VOH)**

Parameter	Default	User defined mode
Vertical Scale (mV)	200	50, 100, 200
Vertical Position for Data (div)	-2.6	+3 to -3
Vertical Offset (V)	0	Do not change
Horizontal Scale (us)	8	8, 10, 12
Sample Rate (GS/s)	12.5	7.5, 10, 12.5
Record Length	1000000	200000, 500000, 1000000

4. Click **Analyze**. View and change the following analysis parameters:

**Table 11: Analysis parameters for Data Lane LP-TX Thevenin Output High Level Voltage (VOH)**

Parameter	Default	User defined mode
Trigger Type	Transition	Transition
Trigger Source	Ch1	Ch1, Ch2, Ch3, Ch4
Trigger Upper Level (V)	1	Enter a value based on the test
Trigger Lower Level (V)	0	0.05, 0.1, 1
Trigger Time (pS)	500	250, 500
Trigger Transition	Greater Than	Greater Than, Less Than
Trigger Slope	Positive	Positive, Negative, Either
Trigger If Violation	Occurs	Occurs, Logic

5. Click **Limits** to view and change the Data Lane LP-TX Thevenin Output High Voltage DP and Data Lane LP-TX Thevenin Output High Voltage DN values, and the compare string.
6. Click **Comments** to enter comments. The comments are shown in the test report.
7. Click **Apply** to accept the changes, and then click **Close**.
8. In the Select panel, click **Run** to start the test. The Acquire panel shows the progress and status of the test as it runs.



## Connections for Data Lane LP-TX Thevenin Output Low Level Voltage (VOL)

[Click here](#) for information on connections for the Data Lane LP-TX Thevenin Output Low Level Voltage (VOL) test.

## Configure and Run Data Lane LP-TX Thevenin Output Low Level Voltage (VOL)

1. In the Select panel, select Device as **D-PHY** and Test Suite as **Transmitter**. The Version default value is **CTS 0.98 (Base Spec 1.0)**.
2. Select the test and click **Configure**.
3. In the Configuration Panel, set the following parameters for the test. The following table lists the compliance mode values and the permitted range of values for the user-defined mode:

**Table 12: Configuration parameters for Data Lane LP-TX Thevenin Output Low Level Voltage (VOL)**

Parameter	Default	User defined mode
Vertical Scale (mV)	200	50, 100, 200
Vertical Position for Data (div)	-2.6	+3 to -3
Vertical Offset (V)	0	Do not change
Horizontal Scale (us)	8	8, 10, 12
Sample Rate (GS/s)	12.5	7.5, 10, 12.5
Record Length	1000000	200000, 500000, 1000000

4. Click **Analyze**. View and change the following analysis parameters:

**Table 13: Analysis parameters for Data Lane LP-TX Thevenin Output Low Level Voltage (VOL)**

Parameter	Default	User defined mode
Trigger Type	Transition	Transition
Trigger Source	Ch1	Ch1, Ch2, Ch3, Ch4
Trigger Upper Level (V)	1	Enter a value based on the test
Trigger Lower Level (V)	0	0.05, 0.1, 1
Trigger Time (pS)	500	250, 500
Trigger Transition	Greater Than	Greater Than, Less Than
Trigger Slope	Negative	Positive, Negative, Either
Trigger If Violation	Occurs	Occurs, Logic

5. Click **Limits** to view and change the Data Lane LP-TX Thevenin Output Low Voltage DP and Data Lane LP-TX Thevenin Output Low Voltage DN values in mV, and the compare string.
6. Click **Comments** to enter comments. The comments are shown in the test report.
7. Click **Apply** to effect the changes, and then click **Close**.
8. In the Select panel, click **Run** to start the test. The Acquire panel shows the progress and status of the test as it runs.

## Connections for Data Lane Rise Time

[Click here](#) for information on connections for the Data Lane Rise Time test.

## Configure and Run Data Lane Rise Time

1. In the Select panel, select Device as **D-PHY** and Test Suite as **Transmitter**. The Version default value is **CTS 0.98 (Base Spec 1.0)**.
2. Select the test and click **Configure**.
3. In the Configuration Panel, set the following parameters for the test. The following table lists the compliance mode values and the permitted range of values for the user-defined mode:

**Table 14: Configuration parameters for Data Lane Rise Time**

Parameter	Default	User defined mode
Vertical Scale (mV)	200	100, 200, 300
Vertical Position (div)	-2.6	+3 to -3
Vertical Offset (V)	0	Do not change
Horizontal Scale (us)	8	8, 10, 12
Sample Rate (GS/s)	12.5	7.5, 10, 12.5
Record Length	1000000	200000, 500000, 1000000

4. Click **Analyze**. View and change the following analysis parameters:

**Table 15: Analysis parameters for Data Lane Rise Time**

Parameter	Default	User defined mode
Trigger Type	Transition	Transition
Trigger Source	Ch1	Ch1, Ch2, Ch3, Ch4
Trigger Upper Level (V)	1	Enter a value based on the test
Trigger Lower Level (V)	0	0.05, 0.1, 1
Trigger Time (pS)	500	250, 500
Trigger Transition	Greater Than	Greater Than, Less Than

**Table 15: Analysis parameters for Data Lane Rise Time (cont.)**

Parameter	Default	User defined mode
Trigger Slope	Positive	Positive, Negative, Either
Trigger If Violation	Occurs	Occurs, Logic

5. Click **Limits** to view and change the Data Lane Rise Time DP and Data Lane Rise Time DN values in nS, and the compare string.
6. Click **Comments** to enter comments. The comments are shown in the test report.
7. Click **Apply** to effect the changes, and then click **Close**.
8. In the Select panel, click **Run** to start the test. The Acquire panel shows the progress and status of the test as it runs.

## Connections for Data Lane Fall Time

[Click here](#) for information on connections for the Data Lane Fall Time test.

## Configure and Run Data Lane Fall Time

1. In the Select panel, select Device as **D-PHY** and Test Suite as **Transmitter**. The Version default value is **CTS 0.98 (Base Spec 1.0)**.
2. Select the test and click **Configure**.
3. In the Configuration Panel, set the following parameters for the test. The following table lists the compliance mode values and the permitted range of values for the user-defined mode:

**Table 16: Configuration parameters for Data Lane Fall Time**

Parameter	Default	User defined mode
Vertical Scale (mV)	200	100, 200, 300
Vertical Position (div)	-2.6	+3 to -3
Vertical Offset (V)	0	Do not change
Horizontal Scale (us)	8	8, 10, 12
Sample Rate (GS/s)	12.5	7.5, 10, 12.5
Record Length	1000000	200000, 500000, 1000000

4. Click **Analyze**. View and change the following analysis parameters:

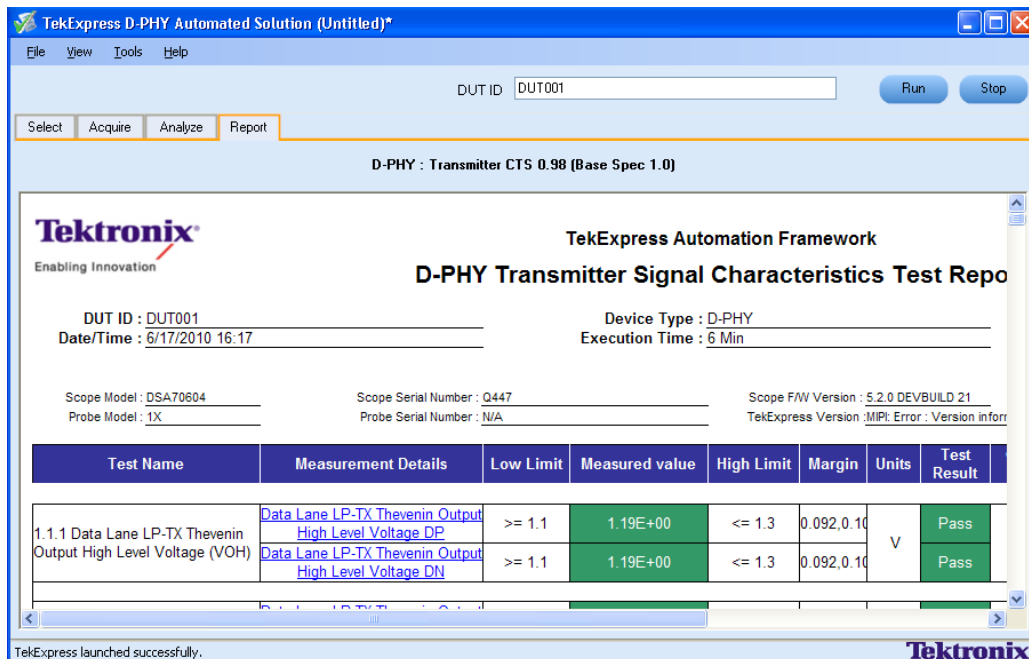
**Table 17: Analysis parameters for Data Lane Fall Time**

Parameter	Default	User defined mode
Trigger Type	Transition	Transition
Trigger Source	Ch3	Ch1, Ch2, Ch3, Ch4
Trigger Upper Level (V)	1	Enter a value based on the test
Trigger Lower Level (V)	0	0.05, 0.1, 1
Trigger Time (pS)	500	250, 500
Trigger Transition	Greater Than	Greater Than, Less Than
Trigger Slope	Negative	Positive, Negative, Either
Trigger If Violation	Occurs	Occurs, Logic

5. Click **Limits** to view and change the Data Lane Fall Time DP and Data Lane Fall Time DN values in nS, and the compare string.
6. Click **Comments** to enter comments. The comments are shown in the test report.
7. Click **Apply** to effect the changes, and then click **Close**.
8. In the Select panel, click **Run** to start the test. The Acquire panel shows the progress and status of the test as it runs.

## View Report for Group 1 Tests

The application automatically displays a report in the Report panel once the test is successfully completed. A table shows the test name, measurement details, test result (pass/fail), compliance mode, and analysis time. Save the report as an MHT file using the **File > Save As** option. The following screen shows the results for Group 1 tests.



## Connections for Clock Lane LP-TX Thevenin Output High Level Voltage (VOH)

[Click here](#) for information on connections for the Clock Lane LP-TX Thevenin Output High Level Voltage (VOH) test.

## Configure and Run Clock Lane LP-TX Thevenin Output High Level Voltage (VOH)

1. In the Select panel, select Device as **D-PHY** and Test Suite as **Transmitter**. The Version default value is **CTS 0.98 (Base Spec 1.0)**.
2. Select the test and click **Configure**.
3. In the Configuration Panel, set the following parameters for the test. The following table lists the compliance mode values and the permitted range of values for the user-defined mode:

**Table 18: Configuration parameters for Clock Lane LP-TX Thevenin Output High Level Voltage (VOH)**

Parameter	Default	User defined mode
Vertical Scale (mV)	200	50, 100, 200

**Table 18: Configuration parameters for Clock Lane LP-TX Thevenin Output High Level Voltage (VOH) (cont.)**

Parameter	Default	User defined mode
Vertical Position for Data (div)	-2.6	+3 to -3
Vertical Offset (V)	0	Do not change
Horizontal Scale (us)	8	8, 10, 12
Sample Rate (GS/s)	12.5	7.5, 10, 12.5
Record Length	1000000	200000, 500000, 1000000

4. Click **Analyze**. View and change the following analysis parameters:

**Table 19: Analysis parameters for Clock Lane LP-TX Thevenin Output High Level Voltage (VOH)**

Parameter	Default	User defined mode
Trigger Type	Transition	Transition
Trigger Source	Ch3	Ch1, Ch2, Ch3, Ch4
Trigger Upper Level (V)	1	Enter a value based on the test
Trigger Lower Level (V)	0	0.05, 0.1, 1
Trigger Time (pS)	500	250, 500
Trigger Transition	Greater Than	Greater Than, Less Than
Trigger Slope	Positive	Positive, Negative, Either
Trigger If Violation	Occurs	Occurs, Logic

5. Click **Limits** to view and change the Clock Lane LP-TX Thevenin Output High Voltage DP and Clock Lane LP-TX Thevenin Output High Voltage DN values in mV, and the compare string.
6. Click **Comments** to enter comments. The comments are shown in the test report.
7. Click **Apply** to effect the changes, and then click **Close**.
8. In the Select panel, click **Run** to start the test. The Acquire panel shows the progress and status of the test as it runs.

## Connections for Clock Lane LP-TX Thevenin Output Low Level Voltage (VOL)

[Click here](#) for information on connections for the Clock Lane LP-TX Thevenin Output Low Level Voltage (VOL) test.

### Configure and Run Clock Lane LP-TX Thevenin Output Low Level Voltage (VOL)

1. In the Select panel, select Device as **D-PHY** and Test Suite as **Transmitter**. The Version default value is **CTS 0.98 (Base Spec 1.0)**.
2. Select the test and click **Configure**.
3. In the Configuration Panel, set the following parameters for the test. The following table lists the compliance mode values and the permitted range of values for the user-defined mode:

**Table 20: Configuration parameters for Clock Lane LP-TX Thevenin Output Low Level Voltage (VOL)**

Parameter	Default	User defined mode
Vertical Scale (mV)	200	50, 100, 200
Vertical Position for Data (div)	-2.6	+3 to -3
Vertical Offset (V)	0	Do not change
Horizontal Scale (us)	8	8, 10, 12
Sample Rate (GS/s)	12.5	7.5, 10, 12.5
Record Length	1000000	200000, 500000, 1000000

4. Click **Analyze**. View and change the following analysis parameters:

**Table 21: Analysis parameters for Clock Lane LP-TX Thevenin Output Low Level Voltage (VOL)**

Parameter	Default	User defined mode
Trigger Type	Transition	Transition
Trigger Source	Ch3	Ch1, Ch2, Ch3, Ch4
Trigger Upper Level (V)	1	Enter a value based on the test
Trigger Lower Level (V)	0.05	0.05, 0.1, 1
Trigger Time (pS)	500	250, 500
Trigger Transition	Greater Than	Greater Than, Less Than
Trigger Slope	Positive	Positive, Negative, Either
Trigger If Violation	Occurs	Occurs, Logic

5. Click **Limits** to view and change the Clock Lane LP-TX Thevenin Output Low Voltage DP and Clock Lane LP-TX Thevenin Output Low Voltage DN values, and the compare string.
6. Click **Comments** to enter comments. The comments are shown in the test report.
7. Click **Apply** to effect the changes, and then click **Close**.
8. In the Select panel, click **Run** to start the test. The Acquire panel shows the progress and status of the test as it runs.

## Connections for Clock Lane Rise Time

[Click here](#) for information on connections for the Clock Lane Rise Time test.

## Configure and Run Clock Lane Rise Time

1. In the Select panel, select Device as **D-PHY** and Test Suite as **Transmitter**. The Version default value is **CTS 0.98 (Base Spec 1.0)**.
2. Select the test and click **Configure**.
3. In the Configuration Panel, set the following parameters for the test. The following table lists the compliance mode values and the permitted range of values for the user-defined mode:

**Table 22: Configuration parameters for Clock Lane Rise Time**

Parameter	Default	User defined mode
Vertical Scale (mV)	200	100, 200, 300
Vertical Position (div)	-2.6	+3 to -3
Vertical Offset (V)	0	Do not change
Horizontal Scale (us)	8	8, 10, 12
Sample Rate (GS/s)	12.5	7.5, 10, 12.5
Record Length	1000000	200000, 500000, 1000000

4. Click **Analyze**. View and change the following analysis parameters:

**Table 23: Analysis parameters for Clock Lane Rise Time**

Parameter	Default	User defined mode
Trigger Type	Transition	Transition
Trigger Source	Ch3	Ch1, Ch2, Ch3, Ch4
Trigger Upper Level (V)	1	Enter a value based on the test
Trigger Lower Level (V)	0	0.05, 0.1, 1
Trigger Time (pS)	500	250, 500
Trigger Transition	Greater Than	Greater Than, Less Than



**Table 23: Analysis parameters for Clock Lane Rise Time (cont.)**

Parameter	Default	User defined mode
Trigger Slope	Positive	Positive, Negative, Either
Trigger If Violation	Occurs	Occurs, Logic

5. Click **Limits** to view and change the Clock Lane Rise Time DP and Clock Lane Rise Time DN values in nS, and the compare string.
6. Click **Comments** to enter comments. The comments are shown in the test report.
7. Click **Apply** to effect the changes, and then click **Close**.
8. In the Select panel, click **Run** to start the test. The Acquire panel shows the progress and status of the test as it runs.

## Connections for Clock Lane Fall Time

[Click here](#) for information on connections for the Clock Lane Rise Time test.

## Configure and Run Clock Lane Fall Time

1. In the Select panel, select Device as **D-PHY** and Test Suite as **Transmitter**. The Version default value is **CTS 0.98 (Base Spec 1.0)**.
2. Select the test and click **Configure**.
3. In the Configuration Panel, set the following parameters for the test. The following table lists the compliance mode values and the permitted range of values for the user-defined mode:

**Table 24: Configuration parameters for Clock Lane Fall Time**

Parameter	Default	User defined mode
Vertical Scale (mV)	200	100, 200, 300
Vertical Position (div)	-2.6	+3 to -3
Vertical Offset (V)	0	Do not change
Horizontal Scale (us)	8	8, 10, 12
Sample Rate (GS/s)	12.5	7.5, 10, 12.5
Record Length	1000000	200000, 500000, 1000000

4. Click **Analyze**. View and change the following analysis parameters:

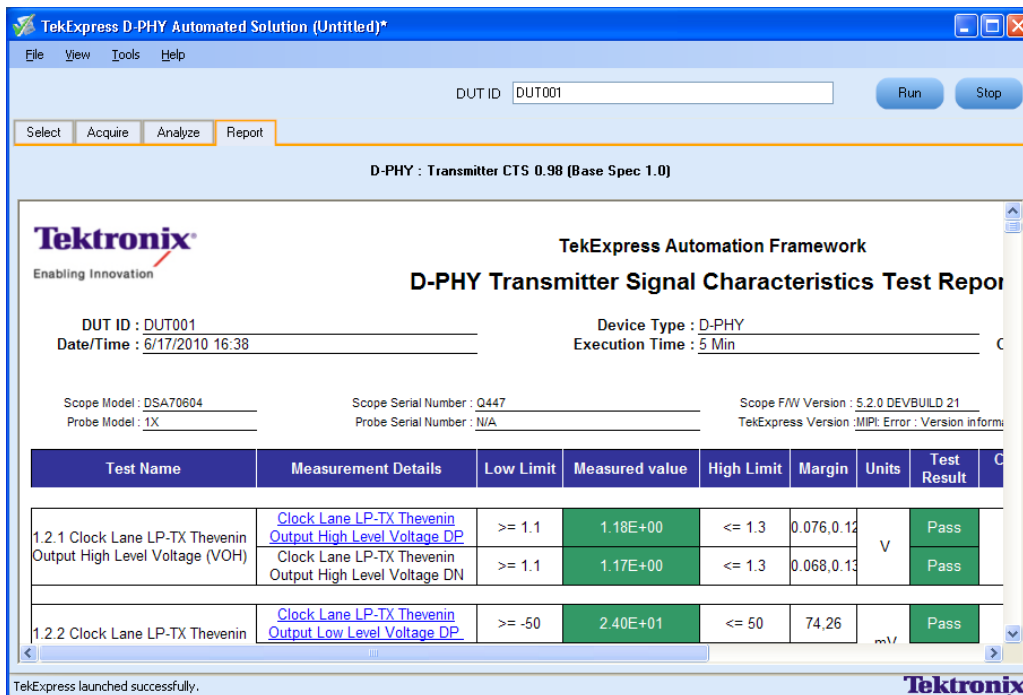
**Table 25: Analysis parameters for Clock Lane Fall Time**

Parameter	Default	User defined mode
Trigger Type	Transition	Transition
Trigger Source	Ch3	Ch1, Ch2, Ch3, Ch4
Trigger Upper Level (V)	1	Enter a value based on the test
Trigger Lower Level (V)	0	0.05, 0.1, 1
Trigger Time (pS)	500	250, 500
Trigger Transition	Greater Than	Greater Than, Less Than
Trigger Slope	Negative	Positive, Negative, Either
Trigger If Violation	Occurs	Occurs, Logic

5. Click **Limits** to view and change the Clock Lane Fall Time DP and Clock Lane Fall Time DN values in nS, and the compare string.
6. Click **Comments** to enter comments. The comments are shown in the test report.
7. Click **Apply** to effect the changes, and then click **Close**.
8. In the Select panel, click **Run** to start the test. The Acquire panel shows the progress and status of the test as it runs.

## View Report for Group 2 Tests

The application automatically displays a report in the Report panel once the test is successfully completed. A table shows the test name, measurement details, test result (pass/fail), compliance mode, and analysis time. Save the report as an MHT file using the **File > Save As** option. The following screen shows the results for Group 2 tests.



## Connections for Data Lane HS Entry: Data Lane TLPX Value

[Click here](#) for information on connections for the Data Lane HS Entry: Data Lane TLPX Value test.

## Configure and Run Data Lane HS Entry: Data Lane TLPX Value

1. In the Select panel, select Device as **D-PHY** and Test Suite as **Transmitter**. The Version default value is **CTS 0.98 (Base Spec 1.0)**.
2. Select the test and click **Configure**.
3. In the Configuration Panel, set the following parameters for the test. The following table lists the compliance mode values and the permitted range of values for the user-defined mode:

**Table 26: Configuration parameters for Data Lane HS Entry: Data Lane TLPX Value**

Parameter	Default	User defined mode
Vertical Scale (mV)	200	50, 100, 200
Vertical Position for Data (div)	-2.6	+3 to -3
Vertical Position for Clock (div)	-2.6	+3 to -3

**Table 26: Configuration parameters for Data Lane HS Entry: Data Lane TLPX Value (cont.)**

Parameter	Default	User defined mode
Vertical Offset (V)	0	Do not change
Horizontal Scale (us)	200	100, 200, 300
Sample Rate (GS/s)	12.5	7.5, 10, 12.5
Record Length	1000000	200000, 500000, 1000000

4. Click **Analyze**. View and change the following analysis parameters:

**Table 27: Analysis parameters for Data Lane HS Entry: Data Lane TLPX Value**

Parameter	Default	User defined mode
Trigger Type	Transition	Transition
Trigger Source	Ch1	Ch1, Ch2, Ch3, Ch4
Trigger Upper Level (V)	1	Enter a value based on the test
Trigger Lower Level (V)	0	0.05, 0.1, 1
Trigger Time (pS)	500	250, 500
Trigger Transition	Greater Than	Greater Than, Less Than
Trigger Slope	Positive	Positive, Negative, Either
Trigger If Violation	Occurs	Occurs, Logic

5. Click **Limits** to view and change the Data Lane HS Entry DATA Lane TLP value and the compare string.
6. Click **Comments** to enter comments. The comments are shown in the test report.
7. Click **Apply** to effect the changes, and then click **Close**.
8. In the Select panel, click **Run** to start the test. The Acquire panel shows the progress and status of the test as it runs.

## Connections for Data Lane HS Entry: THS-PREPARE Value

[Click here](#) for information on connections for the Data Lane HS Entry: THS-PREPARE Value test.

## Configure and Run Data Lane HS Entry: THS-PREPARE Value

1. In the Select panel, select Device as **D-PHY** and Test Suite as **Transmitter**. The Version default value is **CTS 0.98 (Base Spec 1.0)**.
2. Select the test and click **Configure**.

3. In the Configuration Panel, set the following parameters for the test. The following table lists the compliance mode values and the permitted range of values for the user-defined mode:

**Table 28: Configuration parameters for Data Lane HS Entry: THS-PREPARE Value**

Parameter	Default	User defined mode
Vertical Scale (mV)	200	50, 100, 200
Vertical Position for Data (div)	-2.6	+3 to -3
Vertical Position for Clock (div)	-2.6	+3 to -3
Vertical Offset (V)	0	Do not change
Horizontal Scale (us)	200	100, 200, 300
Sample Rate (GS/s)	12.5	7.5, 10, 12.5
Record Length	1000000	200000, 500000, 1000000

4. Click **Analyze**. View and change the following analysis parameters:

**Table 29: Analysis parameters for Data Lane HS Entry: THS-PREPARE Value**

Parameter	Default	User defined mode
Trigger Type	Transition	Transition
Trigger Source	Ch1	Ch1, Ch2, Ch3, Ch4
Trigger Upper Level (V)	1	Enter a value based on the test
Trigger Lower Level (V)	0	0.05, 0.1, 1
Trigger Time (pS)	500	250, 500
Trigger Transition	Greater Than	Greater Than, Less Than
Trigger Slope	Positive	Positive, Negative, Either
Trigger If Violation	Occurs	Occurs, Logic

5. Click **Limits** to view and change the Data Lane HS Entry: THS-PREPARE values in ns, and the compare string.
6. Click **Comments** to enter comments. The comments are shown in the test report.
7. Click **Apply** to effect the changes, and then click **Close**.
8. In the Select panel, click **Run** to start the test. The Acquire panel shows the progress and status of the test as it runs.

## Connections for Data Lane HS Entry: THS-PREPARE + THS-ZERO Value

[Click here](#) for information on connections for the Data Lane HS Entry: THS-PREPARE + THS-ZERO Value test.

## Configure and Run Data Lane HS Entry: THS-PREPARE + THS-ZERO Value

1. In the Select panel, select Device as **D-PHY** and Test Suite as **Transmitter**. The Version default value is **CTS 0.98 (Base Spec 1.0)**.
2. Select the test and click **Configure**.
3. In the Configuration Panel, set the following parameters for the test. The following table lists the compliance mode values and the permitted range of values for the user-defined mode:

**Table 30: Configuration parameters for Data Lane HS Entry: THS-PREPARE + THS-ZERO**

Parameter	Default	User defined mode
Vertical Scale (mV)	200	50, 100, 200
Vertical Position for Data (div)	-2.6	+3 to -3
Vertical Position for Clock (div)	-2.6	+3 to -3
Vertical Offset (V)	0	Do not change
Horizontal Scale (us)	200	100, 200, 300
Sample Rate (GS/s)	12.5	7.5, 10, 12.5
Record Length	1000000	200000, 500000, 1000000

4. Click **Analyze**. View and change the following analysis parameters:

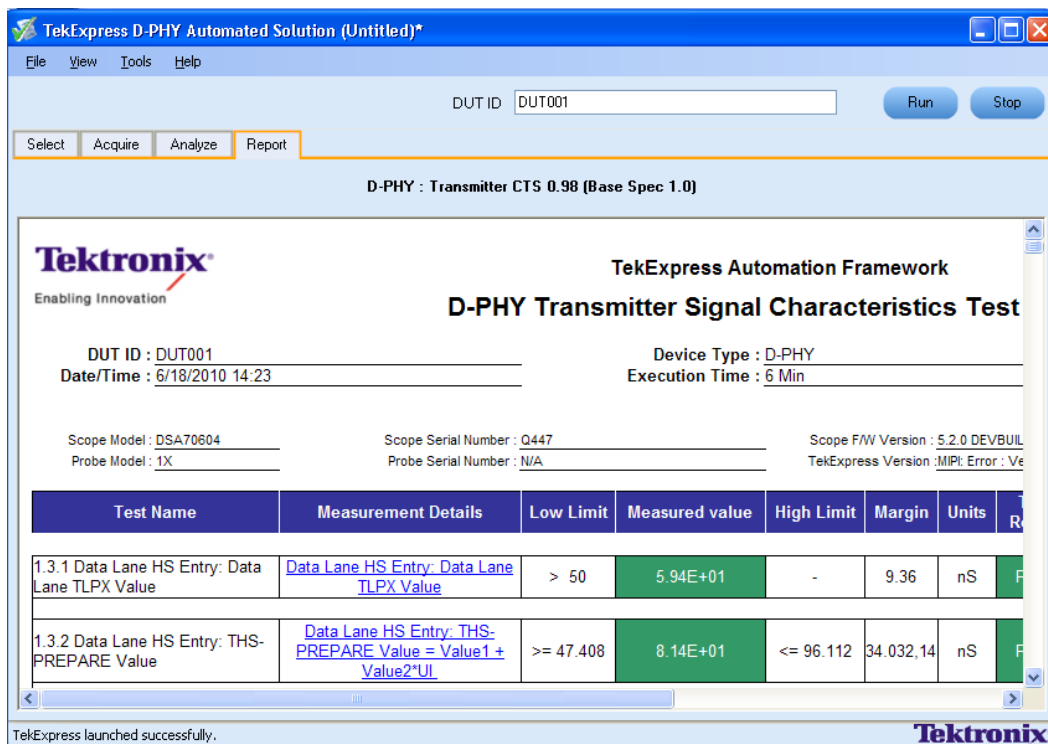
**Table 31: Analysis parameters for Data Lane HS Entry: THS-PREPARE + THS-ZERO**

Parameter	Default	User defined mode
Trigger Type	Transition	Transition
Trigger Source	Ch1	Ch1, Ch2, Ch3, Ch4
Trigger Upper Level (V)	1	Enter a value based on the test
Trigger Lower Level (V)	0	0.05, 0.1, 1
Trigger Time (pS)	500	250, 500
Trigger Transition	Greater Than	Greater Than, Less Than
Trigger Slope	Positive	Positive, Negative, Either
Trigger If Violation	Occurs	Occurs, Logic

5. Click **Limits** to view and change the Data Lane HS Entry: THS-PREPARE + THS-ZERO values in ns, and the compare string.
6. Click **Comments** to enter comments. The comments are shown in the test report.
7. Click **Apply** to effect the changes, and then click **Close**.
8. In the Select panel, click **Run** to start the test. The Acquire panel shows the progress and status of the test as it runs.

## View Report for Group 3 Tests

The application automatically displays a report in the Report panel once the test is successfully completed. A table shows the test name, measurement details, test result (pass/fail), compliance mode, and analysis time. Save the report as an MHT file using the **File > Save As** option. The following screen shows the results for Group 3 tests.



## Connections for HS Entry TCLK PREValue

[Click here](#) for information on connections for the HS Entry TCLK PREValue test.

## Configure and Run HS Entry TCLK PREValue

1. In the Select panel, select Device as **D-PHY** and Test Suite as **Transmitter**. The Version default value is **CTS 0.98 (Base Spec 1.0)**.
2. Select the test and click **Configure**.
3. In the Configuration Panel, set the following parameters for the test. The following table lists the compliance mode values and the permitted range of values for the user-defined mode:

**Table 32: Configuration parameters for HS Entry TCLK PREValue**

Parameter	Default	User defined mode
Vertical Scale (mV)	200	50, 100, 200
Vertical Position for Data (div)	-2.6	+3 to -3
Vertical Position for Clock (div)	-2.6	+3 to -3
Vertical Offset (V)	0	Do not change
Horizontal Scale (us)	8	8, 10, 12
Sample Rate (GS/s)	12.5	7.5, 10, 12.5
Record Length	1000000	200000, 500000, 1000000

4. Click **Analyze**. View and change the following analysis parameters:

**Table 33: Analysis parameters for HS Entry TCLK PREValue**

Parameter	Default	User defined mode
Trigger Type	Transition	Transition
Trigger Source	Ch3	Ch1, Ch2, Ch3, Ch4
Trigger Upper Level (V)	1	Enter a value based on the test
Trigger Lower Level (V)	0	0.05, 0.1, 1
Trigger Time (pS)	500	250, 500
Trigger Transition	Greater Than	Greater Than, Less Than
Trigger Slope	Positive	Positive, Negative, Either
Trigger If Violation	Occurs	Occurs, Logic

5. Click **Limits** to view and change the HS Entry TCLK PREValue in nS, and the compare string.
6. Click **Comments** to enter comments. The comments are shown in the test report.



7. Click **Apply** to effect the changes, and then click **Close**.
8. In the Select panel, click **Run** to start the test. The Acquire panel shows the progress and status of the test as it runs.

## Connections for HS Exit TCLK POSTValue

[Click here](#) for information on connections for the HS Exit TCLK POSTValue test.

## Configure and Run HS Entry TCLK POSTValue

1. In the Select panel, select Device as **D-PHY** and Test Suite as **Transmitter**. The Version default value is **CTS 0.98 (Base Spec 1.0)**.
2. Select the test and click **Configure**.
3. In the Configuration Panel, set the following parameters for the test. The following table lists the compliance mode values and the permitted range of values for the user-defined mode:

**Table 34: Configuration parameters for HS Exit TCLK POSTValue**

Parameter	Default	User defined mode
Vertical Scale (mV)	200	50, 100, 200
Vertical Position for Data (div)	-2.6	+3 to -3
Vertical Position for Clock (div)	-2.6	+3 to -3
Vertical Offset (V)	0	Do not change
Horizontal Scale (ns)	200	100, 200, 300
Sample Rate (GS/s)	12.5	7.5, 10, 12.5
Record Length	1000000	200000, 500000, 1000000

4. Click **Analyze**. View and change the following analysis parameters:

**Table 35: Analysis parameters for HS Exit: THS-POST Value**

Parameter	Default	User defined mode
Trigger Type	Transition	Transition
Trigger Source	Ch3	Ch1, Ch2, Ch3, Ch4
Trigger Upper Level (V)	1	Enter a value based on the test
Trigger Lower Level (V)	0	0.05, 0.1, 1
Trigger Time (pS)	500	250, 500
Trigger Transition	Greater Than	Greater Than, Less Than

**Table 35: Analysis parameters for HS Exit: THS-POST Value (cont.)**

Parameter	Default	User defined mode
Trigger Slope	Positive	Positive, Negative, Either
Trigger If Violation	Occurs	Occurs, Logic

5. Click **Limits** to view and change the HS Exit TCLK POSTValue in nS, and the compare string.
6. Click **Comments** to enter comments. The comments are shown in the test report.
7. Click **Apply** to effect the changes, and then click **Close**.
8. In the Select panel, click **Run** to start the test. The Acquire panel shows the progress and status of the test as it runs.

## Connections for Clock Rising Edge Alignment to First Payload Bit

[Click here](#) for information on connections for the Clock Rising Edge Alignment to First Payload Bit test.

## Configure and Run Clock Rising Edge Alignment to First Payload Bit

1. In the Select panel, select Device as **D-PHY** and Test Suite as **Transmitter**. The Version default value is **CTS 0.98 (Base Spec 1.0)**.
2. Select the test and click **Configure**.
3. In the Configuration Panel, set the following parameters for the test. The following table lists the compliance mode values and the permitted range of values for the user-defined mode:

**Table 36: Configuration parameters for Clock Rising Edge Alignment to First Payload Bit**

Parameter	Default	User defined mode
Vertical Scale (mV)	200	50, 100, 200
Vertical Position for Data (div)	-2.6	+3 to -3
Vertical Position for Clock (div)	-2.6	+3 to -3
Vertical Offset (V)	0	Do not change
Horizontal Scale (us)	200	100, 200, 300
Sample Rate (GS/s)	12.5	7.5, 10, 12.5
Record Length	1000000	200000, 500000, 1000000

4. Click **Analyze**. View and change the following analysis parameters:

**Table 37: Analysis parameters for Clock Rising Edge Alignment to First Payload Bit**

Parameter	Default	User defined mode
Trigger Type	Transition	Transition
Trigger Source	Ch3	Ch1, Ch2, Ch3, Ch4
Trigger Upper Level (V)	1	Enter a value based on the test
Trigger Lower Level (V)	0	0.05, 0.1, 1
Trigger Time (pS)	500	250, 500
Trigger Transition	Greater Than	Greater Than, Less Than
Trigger Slope	Positive	Positive, Negative, Either
Trigger If Violation	Occurs	Occurs, Logic

---

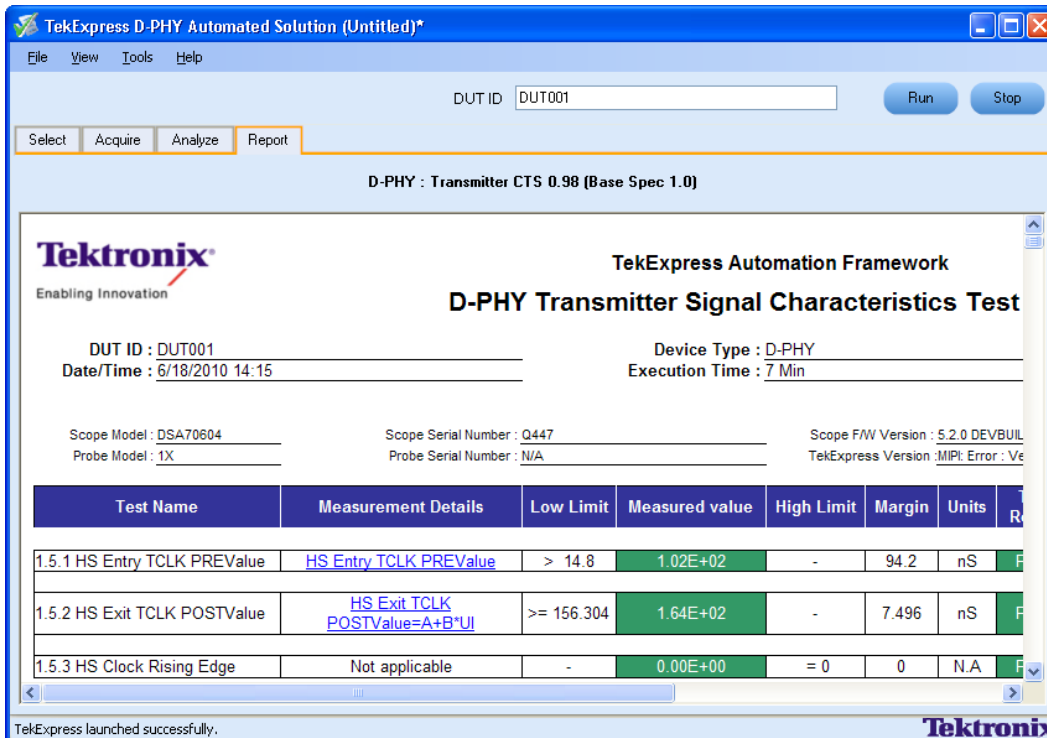
**NOTE.** *The Limits tab has no configurable parameters.*

---

5. Click **Comments** to enter comments. The comments are shown in the test report.
6. Click **Apply** to effect the changes, and then click **Close**.
7. In the Select panel, click **Run** to start the test. The Acquire panel shows the progress and status of the test as it runs.
8. Follow the on-screen prompt and complete the test.

## View Report for Group 5 Tests

The application automatically displays a report in the Report panel once the test is successfully completed. A table shows the test name, measurement details, test result (pass/fail), compliance mode, and analysis time. Save the report as an MHT file using the **File > Save As** option. The following screen shows the results for Group 5 tests.



## Performing Analysis Using Prerecorded Waveforms

1. Open the session that you have saved using **File > Open Session** or **File > Recently Recalled Setup** (a recent file). A drop-down box appears showing the session name with the date details.
2. Select **Use Pre-recorded Waveform Files** in the Acquire panel and select one of the sessions listed in the drop-down box.
3. You can modify the waveform file for each lane of the tests, add or delete a lane, and even add or delete a new test to the existing test group. Once you select run, data corresponding for that run is populated (such as the selected test, the test configuration settings, and the test summary status).

To perform analysis using prerecorded waveform file, [click here \(see page 61\)](#) to view the steps.

**Perform Analysis Using Prerecorded Waveform File**

1. Create a new session, click **File > New Session**.
2. Select a test in the Select panel.
3. Save the session, click **File > Save Session**.
4. In the Acquire panel, select **Use Pre-recorded** and browse to the location where the prerecorded waveform file is located.
5. Click **Run**.



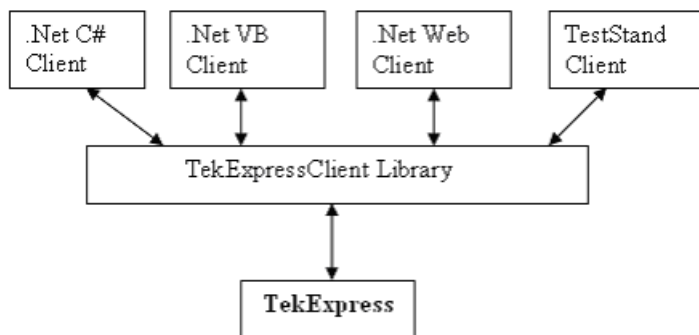
## About the Programmatic Interface

The Programmatic interface allows you to seamlessly integrate the TekExpress test automation application with the high-level automation layer. This also allows you to control the state of TekExpress application running on a local or a remote computer.

For simplifying the descriptions, the following terminologies are used in this section:

- **TekExpress Client:** A High level automation application that communicates with TekExpress using TekExpress Programmatic Interface.
- **TekExpress Server:** The TekExpress application when being controlled by TekExpress Client.

TekExpress leverages .Net Marshalling to enable the Programmatic Interface for TekExpress Client. TekExpress provides a client library for TekExpress clients to use the programmatic interface. The TekExpress client library is inherited from .Net MarshalByRef class to provide the proxy object for the clients. The TekExpress client library maintains a reference to the TekExpress Server and this reference allows the client to control the server state.



Click the following links to get details on them:

### What does one need to have to develop TekExpress Client?

While developing TekExpress Client one needs to use the TekExpressClient.dll. The client can be a VB .Net, C# .Net, TestStand or web application. The examples for interfaces in each of these applications are in Samples folder.

### References required

*TekExpressClient.dll* has internal reference to *IIdlglib.dll* and *IRemoteInterface.dll*.

*IIdlglib.dll* has a reference to *TekDotNetLib.dll*.

*IRemoteInterface.dll* provides the interfaces required to perform the remote automations. It is an interface that forms the communication line between the server and the client.

*IIdlglib.dll* provides the methods to generate and direct the secondary dialog messages at the client-end.

**NOTE.** *The end-user client application does not need any reference to above mentioned DLL files. It is essential to have these DLLs (IRemoteInterface.dll, Iidlglib.dll and TekDotNetLib.dll) in same folder location as that of TekExpressClient.dll.*

**What steps does a client need to follow?**

The following are the steps that a client needs to follow to use the TekExpressClient.dll to programmatically control the server:

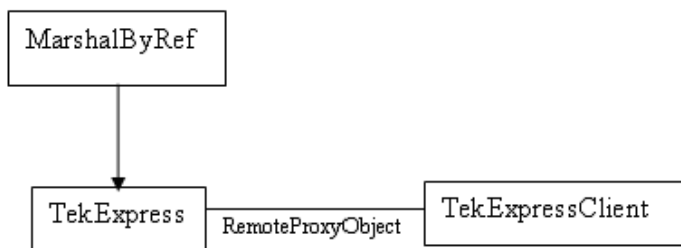
A client UI must be developed to access the interfaces exposed through the server. This client needs to load `TekExpressClient.dll` to access the interfaces. Once the `TekExpressClient.dll` is loaded, the client UI can call the specific functions to run the operations requested by the client. Once the client is up and running, it must do the following to run a remote operation:

1. The client needs to provide the IP address of the PC at which the server is running in order to connect to the server.
2. The client needs to lock the server application to avoid conflict with any other Client that may try to control the server simultaneously. “Lock” would also disable all user controls on server so that server state cannot be changed by manual operation.
3. If any other client tries to access a server which is locked, it will get a notification that the server is locked by another client.
4. When the client has connected to and locked the server, the client can access any of the programmatic controls to run the remote automations.
5. Once the client operations are completed, the server needs to be “unlocked” by the client.

## Server and Client Proxy Objects

### Remote Proxy Object

The server exposes a remote object to let the remote client access and perform the server side operations remotely. The proxy object is instantiated and exposed at the server-end through marshalling.





The following is an example:

```
RemotingConfiguration.RegisterWellKnownServiceType (typeof (TekExpressRemoteInterface), "TekExpress Remote interface", WellKnownObjectMode.Singleton);
```

This object lets the remote client access the interfaces exposed at the server side. The client gets the reference to this object when the client gets connected to the server.

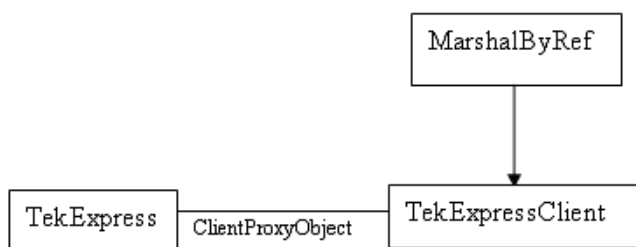
For example,

```
//Get a reference to the remote object
```

```
remoteObject = (IRemoteInterface)Activator.GetObject(typeof(IRemoteInterface), URL.ToString());
```

## Client Proxy Object

Client exposes a proxy object to receive certain information.



For example,

```
//Register the client proxy object
```

```
wellKnownServiceTypeEntry[] e = RemotingConfiguration.GetRegisteredWellKnownServiceTypes();
```

```
clientInterface = new ClientInterface();
```

```
RemotingConfiguration.RegisterWellKnownServiceType(typeof(ClientInterface), "Remote Client Interface", WellKnownObjectMode.Singleton);
```

```
//Expose the client proxy object through marshalling
```

```
RemotingServices.Marshal(clientInterface, "Remote Client Inteface");
```

The client proxy object is used for the following:

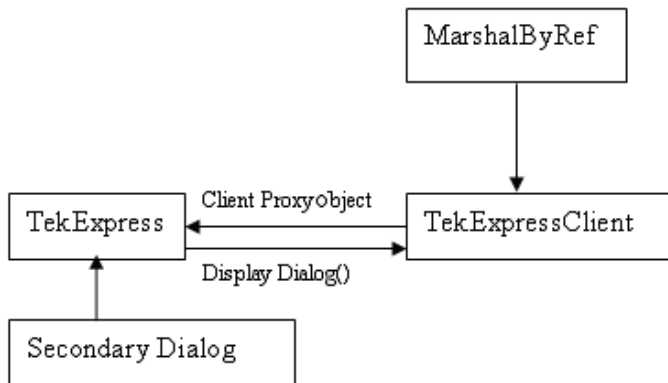
1. To get the secondary dialog messages from the server.
2. To get the file transfer commands from the server while transferring the report.

[Click here to see examples.](#)

```
clientObject.clientIntf.DisplayDialog(caption, msg, iconType, btnType);
clientObject.clientIntf.TransferBytes(buffer, read, fileLength);
```

To know more on the topics below, click the links.

**Secondary Dialog Message Handling**



The secondary dialog messages from the Secondary Dialog library are redirected to the client-end when a client is performing the automations at the remote end.

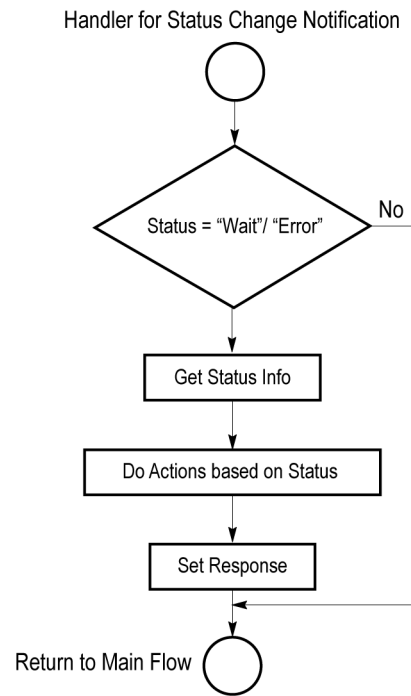
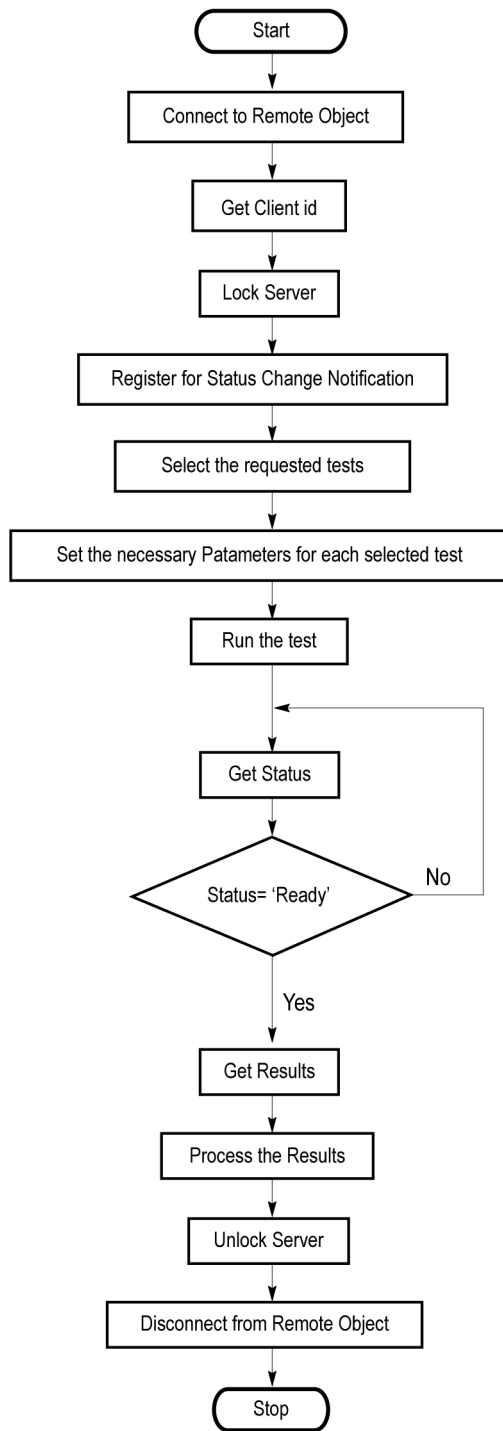
In the secondary dialog library, the assembly that is calling for the dialog box to be displayed is checked and if a remote connection is detected, the messages are directed to the remote end.

**File Transfer Events**

When the client requests the transfer of the report, the server reads the report and transfers the file by calling the file transfer methods at the client-end.

## Client Programmatic Interface: An Example

An example of the client programmatic interface is described and shown as follows:



1. Connect to a server or remote object using the programmatic interface provided.
2. Get the client ID that is created when connecting to the remote object. This client ID is one of the required parameters to communicate with the server.

---

**NOTE.** *Server identifies the client with this ID only and rejects any request if the ID is invalid.*

---

3. Lock the server for further operations. This disables the application interface.

---

**NOTE.** *You can get values from the server or set values from the server to the client only if application is locked.*

---

4. Register for receiving notifications on status change events on the server. To register you need to give a handler as a parameter.

---

**NOTE.** *Whenever there is a change in the status of the server, all the clients registered with the server receive a notification from the server.*

---

5. Select the tests that you want to run through the programmatic interface.
6. Set the necessary parameters for each test.
7. Run the tests.
8. Poll for the status of the application.

---

**NOTE.** *You can skip this step if you are registered for the status change notification and when the status is Ready.*

---

9. After completing the tests, get the results.
10. Create a report or display the results and verify or process the results.
11. Unlock the server once you complete all the tasks.
12. Disconnect from the remote object.

### Handler of Status Change Notification

1. Get the status. If the status is Wait or Error, get the information which contains the title, message description, and the expected responses for the status.
2. Perform the actions based on the status information.
3. Set the response as expected.

## D-PHY Application Command Arguments and Queries

[Connect through an IP address \(see page 75\)](#)

[Lock the server \(see page 76\)](#)

[Disable the popups \(see page 77\)](#)

[Set or get the DUT ID \(see page 78\)](#)

[Set the configuration parameters for a suite or measurement \(see page 78\)](#)

[Query the configuration parameters for a suite or measurement \(see page 80\)](#)

[Select a measurement \(see page 83\)](#)

[Select a suite \(see page 84\)](#)

[Select a channel \(see page 78\)](#)

[Configure the selected measurement \(see page 86\)](#)

[Run with set configurations or stop the run operation \(see page 88\)](#)

[Handle Error Codes \(see page 97\)](#)

[Get or set the timeout value \(see page 89\)](#)

[Wait for the measurement to complete \(see page 89\)](#)

[After the measurement is complete \(see page 91\)](#)

[Save, recall, or check if a session is saved \(see page 95\)](#)

[Unlock the server \(see page 96\)](#)

[Disconnect from server \(see page 96\)](#)

### string id

Name	Type	Direction	Description
id	string	IN	Identifier of the client that is performing the remote function.

Ready: Test configured and ready to start.

Running: Test running.

Paused: Test paused.

Wait: A popup that needs your inputs.

Error: An error is occurred.

<b>string dutName</b>			
<b>Name</b>	<b>Type</b>	<b>Direction</b>	<b>Description</b>
dutName	string	IN	The new DUT ID of the setup.

<b>out bool saved</b>			
<b>Name</b>	<b>Type</b>	<b>Direction</b>	<b>Description</b>
saved	bool	OUT	Boolean representing whether the current session is saved.

This parameter is used as a check in SaveSession() and SaveSessionAs() functions.

<b>string ipAddress</b>			
<b>Name</b>	<b>Type</b>	<b>Direction</b>	<b>Description</b>
ipAddress	string	IN	The ip address of the server to which the client is trying to connect to. This is required to establish the connection between the server and the client.

<b>out string clientID</b>			
<b>Name</b>	<b>Type</b>	<b>Direction</b>	<b>Description</b>
clientid	String	OUT	Identifier of the client that is connected to the server. clientid = unique number + ipaddress of the client. For example, 1065-192.157.98.70

**NOTE.** If the dutName parameter is null, the client is prompted to provide a valid DUT ID.

**NOTE.** The server must be active and running for the client to connect to the server. Any number of clients can be connected to the server at a time.

**NOTE.** When the client is disconnected, it is unlocked from the server and then disconnected. The id is reused.

<b>string dutId</b>			
<b>Name</b>	<b>Type</b>	<b>Direction</b>	<b>Description</b>
dutId	string	OUT	The DUT ID of the setup.

The dutId parameter is set after the server processes the request.

<b>string device</b>			
<b>Name</b>	<b>Type</b>	<b>Direction</b>	<b>Description</b>
device	string	IN	Specifies the name of the device.

<b>string suite</b>			
<b>Name</b>	<b>Type</b>	<b>Direction</b>	<b>Description</b>
suite	string	IN	Specifies the name of the suite.

<b>string test</b>			
<b>Name</b>	<b>Type</b>	<b>Direction</b>	<b>Description</b>
test	string	IN	Specifies the name of the test to obtain the pass or fail status.

<b>string parameterString</b>			
<b>Name</b>	<b>Type</b>	<b>Direction</b>	<b>Description</b>
parameterString	string	IN	Selects or deselects a test.

<b>int rowNr</b>			
<b>Name</b>	<b>Type</b>	<b>Direction</b>	<b>Description</b>
rowNr	int	IN	Specifies the zero based row index of the sub-measurement for obtaining the result value.

**NOTE.** When the client tries to lock a server that is locked by another client, the client gets a notification that the server is already locked and it must wait until the server is unlocked. If the client locks the server and is idle for a certain amount of time then the server is unlocked automatically from that client.

**out string[] status**

Name	Type	Direction	Description
status	string array	OUT	The list of status messages generated during run.

**string name**

Name	Type	Direction	Description
name	string	IN	The name of the session being recalled.

The name parameter cannot be empty. If it is empty, the client is prompted to provide a valid name.

**NOTE.** *When the run is performed, the status of the run is updated periodically using a timer.*

**string name**

Name	Type	Direction	Description
name	string	IN	The name of the session being saved.

The name parameter cannot be empty. If it is empty, the client is prompted to provide a valid name.

Once the session is saved under 'name' you cannot use this method to save the session in a different name. Use SaveSessionAs instead.

**string name**

Name	Type	Direction	Description
name	string	IN	The name of the session being recalled.

The same session is saved under different names using this method. The name parameter cannot be empty. If it is empty, the client is prompted to provide a valid name.

**bool isSelected**

Name	Type	Direction	Description
isSelected	bool	IN	Selects or deselects a test.



**string time**

Name	Type	Direction	Description
time	string	IN	The time in seconds which refers to the timeout period.

The time parameter gives the timeout period, that is the time the client is allowed to be locked and idle. After the timeout period if the client is still idle, it gets unlocked.

The time parameter should be a positive integer. Else, the client is prompted to provide a valid timeout period.

**bool\_verbose**

Name	Type	Direction	Description
_verbose	bool	IN	Specifies whether the verbose mode should be turned ON or OFF.

**NOTE.** *When the session is stopped, the client is prompted to stop the session and is stopped at the consent.*

**string filePath**

Name	Type	Direction	Description
filePath	string	IN	The location where the report must be saved in the client.

**NOTE.** *If the client does not provide the location to save the report, the report is saved at C:\ProgramFiles.*

**NOTE.** *When the client is disconnected, the client is automatically unlocked.*

**out string caption**

Name	Type	Direction	Description
caption	String	OUT	The wait state or error state message sent to you.

<b>out string message</b>			
<b>Name</b>	<b>Type</b>	<b>Direction</b>	<b>Description</b>
message	String	OUT	The wait state/error state message to you.

<b>out string[] buttonTexts</b>			
<b>Name</b>	<b>Type</b>	<b>Direction</b>	<b>Description</b>
buttonTexts	string array	OUT	An array of strings containing the possible response types that you can send.

<b>string response</b>			
<b>Name</b>	<b>Type</b>	<b>Direction</b>	<b>Description</b>
response	string	IN	A string containing the response type that you can select (it must be one of the strings in the string array buttonTexts).

<b>out string clientID</b>			
<b>Name</b>	<b>Type</b>	<b>Direction</b>	<b>Description</b>
clientID	String	OUT	Identifier of the client that is connected to the server. clientID = unique number + ipaddress of the client. For example, 1065-192.157.98.70

## Connect Through an IP Address

Command name	Parameters	Description	Return Value	Example
Connect()	<a href="#">string ipAddress (see page 70)</a> <a href="#">out string clientID (see page 70)</a>	<p>This method connects the client to the server.</p> <p><a href="#">Note (see page 70)</a></p> <p>The client provides the IP address to connect to the server.</p> <p>The server provides a unique client identification number when connected to it.</p>	Return value is either True or False.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as boolean returnval = m_Client.Connect(ipaddress,m_clientID)</pre>

---

**NOTE.** *The Fail condition for PI commands occurs in any of the following cases:*

*The server is LOCKED and the message displayed is "Server is locked by another client".*

*The session is UNLOCKED and the message displayed is "Lock Session to execute the command".*

*The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".*

*When none of these fail conditions occur, then the message displayed is "Failed...".*

---

## Lock the Server

Command name	Parameters	Description	Return Value	Example
LockSession()	<a href="#">string clientID</a> <a href="#">(see page 74)</a>	This method locks the server. <a href="#">Note (see page 71)</a> The client must call this method before running any of the remote automations. The server can be locked by only one client.	String value that gives the status of the operation after it has been performed. The return value is "Session Locked..." on success.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string returnval = m_Client.LockServer(clientID)

---

**NOTE.** *The Fail condition for PI commands occurs in any of the following cases:*

*The server is LOCKED and the message displayed is "Server is locked by another client".*

*The session is UNLOCKED and the message displayed is "Lock Session to execute the command".*

*The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".*

*When none of these fail conditions occur, then the message displayed is "Failed...".*

---

## Disable the Popups

Command name	Parameters	Description	Return Value	Example
SetVerboseMode()	<a href="#">string clientID</a> (see page 74) <a href="#">bool _verbose</a> (see page 73)	<p>This method sets the verbose mode to either true or false.</p> <p>When the value is set to true, then any of the message boxes appearing during the application will be routed to the client machine which is controlling TekExpress.</p> <p>When the value is set to false, then all the message boxes are shown on the server machine.</p>	<p>String that gives the status of the operation after it has been performed.</p> <p>When Verbose mode is set to true, the return value is "Verbose mode turned on. All dialog box will be shown to client ...".</p> <p>When Verbose mode is set to false, the return value is "Verbose mode turned off. All dialog box will be shown to server ...".</p>	<p>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string</p> <p><b>Verbose mode is turned on</b> return=m_Client.SetVerboseMode(clientID, true)</p> <p><b>Verbose mode is turned off</b> returnval=m_Client.SetVerboseMode(clientID, false)</p>

---

**NOTE.** *The Fail condition for PI commands occurs in any of the following cases:*

*The server is LOCKED and the message displayed is "Server is locked by another client".*

*The session is UNLOCKED and the message displayed is "Lock Session to execute the command".*

*The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".*

*When none of these fail conditions occur, then the message displayed is "Failed...".*

---

## Set or Get the DUT ID

Command name	Parameters	Description	Return Value	Example
SetDutId()	<a href="#">string clientID (see page 74)</a> <a href="#">string dutName (see page 70)</a>	This method changes the DUT ID of the set up. The client must provide a valid DUT ID.	String that gives the status of the operation after it has been performed. Return value is "DUT Id Changed..." on success.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string return=m_Client.SetDutId(clientID,desiredDutId)</pre> <a href="#">Note (see page 70)</a>
GetDutId()	<a href="#">string clientID (see page 74)</a> <a href="#">string dutId (see page 71)</a>	This method gets the DUT ID of the current set up.	String that gives the status of the operation after it has been performed.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string return=m_Client.GetDutId(clientID, out DutId)</pre>

---

**NOTE.** The Fail condition for PI commands occurs in any of the following cases:

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

---

## Set the Configuration Parameters for a Suite or Measurement

Command name	Parameters	Description	Return Value	Example
SetGeneralParameter()	<a href="#">string clientID (see page 74)</a> <a href="#">string device (see page 71)</a> <a href="#">string suite (see page 71)</a> <a href="#">string test (see page 71)</a> <a href="#">string parameterString (see page 71)</a>	This method sets the general parameters that are not specific to any given suite or measurement.  <b>NOTE.</b> Using this command we can select a lane, channel, or source type.	String that gives the status of the operation after it has been performed. The return value is "" (an empty String) on success.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string Select Channel (see page 79) Select Measurement Method (see page 79)</pre>

Command name	Parameters	Description	Return Value	Example
SetAnalyzeParameter()	<a href="#">string clientID (see page 74)</a> <a href="#">string device (see page 71)</a> <a href="#">string suite (see page 71)</a> <a href="#">string test (see page 71)</a> <a href="#">string parameterString (see page 71)</a>	This method sets the configuration parameters in the Analyze panel of the Configuration Panel dialog box for a given suite or measurement.	The return value is "" (an empty String) on success.	<code>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string <a href="#">Configure Analyze parameters for Data Lane Rise Time (see page 80)</a></code>
SetAcquireParameter()	<a href="#">string clientID (see page 74)</a> <a href="#">string device (see page 71)</a> <a href="#">string suite (see page 71)</a> <a href="#">string test (see page 71)</a> <a href="#">string parameterString (see page 71)</a>	This method sets the configuration parameters in the Acquire panel of the Configuration Panel dialog box for a given suite or measurement.	<code>returnVal = remoteObject.SetAcquireParameter(id, device, suite, test, parameterString) if ((OP_STATUS) returnVal != OP_STATUS.SUCCESS) return CommandFailed(returnVal)</code>	<a href="#">Configure Acquire Parameters for Data Lane Rise Time (see page 87)</a>

**NOTE.** The Fail condition for PI commands occurs in any of the following cases:

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

#### Select Channel Example

```
returnval=mClient.SetGeneralParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Data + Signal connected to Real Time Scope at$Channel 1")
```

#### Select Measurement Method Example

```
returnval=mClient.SetGeneralParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Measurement Method$Automatic")
```

Configure Analyze parameters for Data Lane Rise Time Example	
Parameter	Example
Trigger Type	returnval=mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Type\$Transition")
Trigger Source	returnval=mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Source\$Ch1")
Trigger Upper Level (V)	returnval=mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Upper Level (V)\$1")
Trigger Lower Level (V)	returnval=mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Lower Level (V)\$0.05")
Trigger Time (pS)	returnval=mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "TriggerTime (pS)\$500")
Trigger Transition	returnval=mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Transition\$Greater Than")
Trigger Slope	returnval=mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Slope\$Positive")
Trigger If Violation	returnval=mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger If Violation\$Occurs")

## Query the Configuration Parameters for a Suite or Measurement

Command name	Parameters	Description	Return Value	Example
GetGeneralParameter()	<a href="#">string clientID (see page 74)</a> <a href="#">string device (see page 71)</a> <a href="#">string suite (see page 71)</a> <a href="#">string test (see page 71)</a> <a href="#">string parameterString (see page 71)</a>	This method gets the general configuration parameters for a given suite or measurement.	The return value is the general configuration parameter for a given suite or measurement that is set.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string <a href="#">Query Channel (see page 81)</a> <a href="#">Query Measurement Method (see page 81)</a></pre>



Command name	Parameters	Description	Return Value	Example
GetAnalyzeParameter()	<a href="#">string clientID (see page 74)</a> <a href="#">string device (see page 71)</a> <a href="#">string suite (see page 71)</a> <a href="#">string test (see page 71)</a> <a href="#">string parameterString (see page 71)</a>	This method gets the configuration parameters set in the Analyze panel of the Configuration Panel dialog box for a given suite or measurement.	The return value is the configuration parameter set in the Analyze panel of the Configuration Panel dialog box for a given suite or measurement.	<code>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string</code> <a href="#">Query Analyze Parameters for Data Lane Rise Time (see page 82)</a>
GetAcquireParameter()	<a href="#">string clientID (see page 74)</a> <a href="#">string device (see page 71)</a> <a href="#">string suite (see page 71)</a> <a href="#">string test (see page 71)</a> <a href="#">string parameterString (see page 71)</a>	This method gets the configuration parameters set in the Acquire panel for a given suite or measurement.	The return value is the configuration parameter set in the Acquire panel for a given suite or measurement.	<a href="#">Query Acquire Parameters for Data Lane Rise Time (see page 82)</a>

**NOTE.** The Fail condition for PI commands occurs in any of the following cases:

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

#### Query Channel Example

```
returnval=mClient.GetGeneralParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Data + Signal connected to Real Time Scope at")
```

#### Query Measurement Method Example

```
returnval=mClient.GetGeneralParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Measurement Method")
```

<b>Query Data Lane Rise Time Acquire Parameter Examples</b>	
<b>Parameter</b>	<b>Example</b>
Vertical Scale (mV)	<code>returnval = mClient.GetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Data Lane Rise Time\$Vertical Scale (mV)")</code>
Vertical Position (div)	<code>returnval = mClient.GetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Data Lane Rise Time\$Vertical Position (div)")</code>
Vertical Offset (V)	<code>returnval = mClient.GetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Data Lane Rise Time\$Vertical Offset (V)")</code>
Horizontal Scale (us)	<code>returnval = mClient.GetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Data Lane Rise Time\$Horizontal Scale (us)")</code>
Sample Rate (GS/s)	<code>returnval = mClient.GetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Data Lane Rise Time\$Sample Rate (GS/s)")</code>
Record Length	<code>returnval = mClient.GetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Data Lane Rise Time\$Record Length")</code>
<b>Query Data Lane Rise Time Analyze Parameter Examples</b>	
<b>Parameter</b>	<b>Example</b>
Trigger Type	<code>returnval = mClient.GetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Type")</code>
Trigger Source	<code>returnval = mClient.GetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Source")</code>
Trigger Upper Level (V)	<code>returnval = mClient.GetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Upper Level (V)")</code>
Trigger Lower Level (V)	<code>returnval = mClient.GetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Lower Level (V)")</code>
Trigger Time (ps)	<code>returnval = mClient.GetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Time (pS)")</code>
Trigger Transition	<code>returnval = mClient.GetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Transition")</code>
Trigger Slope	<code>returnval = mClient.GetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Slope")</code>
Trigger If Violation	<code>returnval = mClient.GetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger If Violation")</code>

## Select a Measurement

Command name	Parameters	Description	Return Value	Example
SelectTest()	<a href="#">string clientID (see page 74)</a> <a href="#">string device (see page 71)</a> <a href="#">string suite (see page 71)</a> <a href="#">string test (see page 71)</a> <a href="#">bool isSelected (see page 72)</a>	<p>This method selects or deselects a given test.</p> <p>Setting parameter isSelected to true, you can select a measurement.</p> <p>Setting parameter isSelected to false, you can deselect a measurement.</p>	<p>String that displays the status of the operation after it has been performed.</p> <p>The return value is "" (an empty String) on success.</p>	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string <b>Select Measurement (Data Lane Rise Time):</b> returnval=m_Client.Se- lectTest(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", true)</pre>

---

**NOTE.** *The Fail condition for PI commands occurs in any of the following cases:*

*The server is LOCKED and the message displayed is "Server is locked by another client".*

*The session is UNLOCKED and the message displayed is "Lock Session to execute the command".*

*The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".*

*When none of these fail conditions occur, then the message displayed is "Failed...".*

---

## Select a Suite

Command name	Parameters	Description	Return Value	Example
SelectSuite()	<a href="#">string clientID (see page 74)</a> <a href="#">string device (see page 71)</a> <a href="#">string suite (see page 71)</a> <a href="#">bool isSelected (see page 72)</a>	<p>This method selects or deselects a given suite.</p> <p>Setting parameter isSelected to true, you can select a suite.</p> <p>Setting parameter isSelected to false, you can deselect a suite.</p>	<p>String that gives the status of the operation after it has been performed.</p> <p>The return value is "" (an empty String) on success.</p>	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string <b>Select Suite (Default):</b> returnval=m_Client.Select- Suite(clientID, "D-PHY", "Trans- mitter", true)</pre>

---

**NOTE.** *The Fail condition for PI commands occurs in any of the following cases:*

*The server is LOCKED and the message displayed is "Server is locked by another client".*

*The session is UNLOCKED and the message displayed is "Lock Session to execute the command".*

*The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".*

*When none of these fail conditions occur, then the message displayed is "Failed...".*

---

## Select a Channel

Command name	Parameters	Description	Return Value	Example
SetGeneralParameter()	<a href="#">string clientID (see page 74)</a> <a href="#">string device (see page 71)</a> <a href="#">string suite (see page 71)</a> <a href="#">string test (see page 71)</a> <a href="#">string parameterString (see page 71)</a>	This method sets the parameters that are not specific to any given test.	The return value is "" (an empty String) on success.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL

---

**NOTE.** *The Fail condition for PI commands occurs in any of the following cases:*

*The server is LOCKED and the message displayed is "Server is locked by another client".*

*The session is UNLOCKED and the message displayed is "Lock Session to execute the command".*

*The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".*

*When none of these fail conditions occur, then the message displayed is "Failed...".*

---

## Configure the Selected Measurement

Command name	Parameters	Description	Return Value	Example
SetAnalyzeParameter()	<a href="#">string clientID (see page 74)</a> <a href="#">string device (see page 71)</a> <a href="#">string suite (see page 71)</a> <a href="#">string test (see page 71)</a> <a href="#">string parameterString (see page 71)</a>	This method sets the Analyze parameters (Configuration parameters) for a given test.	The return value is "" (an empty String) on success.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string</pre> <a href="#">Configure Analyze Parameters for Data Lane Rise Time (see page 87)</a>
SetAcquireParameter()	<a href="#">string clientID (see page 74)</a> <a href="#">string device (see page 71)</a> <a href="#">string suite (see page 71)</a> <a href="#">string test (see page 71)</a> <a href="#">string parameterString (see page 71)</a>	This method sets the Acquire parameters in the Acquire panel of the Configure Dialog box for a given test.	<pre>returnVal = remoteObject.SetAcquireParameter(id, device, suite, test, parameterString) if ((OP_STATUS) returnVal != OP_STATUS.SUCCESS) return CommandFailed(returnVal)</pre>	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string</pre> <a href="#">Configure Acquire Parameters for Data Lane Rise Time (see page 87)</a>

**NOTE.** The Fail condition for PI commands occurs in any of the following cases:

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

<b>Data Lane Rise Time Configure Parameter Example</b>	
<b>Parameter</b>	<b>Example</b>
Vertical Scale (mV)	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Data Lane Rise Time\$Vertical Scale (mV)\$200")
Vertical Position (div)	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Data Lane Rise Time\$Vertical Position (div)\$-2.6")
Vertical Offset (V)	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Data Lane Rise Time\$Vertical Offset (V)\$0")
Horizontal Scale (us)	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Data Lane Rise Time\$Horizontal Scale (us)\$8")
Sample Rate (GS/s)	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Data Lane Rise Time\$Sample Rate (GS/s)\$12.5")
Record Length	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Data Lane Rise Time\$Record Length\$500000")
<b>Data Lane Rise Time Analyze Parameter Example</b>	
<b>Parameter</b>	<b>Example</b>
Trigger Type	returnval=mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Type\$Transition")
Trigger Source	returnval=mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Source\$Ch1")
Trigger Upper Level (V)	returnval=mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Upper Level (V)\$1")
Trigger Lower Level (V)	returnval=mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Lower Level (V)\$0.05")
Trigger Time (pS)	returnval=mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "TriggerTime (pS)\$500")
Trigger Transition	returnval=mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Transition\$Greater Than")
Trigger Slope	returnval=mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Slope\$Positive")
Trigger If Violation	returnval=mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger If Violation\$Occurs")

## Run with Set Configurations or Stop the Run Operation

Command name	Parameters	Description	Return Value	Example
Run()	<a href="#">string clientID</a> (see page 74)	Runs the selected tests. <a href="#">Note (see page 72)</a> Once the server is set up and is configured, it can be run remotely using this function.	String that gives the status of the operation after it has been performed. The return value is "Run started..." on success.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string returnval=m_Client.Run(clientID)</pre>
Stop()	<a href="#">string clientID</a> (see page 74)	Stops the currently running tests. <a href="#">Note (see page 73)</a>	String that gives the status of the operation after it has been performed. The return value is "Stopped..." on success.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string returnval=m_Client.Stop(clientID)</pre>

---

**NOTE.** The Fail condition for PI commands occurs in any of the following cases:

*The server is LOCKED and the message displayed is "Server is locked by another client".*

*The session is UNLOCKED and the message displayed is "Lock Session to execute the command".*

*The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".*

*When none of these fail conditions occur, then the message displayed is "Failed...".*

---



## Get or Set the Timeout Value

Command name	Parameters	Description	Return Value	Example
GetTimeOut()	<a href="#">string clientID</a> (see page 74)	Returns the current timeout period set by the client.	String that gives the status of the operation after it has been performed. The default return value is 1800000.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string returnval=m_Client.GetTimeOut()
SetTimeOut()	<a href="#">string clientID</a> (see page 74) <a href="#">string time</a> (see page 73)	Sets a timeout period specified by client. After expiry of this timeout period, the server is automatically unlocked.	String that gives the status of the operation after it has been performed. On success the return value is "TimeOut Period Changed".	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string returnval=m_Client.SetTimeOut(clientID, desiredTimeOut)

**NOTE.** The Fail condition for PI commands occurs in any of the following cases:

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

## Wait for the Test to Complete

The commands in this group are executed while tests are running. The GetCurrentStateInfo() and SendResponse() commands are executed when application is running and in wait state.

Command name	Parameters	Description	Return Value	Example
ApplicationStatus()	<a href="#">string clientID</a> (see page 74)	This method gets the status of the server application. The states at a given time are <a href="#">Ready</a> , <a href="#">Running</a> , <a href="#">Paused</a> , <a href="#">Wait</a> , or <a href="#">Error</a> . (see page 69)	String value that gives the status of the server application.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string returnval=m_Client.ApplicationStatus(clientID)

Command name	Parameters	Description	Return Value	Example
QueryStatus()	<a href="#">string clientID (see page 74)</a> <a href="#">out string[] status (see page 72)</a>	It is an interface for the user to transfer Analyze panel status messages from the server to the client.	String that gives the status of the operation after it has been performed. On success the return value is "Transferred...".	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnVal as string returnVal=m_Client.QueryStatus(clientID, out statusMessages) if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)     return "Status updated..." else     return CommandFailed(returnVal)</pre>
GetCurrentStateInfo() <b>NOTE.</b> This command is used when the application is running and is in the wait or error state.	<a href="#">string clientID (see page 74)</a> <a href="#">out string caption (see page 73)</a> <a href="#">out string message (see page 74)</a> <a href="#">out string[] buttonTexts (see page 74)</a>	This method gets the additional information of the states when the application is in Wait or Error state. Except client ID, all the others are out parameters.	This command does not return any value. This function fills up the out parameters that are passed when invoking this function.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL m_Client.GetCurrentStateInfo(clientID, caption,message, buttonTexts)</pre>
SendResponse() <b>NOTE.</b> This command is used when the application is running and is in the wait or error state.	<a href="#">string clientID (see page 74)</a> <a href="#">out string caption (see page 73)</a> <a href="#">out string message (see page 74)</a> <a href="#">string response (see page 74)</a>	After receiving the additional information using the method GetCurrentStateInfo(), the client can decide on the response to send and send the response to the application using this function. The response should be one of the strings that was earlier received as a string array in the GetCurrentStateInfo function. The _caption and _message should match the information received earlier in the GetCurrentStateInfo function.	This command does not return any value.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL m_Client.SendResponse(clientID, caption,message, response)</pre>

---

**NOTE.** The Fail condition for PI commands occurs in any of the following cases:

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

---

## After the Test is Complete

Command name	Parameters	Description	Return Value	Example
GetPassFailStatus()	<a href="#">string clientID (see page 74)</a> <a href="#">string device (see page 71)</a> <a href="#">string suite (see page 71)</a> <a href="#">string test (see page 71)</a>	This method gets the pass or fail status of the measurement after test completion.  <b>NOTE.</b> Execute this command after completing the measurement.	String that gives the status of the operation after it has been performed.  Returns the pass or fail status in the form of a string.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string returnval=m_Client.GetPassFailStatus(clientID, device, suite, "1.1.3 Data Lane Rise Time") //Pass or Fail</pre>
GetResultsValue()	<a href="#">string clientID (see page 74)</a> <a href="#">string device (see page 71)</a> <a href="#">string suite (see page 71)</a> <a href="#">string test (see page 71)</a> <a href="#">string parameterString (see page 71)</a>	This method gets the result values of the measurement after the run.	String that gives the status of the operation after it has been performed.  Returns the result value in the form of a string.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string returnval=m_Client.GetResultsValue(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Measured Value")</pre>

Command name	Parameters	Description	Return Value	Example
GetResultsValueForSubMeasurements()	<a href="#">string clientID (see page 74)</a> <a href="#">string device (see page 71)</a> <a href="#">string suite (see page 71)</a> <a href="#">string test (see page 71)</a> <a href="#">string parameterString (see page 71)</a> <a href="#">int rowNr (see page 71)</a>	This method gets the result values for individual sub-measurements, after the run.	<p>String that gives the status of the operation after it has been performed.</p> <p>Returns the result value in the form of a string.</p>	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string <b>Data Lane Rise Time DP (nS)</b> returnval=m_Client.GetRe- sultsValueForSubMeasure- ments(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "1.1.3 Data Lane Rise Time", "Measured Value",0) //For DP wfm returnval=m_Client.GetRe- sultsValueForSubMeasure- ments(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Measured Value",1) //For DN wfm</pre>
GetReportParameter()	<a href="#">string clientID (see page 74)</a> <a href="#">string device (see page 71)</a> <a href="#">string suite (see page 71)</a> <a href="#">string test (see page 71)</a> <a href="#">string parameterString (see page 94)</a>	This method gets the general report details such as oscilloscope model, TekExpress version, and D-PHY version.	The return value is the oscilloscope model, TekExpress version, and D-PHY version.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string <b>Oscilloscope Model</b> returnval=m_Client.GetRe- portParameter(clientID,"Scope Model") <b>TekExpress Version</b> returnval=m_Client.GetReport- Parameter(clientID,"TekExpress Version") <b>D-PHY Version</b> returnval=m_Client.GetReport- Parameter(clientID,"Application Version")</pre>

Command name	Parameters	Description	Return Value	Example
TransferReport()	<a href="#">string clientID</a> (see page 74) <a href="#">string filePath</a> (see page 73)	This method transfers the report generated after the run. The report contains the summary of the run. The client must provide the location where the report is to be saved at the client-end.	String that gives the status of the operation after it has been performed. Transfers all the result values in the form of a string.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string returnval=m_Client.TransferRe- port(clientID,"C:\Report")</pre>
TransferWaveforms()	<a href="#">string clientID</a> (see page 74) <a href="#">string filePath</a> (see page 73)	This method transfers all the waveforms from the folder for the current run. <b>NOTE.</b> For each click of Run button, a folder is created in the X: drive. Transfer the waveforms before clicking the Run button.	String that gives the status of the operation after it has been performed. Transfers all the waveforms in the form of a string. On success the return value is "Transferred...".	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string returnval=m_Client.Transfer- Waveforms(clientID,"C:\Wave- forms")</pre>
TransferImages()	<a href="#">string clientID</a> (see page 74) od <a href="#">string filePath</a> (see page 73)	This method transfers all the images (screenshots) from the folder for the current run (for a given suite or measurement). <b>NOTE.</b> For each click of Run button, a folder is created in the X: drive. Transfer the waveforms before clicking the Run button.	String that gives the status of the operation after it has been performed. Transfers all the images in the form of a string.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string returnval=m_Client.TransferIm- ages(clientID, "C:\Waveforms")</pre>

---

**NOTE.** *The Fail condition for PI commands occurs in any of the following cases:*

*The server is LOCKED and the message displayed is "Server is locked by another client".*

*The session is UNLOCKED and the message displayed is "Lock Session to execute the command".*

*The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".*

*When none of these fail conditions occur, then the message displayed is "Failed...".*

---

<b>string parameterString</b>			
<b>Name</b>	<b>Type</b>	<b>Direction</b>	<b>Description</b>
parameterString	string	IN	Specifies the oscilloscope model, TekExpress version, and D-PHY version.

## Save, Recall, or Check if a Session is Saved

Command name	Parameters	Description	Return Value	Example
CheckSession-Saved()	<a href="#">string clientID</a> (see page 74) <a href="#">out bool saved</a> (see page 70)	This method is called when a check is to be made to know if the current session is saved.	Return value is either True or False.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string returnval=m_Client.CheckSessionSaved(m_clientID, out savedStatus)
RecallSession()	<a href="#">string clientID</a> (see page 74) <a href="#">string name</a> (see page 72)	Recalls a saved session. The name of the session is provided by the client.	String that gives the status of the operation after it has been performed. The return value is "Session Recalled...".	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string returnval=m_Client.RecallSession(clientID, savedSessionName)
SaveSession()	<a href="#">string clientID</a> (see page 74) <a href="#">string name</a> (see page 72)	Saves the current session. The name of the session is provided by the client.	String that gives the status of the operation after it has been performed. The return value is "Session Saved..." or "Failed...".	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string returnval=m_Client.SaveSession(clientID, desiredSessionName)
SaveSessionAs()	<a href="#">string clientID</a> (see page 74) <a href="#">string name</a> (see page 72)	Saves the current session in a different name every time this method is called. The name of the session is provided by the client.	String that gives the status of the operation after it has been performed. The return value is "Session Saved...".	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string returnval=m_Client.SaveSessionAs(clientID, desiredSessionName)

**NOTE.** The Fail condition for PI commands occurs in any of the following cases:

The server is **LOCKED** and the message displayed is "Server is locked by another client".

The session is **UNLOCKED** and the message displayed is "Lock Session to execute the command".

The server is **NOTFOUND** and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

## Unlock the Server

Command name	Parameters	Description	Return Value	Example
UnlockSession()	<a href="#">string clientID</a> (see page 74)	This method unlocks the server from the client. The ID of the client to be unlocked must be provided. <a href="#">Note (see page 73)</a>	String that gives the status of the operation after it has been performed. The return value is "Session Un-Locked...".	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string returnval=m_Client.Unlock-Server(clientID)

**NOTE.** The Fail condition for PI commands occurs in any of the following cases:

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".

## Disconnect from the Server

Command name	Parameters	Description	Return Value	Example
Disconnect()	<a href="#">string clientID</a> (see page 74)	This method disconnects the client from the server it is connected to. <a href="#">Note (see page 70)</a>	Integer value that gives the status of the operation after it has been performed. 1 for Success -1 for Failure	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string returnval=m_Client.Discon-nect(m_clientID)

**NOTE.** The Fail condition for PI commands occurs in any of the following cases:

The server is LOCKED and the message displayed is "Server is locked by another client".

The session is UNLOCKED and the message displayed is "Lock Session to execute the command".

The server is NOTFOUND and the message displayed is "Server not found...Disconnect!".

When none of these fail conditions occur, then the message displayed is "Failed...".



## Handle Error Codes

The return value of the remote automations at the server-end is `OP_STATUS` which is changed to a string value depending on its code and returned to the client. The values of `OP_STATUS` are as follows:

Value	Code	Description
FAIL	-1	The operation failed.
SUCCESS	1	The operation succeeded.
NOTFOUND	2	Server not found
LOCKED	3	The server is locked by another client, so operation cannot be performed.
UNLOCK	4	The server is not locked. Lock the server before performing the operation.
NULL	0	Nothing

## Program Example

This program example shows how to communicate between a PC and TekExpress D-PHY remotely.

A typical application does the following:

1. Start the application.

2. Connect through an IP address.

```
m_Client.Connect("localhost") 'True or False
clientID = m_Client.getClientID
```

3. Lock the server.

```
m_Client.LockServer(clientID)
```

4. Disable the Popups.

```
m_Client.SetVerboseMode(clientID, false)
```

5. Set the Dut ID.

```
m_Client.SetDutId(clientID, "DUT_Name")
```

6. Select a measurement.

```
mClient.SelectTest(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", true) 'Data Lane Rise Time measurement selected
```

7. Select a channel.

```
mClient.SetGeneralParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Data + Signal connected to Real Time Scope at$Channel 1")
Data Lane Rise Time measurement selected
```

**8.** Configure the selected measurement.

```
mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Type$Transition") 'Data Lane Rise Time measurement selected
```

```
mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Source$Ch1") 'Data Lane Rise Time measurement selected
```

```
mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Upper Level(V)$1") 'Data Lane Rise Time measurement selected
```

```
mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Lower Level(V)$0") 'Data Lane Rise Time measurement selected
```

```
mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Time(pS)$500") 'Data Lane Rise Time measurement selected
```

```
mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Transition$Greater Than") 'Data Lane Rise Time measurement selected
```

```
mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger Slope$Positive") 'Data Lane Rise Time measurement selected
```

```
mClient.SetAnalyzeParameter(clientID, "D-PHY", "Transmitter", "1.1.3 Data Lane Rise Time", "Trigger If Violation$Occurs") 'Data Lane Rise Time measurement selected
```

**9.** Run with set configurations.

```
m_Client.Run(clientID)
```

**10.** Wait for the test to complete.

```
Do
```

```
Thread.Sleep(500)
```

```
m_Client.Application_Status(clientID)
```

```
Select Case status
```

```
Case "wait"
```

```
'Get the Current State Information
```

```
mClient.GetCurrentStateInfo(clientID, waitingMsbBxCaption, waitingMsbBxMessage, waitingMsbBxButtonTexts)
```

```
'Send the Response
```

```
mClient.SendResponse(clientID, waitingMsBxCaption, waitingMsBxMessage,  
waitingMsBxResponse)
```

```
End Select
```

```
Loop Until status = "Ready"
```

**11.** After the Test is Complete.

'Save all results values from folder for current run

```
m_Client.TransferResult(clientID, logDirname)
```

'Save all waveforms from folder for current run

```
m_Client.TransferWaveforms(clientID, logDirname)
```

'Save all images from folder for current run

```
m_Client.TransferImages(clientID, logDirname)
```

**12.** Unlock the server.

```
m_Client.UnlockServer(clientID)
```

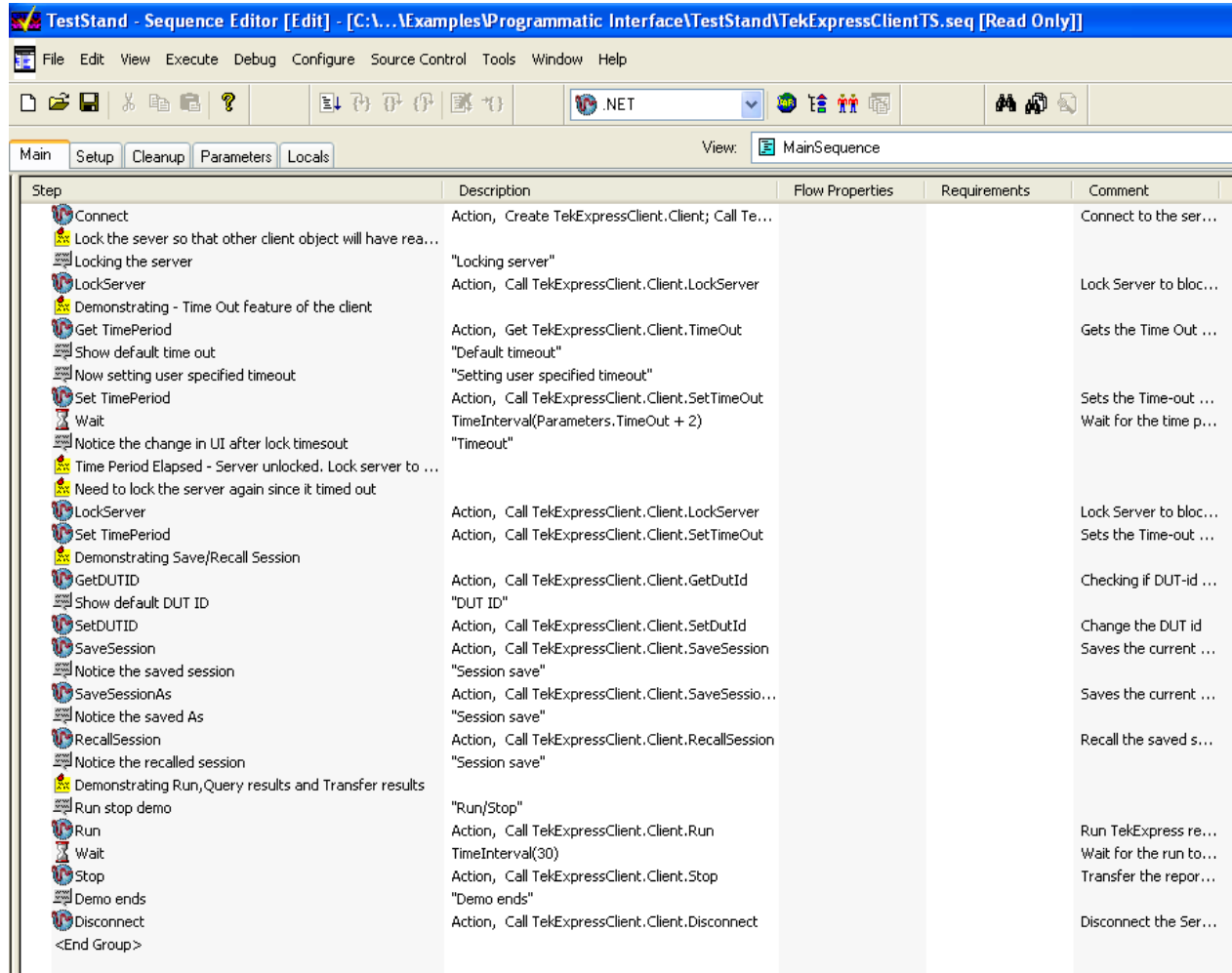
**13.** Disconnect from server.

```
m_Client.Disconnect()
```

**14.** Exit the application.

### NI TestStand Client Example

The following is an example for NI TestStand Client available in the path, C:\Program Files\Tektronix\TekExpress\TekExpress D-PHY\Examples\Programmatic Interface\TestStand



## Group 1 Tests

### Test 1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)

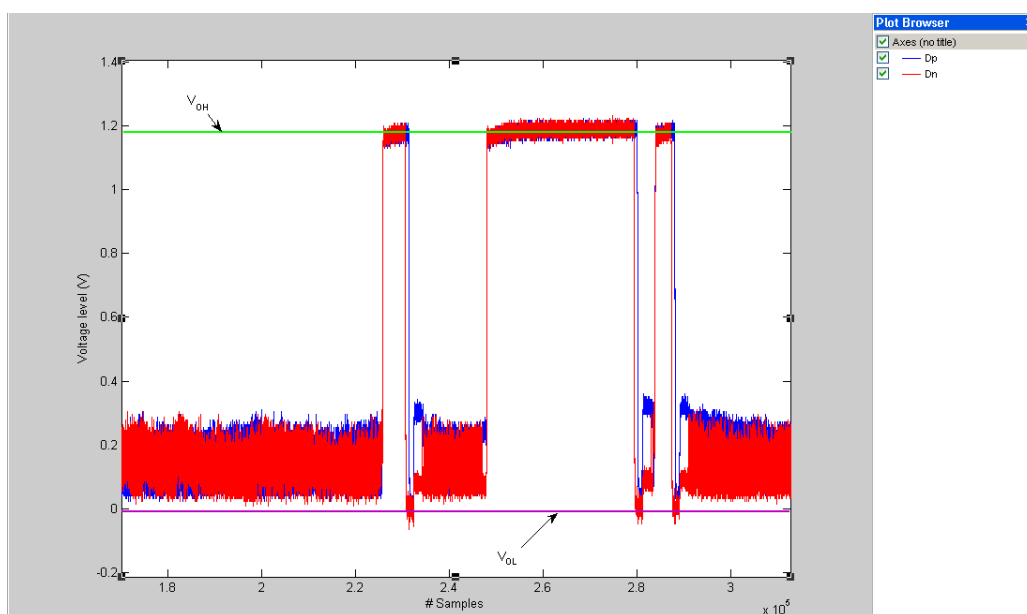
Data Lane LP-TX Thevenin Output High Level Voltage is calculated using the following algorithm:

- Acquire the Dp and Dn waveforms by setting the transition trigger from LP to HS mode (LP11-10) on the oscilloscope.
- Using post processing:
  - Traverse the Dp waveform and find the LP signaling sequence.
  - In the LP sequence region, find the absolute peak-to-peak value.  $V_{OH}$  is measured as the mode of all the waveform samples greater than 50% of the absolute peak-to-peak.

where,

Dp – Data positive waveform

Dn – Data negative waveform



### Test 1.1.2 Data Lane LP-TX Thevenin Output Low Level Voltage (VOL)

Data Lane LP-TX Thevenin Output Low Level Voltage is calculated using the following algorithm:

- Acquire the Dp and Dn waveforms by setting the transition trigger from LP to HS mode (LP11-10) on the oscilloscope.
- Using post processing:
  - Traverse the Dp waveform and find the LP signaling sequence.
  - In the LP sequence region, find the absolute peak-to-peak value.  $V_{OH}$  is measured as the mode of all the waveform samples lower than 50% of the absolute peak-to-peak.

where,

Dp – Data positive waveform

Dn – Data negative waveform

### Test 1.1.3 Data Lane Rise Time

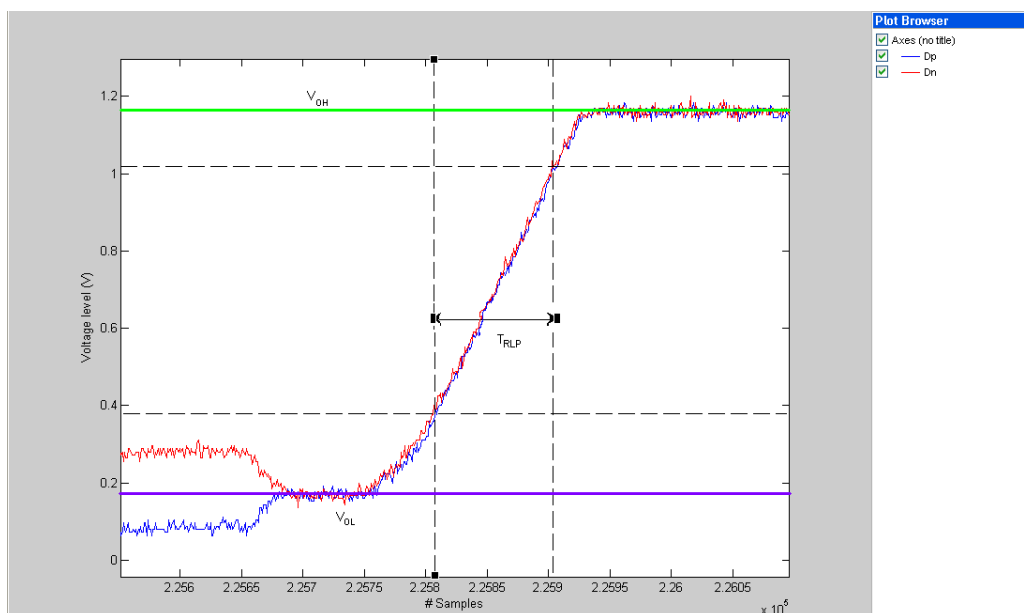
Data Lane Rise Time is calculated using the following algorithm:

- Acquire the Dp and Dn waveforms by setting the transition trigger from LP to HS mode (LP11-10) on oscilloscope.
- Using post processing:
  - Traverse the Dp waveform and find the LP signaling sequence.
  - In the LP sequence region, find the absolute peak-to-peak.  $V_{OH}$  is measured as the mode of all the waveform samples greater than 50% of the absolute peak-to-peak and  $V_{OL}$  is measured as the mode of all the waveform samples lower than 50% of the absolute peak-to-peak.
  - Find all the rising edges in the Dp waveform and find the 15%-85% rise time with reference to  $V_{OL}$  and  $V_{OH}$  for each of the edges and average the results. This result is Rise Time ( $T_{RLP}$ ).

where,

Dp – Data positive waveform

Dn – Data negative waveform



### Test 1.1.4 Data Lane Fall Time

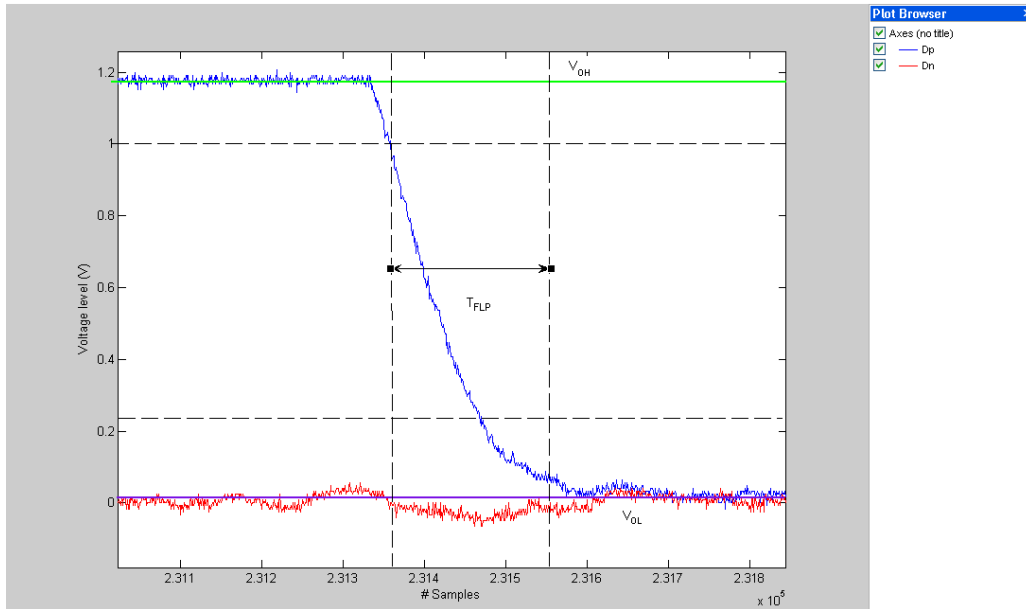
Data Lane Fall Time is calculated using the following algorithm:

- Acquire the Dp and Dn waveforms by setting the transition trigger from LP to HS mode (LP11-10) on oscilloscope.
- Using post processing:
  - Traverse the Dp waveform and find the LP signaling sequence.
  - In the LP sequence region, find the absolute peak-to-peak.  $V_{OH}$  is measured as the mode of all the waveform samples greater than 50% of the absolute peak-to-peak and  $V_{OL}$  is measured as the mode of all the waveform samples lower than 50% of the absolute peak-to-peak..
  - Find all the falling edges in the Dp waveform and find the 15%-85% fall time with reference to  $V_{OL}$  and  $V_{OH}$  for each of the edges and average the results. This result is Fall Time ( $T_{FLP}$ ).

where,

Dp – Data positive waveform

Dn – Data negative waveform



## Group 2 Tests

### Test 1.2.1 Clock Lane LP-TX Thevenin Output High Level Voltage (VOH)

Clock Lane LP-TX Thevenin Output High Level Voltage is calculated using the following algorithm:

- Acquire the Cp and Cn waveforms by setting the transition trigger from LP to HS mode (LP11-10) on oscilloscope.
- Using post processing:
  - Traverse the Cp waveform and find the LP signaling sequence.
  - In the LP-11 sequence region, find the absolute peak-to-peak.  $V_{OH}$  is measured as the mode of all the waveform samples greater than 50% of the absolute peak-to-peak.

where,

Cp – Clock positive waveform

Cn – Clock negative waveform

### Test 1.2.2 Clock Lane LP-TX Thevenin Output Low Level Voltage (VOL)

Clock Lane LP-TX Thevenin Output Low Level Voltage is calculated using the following algorithm:



- Acquire the Cp and Cn waveforms by setting the transition trigger from LP to HS mode (LP11-10) on oscilloscope.
- Using post processing:
  - Traverse the Cp waveform and find the LP signaling sequence.
  - In the LP-00 sequence region, find the absolute peak-to-peak.  $V_{OL}$  is measured as the mode of all the waveform samples lower than 50% of the absolute peak-to-peak.

where,

Cp – Clock positive waveform

Cn – Clock negative waveform

### Test 1.2.3 Clock Lane Rise Time

Clock Lane Rise Time is calculated using the following algorithm:

- Acquire the Cp and Cn waveforms by setting the transition trigger from LP to HS mode (LP11-10) on oscilloscope.
- Using post processing:
  - Traverse the Cp waveform and find the LP signaling sequence.
  - In the LP sequence region, find the absolute peak-to-peak and  $V_{OH}$  is measured as the mode of all the waveform samples greater than 50% of the absolute peak-to-peak and  $V_{OL}$  is measured as the mode of all the waveform samples lower than 50% of the absolute peak-to-peak.
  - Find all the rising edges in the Cp waveform and find the 15%-85% rise time with reference to  $V_{OL}$  and  $V_{OH}$  for each of the edges and average the results. This result is Rise Time ( $T_{RLP}$ ).

where,

Cp – Clock positive waveform

Cn – Clock negative waveform

### Test 1.2.4 Clock Lane Fall Time

Clock Lane Fall Time is calculated using the following algorithm:

- Acquire Cp and Cn waveform by setting the Transition trigger from LP to HS mode (LP11-10) on oscilloscope.
- Using post processing:
  - Traverse the Cp waveform and find the LP signaling sequence.
  - In the LP sequence region, find the absolute peak-to-peak and  $V_{OH}$  is measured as the mode of all the waveform samples greater than 50% of the absolute peak-to-peak and  $V_{OL}$  is measured as the mode of all the waveform samples lower than 50% of the absolute peak-to-peak.
  - Find all the falling edges in the Cp waveform and find the 15%-85% fall time with reference to  $V_{OL}$  and  $V_{OH}$  for each of the edges and average the results. This result is Fall Time ( $T_{FLP}$ ).

where,

Cp – Clock positive waveform

Cn – Clock negative waveform

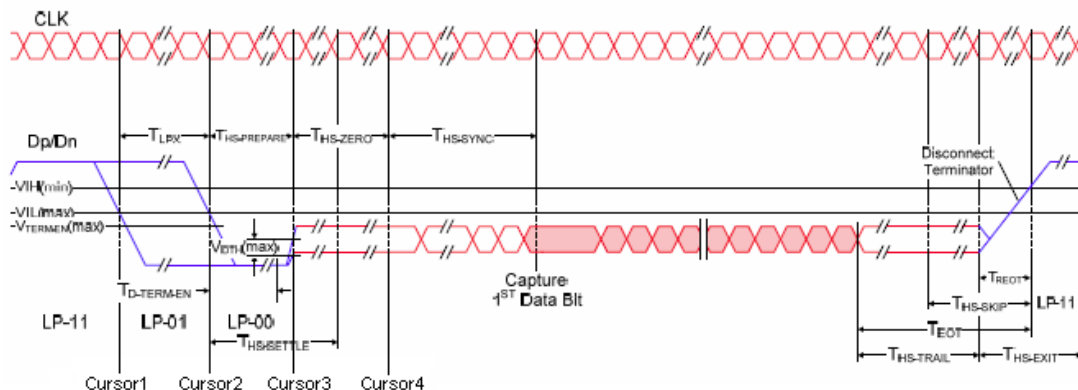
## Group 3 Tests

### Test 1.3.1 Data Lane HS Entry: Data Lane TLPX Value

Data Lane TLPX Value is calculated using the following algorithm:

- Acquire Dp and Dn waveform by setting the Transition trigger from LP to HS mode (LP11-10) on oscilloscope.
- Using post processing, traverse the Dp waveform and find the Cursor 1 and Cursor 2 positions corresponding to LP-01 signaling sequence as given:
  - Compute Dp–Dn and look for points with voltage greater than 550 mV. If the number of points found is 0, then compute Dn–Dp and look for points with voltage greater than 550 mV. If the number of points found is 0, then the signal is not correct. If the signal is correct, proceed with the following steps on the difference signal (Diff):
    - In the Diff waveform, in the region corresponding to the LP signaling sequence region, find the first rising edge and successive falling edge.
    - The time difference between these two edges at voltage level of 550 mV is  $T_{LPX}$ .
    - where,
      - Dp – Data positive waveform
      - Dn – Data negative waveform
      - Cp – Clock positive waveform

Cn – Clock negative waveform



### Test 1.3.2 Data Lane HS Entry: THS-PREPARE Value

Data Lane THS-PREPARE Value is calculated using the following algorithm:

- Acquire Dp and Dn waveform by setting the Transition trigger from LP to HS mode (LP11-10) on the oscilloscope.
- Using post processing, traverse the Dp waveform and find the Cursor2 and Cursor3 positions corresponding to LP-00 signaling sequence as given below:
  - Subtract the Dp and Dn waveforms to obtain Dp-Dn.
  - In the LP signaling sequence, in Dp-Dn find the first rising edge and successive falling edge. Cursor 2 is the point where the falling edge touches 550 mV.
- Compute V<sub>OLHS</sub>:
  - Find the high speed sequence in the input waveform.
  - Find the highest and the lowest voltage values in the high speed sequence. The mode of all the values greater than 50% of the peak-to-peak is the V<sub>OHHS</sub> and the mode of all the values less than 50% of the peak-to-peak is the V<sub>OLHS</sub>.
- Find the Cursor 3 position:
  - From the point corresponding to Cursor 2 on Dn, traverse to the right of the Dn waveform and find a point that is less than or equal to the V<sub>OLHS</sub>. Mark this point as Marker 1.
  - From Marker 1, traverse both the Dn and Dp waveforms and find the first point which is greater than or equal to V<sub>OHHS</sub> on either of Dn or Dp waveforms. Mark this point as Marker 2.
  - On the waveform where the Marker 2 point was found, traverse to the left of the waveform until a point less than or equal to 70 mV is located. This is Cursor 3.
- The time between Cursor 3 and Cursor 2 is T<sub>HS-PREPARE</sub>.

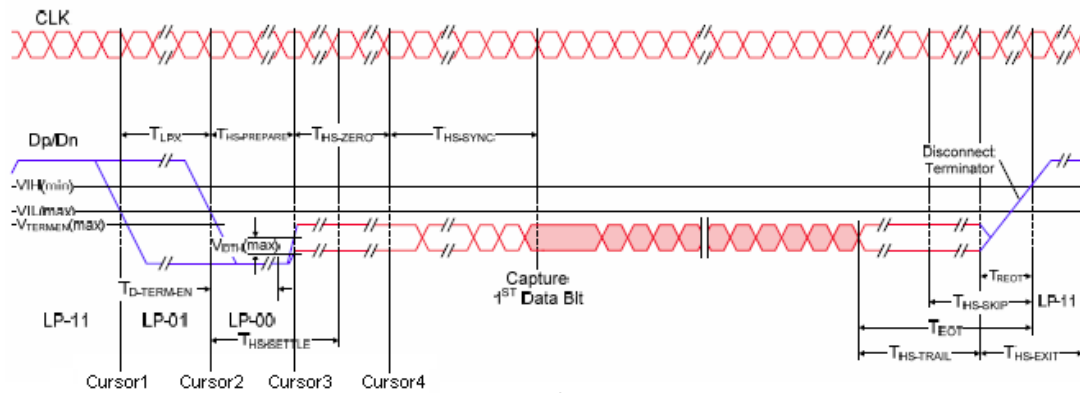
where,

Dp – Data positive waveform

Dn – Data negative waveform

Cp – Clock positive waveform

Cn – Clock negative waveform



### Test 1.3.3 Data Lane HS Entry: THS-PREPARE + THS-ZERO Value

Data Lane THS-PREPARE + THS-ZERO Value is calculated using the following algorithm:

- Acquire Dp and Dn waveform by setting the Transition trigger from LP to HS mode (LP11-10) on the oscilloscope.
- Using post processing, traverse the Dp waveform and find the Cursor 2 and Cursor 4 positions as below:
  - Subtract the Dp and Dn waveforms to obtain Dp–Dn.
  - In the LP signaling sequence, in Dp–Dn find the first rising edge and successive falling edge. Marker 1 is the point where the falling edge touches 550 mV. Cursor 2 = Marker 1.
- Compute  $V_{OLHS}$ :
  - Find the high speed sequence in the input data waveforms.
  - Find the highest and the lowest voltage values in the high speed sequence. The mode of all the values greater than 50% of the peak-to-peak is the  $V_{OHHS}$  and the mode of all the values less than 50% of the peak-to-peak is the  $V_{OLHS}$ .



## Group 5 Tests

### Test 1.5.1 HS Entry TCLK PREValue

HS Entry TCLK PREValue is calculated using the following algorithm:

- Acquire the Dp and Dn waveforms by setting the Transition trigger from LP to HS mode (LP11-10) on the oscilloscope.
- Using post processing, traverse the Cp and Cn waveforms and find the position corresponding to Cursor 3. Traverse the Dp and Dn waveforms and find the position corresponding to Cursor 4 as below:
  - Subtract the Cp and Cn waveforms to obtain Cp–Cn.
  - In the LP signaling sequence, find in Cp–Cn, the first rising edge and successive falling edge. Marker 1 is the point where the falling edge touches 550 mV.
- Compute  $V_{OLHS}$ :
  - Find the high speed sequence in the input clock waveforms.
  - Find the highest and the lowest voltage values in the high speed sequence. The mode of all the values greater than 50% of the peak-to-peak is the  $V_{OHHS}$  and the mode of all the values less than 50% of the peak-to peak is the  $V_{OLHS}$ .
- Find the Cursor 3 position.
  - From the point corresponding to Marker 1 on Cn, traverse to the right of Cn waveform and find a point that is less than or equal to the  $V_{OLHS}$ . Mark this point as Marker 2.
  - From Marker 2, traverse both the Cn and Cp waveforms and find the first point which is greater than or equal to  $V_{OHHS}$  on either of the Cn or Cp waveforms. Mark this point as Marker 3.
  - On the waveform where the Marker 3 point was found, traverse to the left of the waveform until a point less than or equal to 70 mV is located. This is Marker 4.
  - On the waveform where the point corresponding to Marker 3 was found, traverse to the right and find a point where the voltage is less than or equal to  $(V_{OHHS} + V_{OLHS})/2$ . This is Marker 5.
  - Starting from Marker 5, on the same waveform where the point corresponding to Marker 5 was found, traverse further and find a point where the voltage is greater than or equal to  $(V_{OHHS} + V_{OLHS})/2$ . This is Marker 6. Marker 5–Marker 6 and is almost 1 High speed UI. Hence Marker 5 – (Marker 6–Marker 5) is the end point of  $T_{CLK-ZERO}$ . Call this Marker 7.
  - Cursor 3 = Marker 7.
- On the Dn and Dp waveforms, find the point corresponding to Marker 7 and check if the value of that point is greater than 0.8 V. If yes, continue with the following steps, else, the signal is not correct.
- On the Data waveform, find a point corresponding to Marker 7. From this point, traverse to the right and find the first falling edge. The point at which this falling edge crosses 550 mV is Cursor 4.
- The time between Cursor 3 and Cursor 4 is  $T_{CLK-PRE}$ .



voltage level reaches  $V_{OL}$ . This point is Marker 1. At Marker 1 position, one of  $C_p$  and  $C_n$  voltages is greater than the other. Let us say  $C_n$  value is greater than  $C_p$  value at this point.

- Traverse backwards from Marker 1 and find on the clock waveforms where  $C_p$  value becomes greater than  $C_n$ . This point is Marker 2.
- Continue traversing backwards on the clock waveforms and find a point where  $C_n$  value becomes greater than  $C_p$ .
- Cursor 2 = Marker 2 + (Marker 2–Marker 3).
- At a point corresponding to Cursor 2, check if the data waveform value is greater than 0.8 V. If yes, then proceed, else the signal is not correct.
- From the point corresponding to Cursor 2 position on Data waveform, traverse backwards and find the point where a rising edge value becomes either equal to or just less than  $V_{OL}$ . This point is Cursor 1.
- The time between Cursor 1 and Cursor 2 is  $T_{CLK-POST}$  as shown in the following figure.

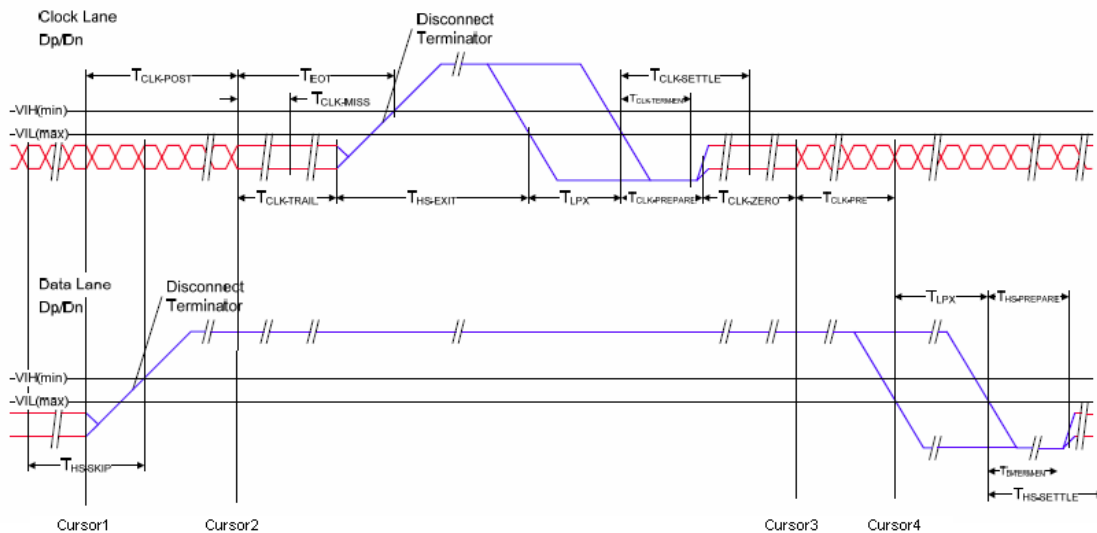
where,

$D_p$  – Data positive waveform

$D_n$  – Data negative waveform

$C_p$  – Clock positive waveform

$C_n$  – Clock negative waveform





## Test 1.5.3 HS Clock Rising Edge Alignment to First Payload Bit

HS Clock Rising Edge Alignment to First Payload Bit is determined by visual verification.

### Shortcut Keys

The following table lists the short cut keys to access the application:

**Table 38: Shortcut keys**

Menu	Shortcut keys
File	Alt + F
New Session	Ctrl + N
Open Session	Ctrl + O
Save Session	Ctrl + S
Save Session As	Alt + F + A
Save Report As	Alt + F + R Ctrl + R
Print Preview Report	Alt + F + V Ctrl+ V
Print Report	Ctrl + P
Exit	Ctrl + X
View	Alt + V
Log	Ctrl + L
Tools	Alt + T
Instrument Bench	Ctrl + I
Help	Alt + H
TekExpress Help (F1)	Alt + H + H
About TekExpress	Alt + H + A



# Index

## A

- About TekExpress, 13
- Acquire parameters, 23
- Acquire source options, 30
- Activating license, 8
- Algorithm
  - Clock Lane Fall Time, 105
  - Clock Lane LP-TX Thevenin Output High Level Voltage, 104
  - Clock Lane LP-TX Thevenin Output Low Level Voltage, 104
  - Clock Lane Rise Time, 105
  - Data Lane Fall Time, 103
  - Data Lane HS Entry: Data Lane TLPX Value, 106
  - Data Lane HS Entry: THS-PREPARE + THS-ZERO Value, 108
  - Data Lane HS Entry: THS-PREPARE Value, 107
  - Data Lane LP-TX Thevenin Output High Level Voltage, 101
  - Data Lane LP-TX Thevenin Output Low Level Voltage, 101
  - Data Lane Rise Time, 102
  - HS Clock Rising Edge Alignment to First Payload Bit, 113
  - HS Entry TCLK PREValue, 110
  - HS Exit TCLK POSTValue, 111
- Analyze parameters, 23
- Application summary, 13
- ApplicationStatus(), 89

## B

- Before clicking Run, 10

## Button

- Configure, 22
- Deselect All, 22
- Select All, 22
- Show Schematic, 22

## C

- CheckSessionSaved(), 95
- Client, 63
- Configure
  - Acquire tab, 23
  - Analyze tab, 23
  - Apply button, 24
  - Clock lane fall time, 49
  - Clock lane LP-TX Thevenin output high level voltage (VOH), 45
  - Clock lane LP-TX Thevenin output low level voltage (VOL), 47
  - Clock lane rise time, 48
  - Clock Rising Edge Alignment to First Payload Bit, 58
  - Close button, 24
  - Comments tab, 23
  - Data lane fall time, 43
  - Data Lane HS Entry: Data Lane TLPX Value, 51
  - Data Lane HS Entry: THS-PREPARE + THS-ZERO, 54
  - Data Lane HS Entry: THS-PREPARE Value, 52
  - Data lane LP-TX Thevenin output high level voltage (VOH), 39
  - Data lane LP-TX Thevenin output low level voltage (VOL), 41
  - Data lane rise time, 42
  - HS Entry TCLK PREValue, 56

- HS Exit TCLK POSTValue, 57

- Limits tab, 23
- Restore button, 23

- Configure and run the Test(s), 22
- Connect(), 75

## Connections

- Clock lane fall time, 49
- Clock lane LP-TX Thevenin output high level voltage (VOH), 45
- Clock lane LP-TX Thevenin output low level voltage (VOL), 47
- Clock lane rise time, 48
- Clock Rising Edge Alignment to First Payload Bit, 58
- Data lane fall time, 43
- Data Lane HS Entry: Data Lane TLPX Value, 51
- Data Lane HS Entry: THS-PREPARE + THS-ZERO, 54
- Data Lane HS Entry: THS-PREPARE Value, 52
- Data lane LP-TX Thevenin output high level voltage (VOH), 39
- Data lane LP-TX Thevenin output low level voltage (VOL), 41
- Data lane rise time, 42
- HS Entry TCLK PREValue, 56
- HS Exit TCLK POSTValue, 57
- Controlling the Server, 64
- Conventions, 2

## D

- D-PHY Application Commands, 69

Data storage, 35  
 Default directory usage, 7  
 Device  
   D-PHY, 21  
 Directories and usage, 7  
 Disconnect(), 96  
 Disk space, 15  
 dongle, 13  
 DUT, 15

**E**

E-mail Settings, 18  
 Exiting the application, 15

**F**

Features, 13  
 File name extensions, 8  
 Flowchart for Client  
   Programmatic Interface, 66

**G**

GetAcquireParameter(), 81  
 GetAnalyzeParameter(), 81  
 GetCurrentStateInfo(), 90  
 GetDutId(), 78  
 GetGeneralParameter(), 80  
 GetPassFailStatus(), 91  
 GetReportParameter(), 92  
 GetResultsValue(), 91  
 GetResultsValueForSubMeasurements(), 92  
 GetTimeOut(), 89  
 Global controls, 15  
 Group 1 test  
   Clock lane LP-TX Thevenin  
     output high level voltage (VOH), 45  
   Clock lane LP-TX Thevenin  
     output low level voltage (VOL), 47  
   Data lane fall time, 43  
   Data lane LP-TX Thevenin  
     output high level voltage (VOH), 39  
   Data lane LP-TX Thevenin  
     output low level voltage (VOL), 41

  Data lane rise time, 42  
 Group 1 tests, 21  
 Group 2 test  
   Clock lane fall time, 49  
   Clock lane rise time, 48  
 Group 2 tests, 21  
 Group 3 test  
   Data Lane HS Entry: Data  
     Lane TLPX Value, 51  
   Data Lane HS Entry:  
     THS-PREPARE  
     Value, 52  
 Group 3 tests, 21  
 Group 5 test  
   Clock Rising Edge Alignment  
     to First Payload Bit, 58  
   HS Entry TCLK  
     PREValue, 56  
   HS Exit TCLK  
     POSTValue, 57  
 Group 5 tests, 21

**I**

Installation  
   software, 6  
   upgrade, 6  
 Instrument discovery, 28  
 Interface, 63  
 Interface error codes, 97

**L**

LockSession(), 76

**M**

Mapping My TekExpress  
   folder, 11  
 Measurement method  
   automatic, 39  
   manual, 39  
 Menu  
   File, 15  
   Help, 18  
   Tools, 16  
   View, 16  
 Minimum system requirements, 5  
 My TekExpress, 35  
 My TekExpress folder, 10

**O**

Oscilloscopes supported, 5  
 Overview, 13

**P**

Perform analysis  
   using live waveform files, 31  
   using prerecorded waveform  
     files, 29  
 Performing Analysis  
   using prerecorded  
     waveforms, 60  
 Prerecorded waveform files, 30  
 Program example, 97  
 Programmatic Interface, 63  
 Proxy Object  
   Client, 65  
   Remote, 64

**Q**

QueryStatus(), 90

**R**

Reactivating License, 8  
 RecallSession(), 95  
 Related Documentation, 1  
 Report panel overview, 34  
 Resizing the application, 14  
 Run  
   Run button, 15  
 Run button, 15  
 Run(), 88

**S**

SaveSession(), 95  
 SaveSessionAs(), 95  
 Select  
   connected instruments, 27  
 Select Test(s), 21  
   Group 1, 21  
   Group 2, 21  
   Group 3, 21  
   Group 5, 21  
 Selecting acquisition source, 30  
 Selecting connected  
   instruments, 29

SelectTest(), 84  
SendResponse(), 90  
Server, 63  
Session folder, 35  
SetAcquireParameter(), 86  
SetAnalyzeParameter(), 86  
SetDutId(), 78  
SetGeneralParameter(), 78  
SetTimeOut(), 89  
SetVerboseMode(), 77  
Shortcut keys, 113  
Signal Path Compensation  
(SPC), 12  
Software  
  upgrade, 6  
Starting the application, 14  
Stop(), 88

## T

Technical support, 3  
TekVISA Instrument Manager, 27  
Test limits, 24  
Test Suite  
  Transmitter, 21  
Test(s)  
  Report, 34  
TestStand Client example, 100  
Tools  
  Instrument Bench, 17  
TransferImages(), 93  
TransferReport(), 93  
TransferWaveforms(), 93

## U

UnlockSession(), 96  
Untitled Session, 10  
USB dongle, 13

## V

Version  
  CTS, 21  
View  
  connected instruments, 27  
  progress of analysis, 32  
  report, 34  
  test related files, 35

## W

Waveform file, 30