



**TekExpress® D-PHY Software  
Compliance Test Application  
Printable Application Help**







**TekExpress® D-PHY Software  
Compliance Test Application  
Printable Application 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.

### **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.tek.com](http://www.tek.com) to find contacts in your area.

---

# Table of Contents

Welcome .....	5
---------------	---

## Getting help and support

Conventions .....	1
Related documentation .....	1
Technical Support .....	2

## Getting started

Supported oscilloscopes .....	3
Recommended accessories .....	3
Recommended probes .....	3
Downloading and installing the software .....	3
Activate the license .....	4
View software version .....	4
Application directories .....	5
File name extensions .....	5

## Operating basics

Launch the application .....	7
Exit the application .....	8
Application controls .....	8
Global application controls .....	10
Options menu overview .....	10
TekExpress instrument control settings .....	11
View connected instruments .....	12
Configure email settings .....	13
Application panels overview .....	14
Setup panel .....	15
Setup panel overview .....	15
Set DUT parameters .....	15
Select tests .....	18
Set acquisition tab parameters .....	19
Set configuration tab parameters .....	20
Set preferences tab parameters .....	24
Status panel overview .....	25

Results panel .....	27
Results panel overview .....	27
Preferences menu .....	28
View test-related files .....	28
Reports panel .....	29
Reports panel overview .....	29
Select report options .....	29
View a report .....	31
Report content .....	32

## Running tests

Equipment connection diagram .....	35
Prerequisite .....	36
Running tests .....	38
View test results .....	38

## Saving and recalling test setup

Test setup files overview .....	39
Save a test setup .....	39
Open load a saved test setup .....	39
Create a test setup from default settings .....	40
Create a test setup using an existing one .....	40

## TekExpress programmatic interface

About the programmatic interface .....	41
Client programmatic interface: An example .....	42
D-PHY application command arguments and queries .....	43
Connect through an IP address .....	44
Lock the server .....	46
Disable the popups .....	47
Set or get the DUT ID .....	48
Set the configuration parameters for a suite or measurement .....	49
Query the configuration parameters for a suite or measurement .....	54
Select a measurement .....	57
Select a single measurement .....	58
Select a suite .....	59
Select a channel .....	61
Configure the selected measurement .....	64

Run with set configurations or stop the run operation .....	65
Get or set the timeout value .....	66
Wait for the test to complete .....	67
After the test is complete .....	71
Save, recall, or check if a session is saved .....	75
Unlock the server .....	76
Disconnect from the server .....	77
Select a lane .....	78
Assign input signals to scope channels .....	79
Handle error codes .....	80
Program example .....	81

## SCPI commands

About SCPI command .....	83
Socket configuration for SCPI commands .....	83
TEKEXP:*IDN? .....	90
TEKEXP:*OPC? .....	90
TEKEXP:ACQUIRE_MODE .....	91
TEKEXP:ACQUIRE_MODE? .....	91
TEKEXP:EXPORT .....	92
TEKEXP:INFO? .....	92
TEKEXP:INSTRUMENT .....	92
TEKEXP:INSTRUMENT? .....	93
TEKEXP:LASTERROR? .....	94
TEKEXP:LIST? .....	94
TEKEXP:MODE .....	95
TEKEXP:MODE? .....	95
TEKEXP:POPOP .....	96
TEKEXP:POPOP? .....	96
TEKEXP:REPORT .....	97
TEKEXP:REPORT? .....	97
TEKEXP:RESULT? .....	98
TEKEXP:SELECT .....	98
TEKEXP:SELECT? .....	99
TEKEXP:SETUP .....	99
TEKEXP:STATE .....	100
TEKEXP:STATE? .....	100
TEKEXP:VALUE .....	100
TEKEXP:VALUE? .....	101

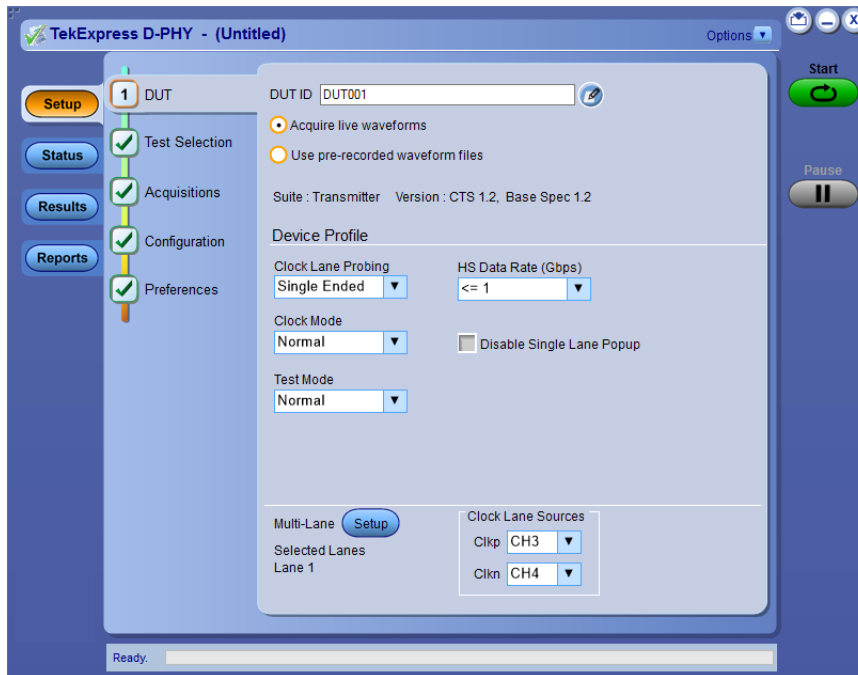
Command parameters list .....	102
Examples .....	120

## References

Pre-recorded waveform file names for test measurements .....	123
--	-----



# Welcome



The Tektronix TekExpress® D-PHY (Opt. 6-CMDPHY / DPHY12) offers a complete physical layer test solution for transmitter conformance and characterization as defined in the MIPI D-PHY v1.2 specification. The TekExpress® D-PHYTX automated test solution, along with a Tektronix 6 Series / MSO/70K oscilloscope, provides an easy way to test, debug and characterize the electrical and timing measurements of D-PHY data links.

## Key Features

- **Test time**
  - Fully automated solution: Performs D-PHY transmitter test with single-button click across High Speed (HS), Low Power (LP), Low Power-High Speed (LP-HS), and Ultra-Low Power State (ULPS) sequences in the D-PHY signal.
  - Lets you select individual tests or groups of tests.

- Eye Diagram Capability: Displays the eye diagram of the High Speed signal with option DJA enabled on the scope for test 1.5.4, Data to clock skew test.
- Image Snapshot: Option of including the snapshot of the first reference region of analysis in the captured waveform
- **100% test coverage as per D-PHY v1.2, CTS v1.2**
  - Performs fully-automated tests including Bus Turn Around (BTA) and Ultra-Low Power State (ULPS) measurements, as per D-PHY specifications up to v1.2.
- **Measurement variety**
  - D-PHYTX runs multiple scenarios like Continuous or Burst mode, Termination variations, and varying idle time.
- **Transmitter conformance test and beyond (Debug)**
  - Easily modify limits of test parameters in TekExpress for debug, margin, and characterization testing
  - Performs characterization by running TekExpress application in continuous mode to collect data
- **Signal access**
  - Use the low loading, single-ended or differential Tektronix TDP7700(6 Series) / P7700(70KC/DX/SX) Series High Impedance TriMode probe to measure MIPI signals
  - Supports TekFlex™ accessories for flexible probing
- **Offline and remote analysis**
  - Analyze live or pre-acquired waveforms
  - Allows remote execution of tests




# Getting help and support

## Conventions

Help uses the following conventions:

- The term "Application," and "Software" refers to the TekExpress D-PHY application.
- The term "DUT" is an abbreviation for Device Under Test.
- The term "select" is a generic term that applies to the two methods of choosing a screen item (button control, list item): using a mouse or using the touch screen.
- A Note identifies important information.

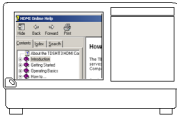

**Table 1: Icon descriptions**

Icon	Meaning
	This icon identifies important information.
	This icon identifies conditions or practices that could result in loss of data.
	This icon identifies additional information that will help you use the application more efficiently.

## Related documentation

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

**Table 2: Product documentation**

Item	Purpose	Location
Online Help	In-depth operation and UI help.	
PDF of the Online Help (077-0XXX-XX)	In-depth operation and UI help.	

## 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. Contact Tektronix through mail, telephone, or the Web site. See [Contacting Tektronix](#) at the front of this document for contact information.

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
- Modules 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

# Getting started

## Supported oscilloscopes

### Supported models

- MSO/DPO70000 C/DX Series Oscilloscope with bandwidth 4 GHz and above.
- DPO70000 SX Series Oscilloscope (Four channel) with bandwidth upto 33 GHz.

## Recommended accessories

### Termination board

TMPC-CTB D-PHY UNH-IOL board

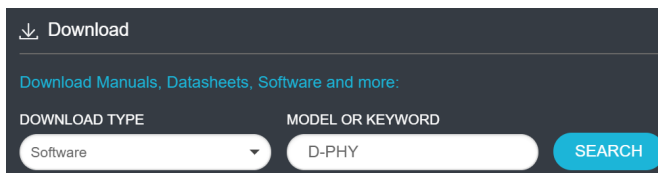
## Recommended probes

P7700 Series Tri-mode probe with P77STFLXA solder-in tip with TekFlex connector technology (required four numbers)

## Downloading and installing the software

Complete the following steps to download and install the latest D-PHY application. See [Supported oscilloscopes](#) on page 3 for compatibility.

1. Go to [www.tek.com](http://www.tek.com).
2. Click **Downloads**. In the Downloads menu, select DOWNLOAD TYPE as Software and enter *D-PHY* in the MODEL OR KEYWORD field and click **SEARCH**.



3. Select the latest version of the software and follow the instructions to download. Copy the executable file to the oscilloscope.
4. Double-click the executable and follow the on-screen instructions. The software is installed at `C:\Program Files\Tektronix\TekExpress\TekExpress D-PHY`.
5. Select **Analyze > TekExpress D-PHY** from the TekScope menu to launch the application.

## Activate the license

Activate the license using the **Option Installation** wizard in the TekScope application:

1. In the TekScope application menu bar, click **Utilities > Option Installation**.  
The **TekScope Option Installation** wizard opens.
2. Push the **F1** key on the oscilloscope keyboard to open the Option Installation help topic.
3. Follow the directions in the help topic to activate the license.

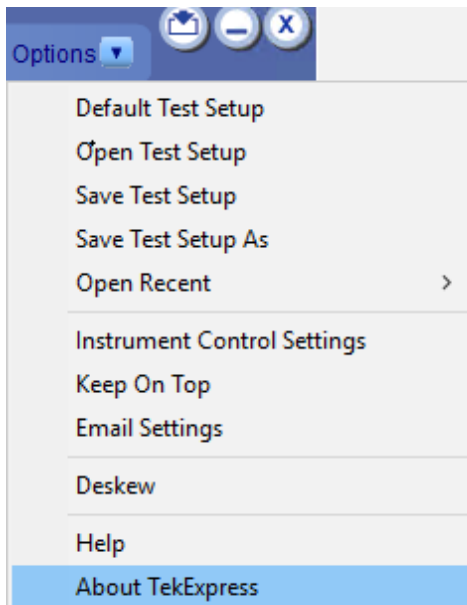
### See also

[View version and license information](#)

## View software version

Use the following instructions to view version information for the application and for the application modules such as the Programmatic Interface and the Programmatic Interface Client.

To view version information for D-PHY, click **Options > About TekExpress**.

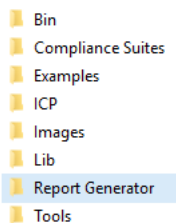


## Application directories

The TekExpress D-PHY application files are installed at the following location:

C:\Program Files\Tektronix\TekExpress\TekExpress D-PHY

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
Bin	Contains application libraries
Compliance Suites	Contains test suite specific files
Examples	Contains various support files
ICP	Contains instrument and application specific interface libraries
Images	Contains images of the application
Lib	Contains utility files specific to the application
Report Generator	Contains style sheets for report generation
Tools	Contains instrument and application specific files

## File name extensions

The TekExpress D-PHY software uses the following file name extensions:

**Table 4: File name extension**

File name extension	Description
.TekX	Application session files (the extensions may not be displayed)
.py	Python sequence file
.xml	Test-specific configuration information (encrypted) files Application log files
.csv	Test result reports Plot data
.mht	Test result reports (default) Test reports can also be saved in HTML format
.pdf	Test result reports Application help document

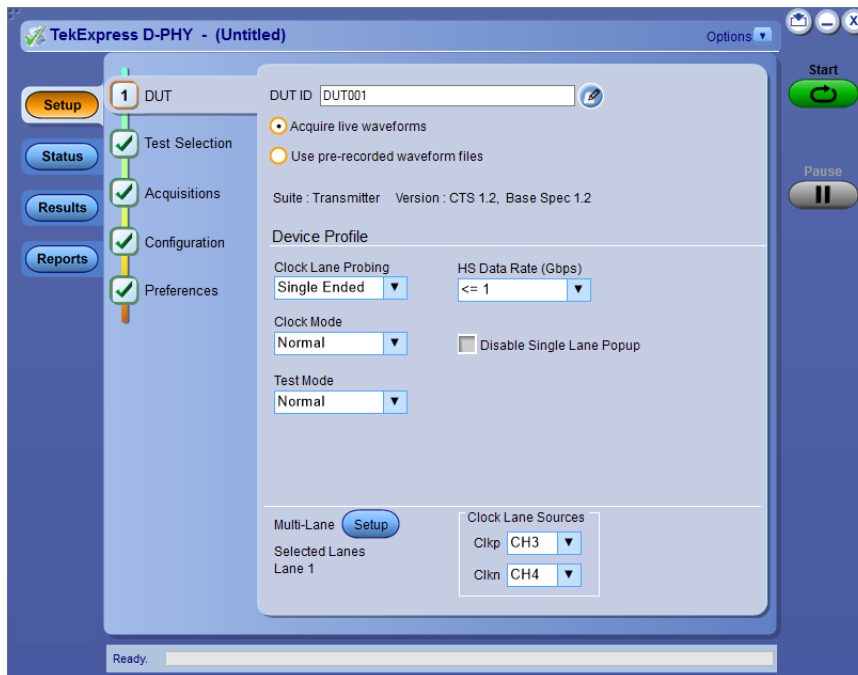
File name extension	Description
.xslt	Style sheet used to generate reports
.png	Captured images



# Operating basics

## Launch the application

To launch the TekExpress D-PHY application, select **Analyze > TekExpress D-PHY** from the oscilloscope menu.



After first launch of Tekexpress D-PHY application following changes take place on the oscilloscope.

During launch, "My TekExpress" folder is created in the Documents folder of the current user and gets mapped to "X" drive. When the application is closed properly, the "X" drive will get unmapped.

---

**NOTE.** If a user with new login ID launches "TekExpress D-PHY.exe", "My TekExpress" folder is created in the Documents folder of the new user.

---

When you first run the application after installation, the application checks for Resources.xml located in the X:\ folder. The Resources.xml file gets created into X: drive, if the file is not found, then the application creates file with equipment details. Session files are then stored inside the X:\D-PHY folder. If this file is not found, the application runs an instrument discovery program to detect connected instruments before launching TekExpress D-PHY.

To keep the TekExpress D-PHY application window on top, select Keep On Top from the [Options menu](#). If the application goes behind the oscilloscope application, click Application > TekExpress D-PHY to move the application to be in front.

---


**NOTE.** When **Keep on Top** is selected, you cannot access the combo boxes in the application panels.

---

**See also**

[Exit the application](#)

## Exit the application

To exit the application, click  on the application title bar. Follow on-screen prompts to save any unsaved session, save test setup files, or exit the application.

---


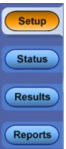



**NOTE.** Using other methods to exit the application can result in abnormal termination of the application.








---

## Application controls

This section describes the application controls.

**Table 5: Application control description**


Item	Description
<p><i>Options menu</i></p> 	Menu to display global application controls.
<p><i>Test panel</i></p> 	Controls that open tabs for configuring test settings and options.
<p>Start / Stop button</p>  	Use the <b>Start</b> button to start the test run of the measurements in the selected order. If prior acquired measurements are not cleared, then new measurements are added to the existing set. The button toggles to the Stop mode while tests are running. Use the <b>Stop</b> button to abort the test.
<p>Pause / Continue button</p> 	Use the <b>Pause</b> button to pause the acquisition. When a test is paused, this button changes as <b>Continue</b> .

Item	Description
<p>Clear button</p> 	<p>Use the <b>Clear</b> button to clear all existing measurement results. Adding or deleting a measurement, or changing a configuration parameter of an existing measurement, also clears measurements. This is to prevent the accumulation of measurement statistics or sets of statistics that are not coherent. This button is available only on <a href="#">Results panel</a>.</p> <hr/> <p><b>NOTE.</b> <i>This button is visible only when there are results data on the panel.</i></p>
<p>Application window move icon</p> 	<p>Place the cursor over the top of the application window to move the application window to the desired location</p>
<p>Minimize icon</p> 	<p>Minimizes the application.</p>
<p>Close icon</p> 	<p>Close the application.</p>
<p>Mini view / Normal view</p>  	<p>Toggles the application between mini view and normal view. Mini view displays the run messages with the time stamp, progress bar, Start / Stop button, and Pause / Continue button. The application moves to mini view when you click the <b>Start</b> button.</p> 

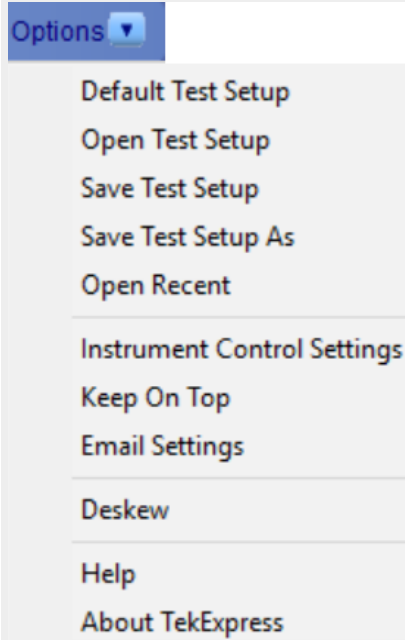
## Global application controls

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

### Options menu overview

To access Options menu, click  in the upper-right corner of the application. It has the following selections:

#### Options menu



Menu	Function
Default Test Setup	Opens an untitled test setup with defaults selected
Open Test Setup	Opens a saved test setup
Save Test Setup	Saves the current test setup
Save Test Setup As	Saves the current test setup with a different file name or file type
Open Recent	Displays the recently opened test setups to open
<i>Instrument Control Settings</i>	Detects, lists, and refreshes the connected instruments found on specified connections (LAN, GPIB, USB, and so on)
Keep On Top	Keeps the TekExpress D-PHY application on top of all the application  <i><b>NOTE.</b> When <b>Keep on Top</b> is selected, you cannot access the combo boxes in the application panels.</i>
<i>Email Settings</i>	Use to configure email options for test run and results notifications
<i>Deskew</i>	Allows to read the deskew, attenuation, and bandwidth values from the TekScope application. Before using this option, manually compensate for skew and attenuations in TekScope application. The values read are stored into a text file in the ICP folder. The stored values can also be set on the TekScope through this option.

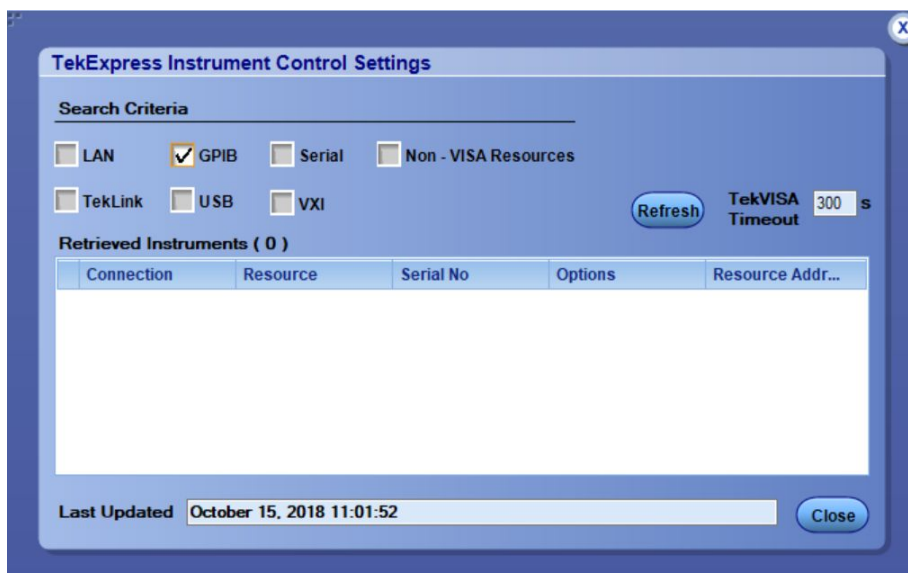
Menu	Function
Help	Displays the TekExpress D-PHY help
<a href="#">About TekExpress</a>	<ul style="list-style-type: none"> <li>■ Displays application details such as software name, version number, and copyright</li> <li>■ Provides a link to the end-user license agreement</li> <li>■ Provides a link to the Tektronix Web site</li> </ul>

See also. [Application controls](#)

## TekExpress instrument control settings

Use **TekExpress Instrument Control Settings** dialog box to search the instruments (resources) connected to the application. You can use the Search Criteria to search the connected instruments depending on the connection type. The details of the connected instrument is displayed in the Retrieved Instruments window.

To access, click **Options > Instrument Control Settings**.



The connected instruments displayed here can be selected for use under Global Settings in the test configuration section.

---

**NOTE.** Select GPIB (Default) and LAN when using TekExpress D-PHY application on MSO6.

---

See also. [Options menu overview](#)

## View connected instruments

Use TekExpress Instrument Control Settings dialog box to search the instruments (resources) connected to the application. The application uses TekVISA to discover the connected instruments.

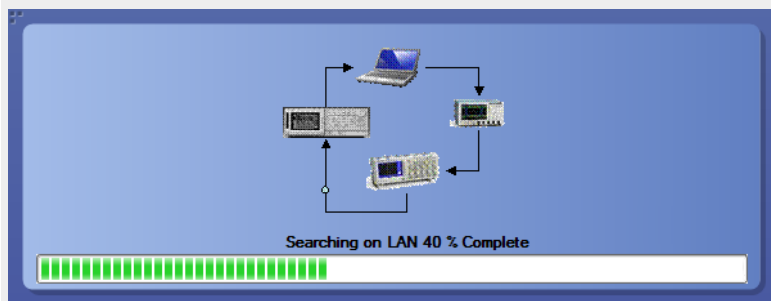
**NOTE.** The instruments required for the test setup must be connected and it must be recognized by the application before running the test.

To refresh the list of connected instruments:

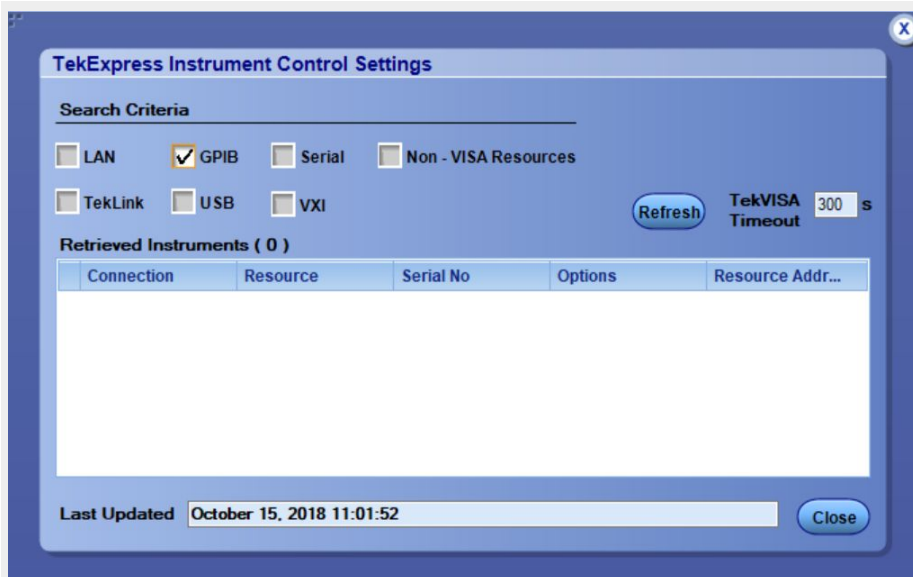
1. From the Options menu, select **Instrument Control Settings**.
2. In the **Search Criteria** section of the Instrument Control Settings dialog box, select the connection types of the instruments to search.

Instrument search is based on the VISA layer, but different connections determine the resource type, such as LAN, GPIB, and USB. For example, if you choose LAN, the search will include all the instruments supported by TekExpress that are communicating over the LAN.

3. Click **Refresh**. TekExpress searches for connected instruments.



4. After searching, the dialog box lists the instrument-related details based on the search criteria. For example, for the Search Criteria as LAN and GPIB, the application displays all LAN and GPIB instruments connected to the application.



The details of the instruments are displayed in the Retrieved Instruments table. The time and date of instrument refresh is displayed in the Last Updated field.

**See also.** [Equipment connection diagram](#) on page 35

## Configure email settings

Use the Email Settings utility to get notified by email when a measurement completes, or produces any error condition. Follow the steps to configure email settings:

### Email Settings

1. Select **Options > Email Settings** to open the Email Settings dialog box.
2. (Required) For **Recipient email Address(es)**, enter one or more recipient email addresses. To include multiple addresses, separate the addresses with commas.
3. (Required) For **Sender's Address**, enter the email address used by the instrument. This address consists of the instrument name, followed by an underscore, followed by the instrument serial number, then the @ symbol, and the email server ID. For example: user@yourcompany.com.
4. (Required) In the **Server Configuration** section, type the SMTP Server address of the Mail server configured at the client location, and the SMTP Port number, in the corresponding fields.

If this server requires password authentication, enter a valid login name, password, and host name in the corresponding fields.

---

**NOTE.** If any of the above required fields are left blank, the settings will not be saved and email notifications will not be sent.

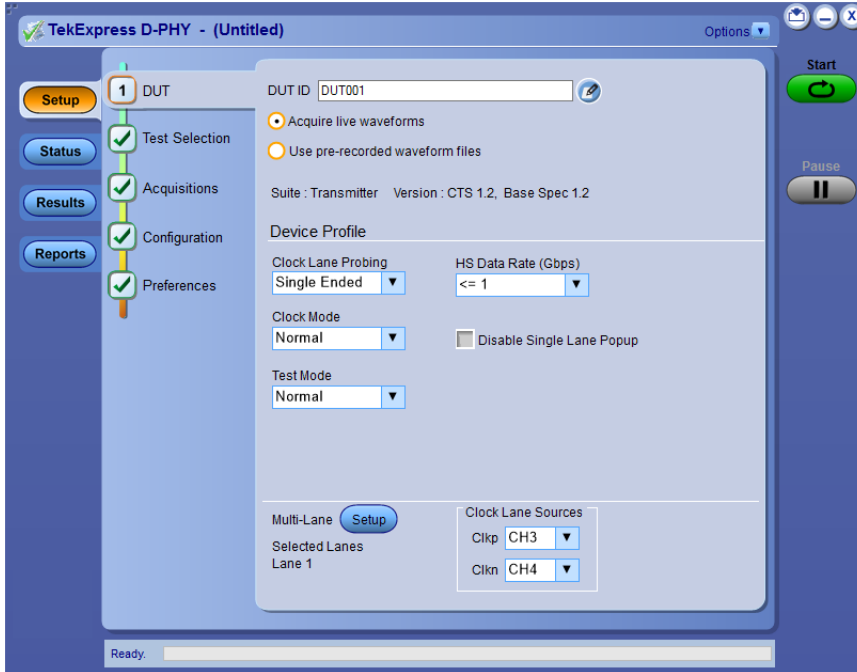
---

5. In the **Email Attachments** section, select from the following options:
  - **Reports:** Select to receive the test report with the notification email.
  - **Status Log:** Select to receive the test status log with the notification email. If you select this option, then also select whether you want to receive the full log or just the last 20 lines.
6. In the **Email Configuration** section:
  - Enter a maximum file size for the email message. Messages with attachments larger than this limit will not be sent. The default is 5 MB.
  - Enter the number in the Number of Attempts to Send field, to limit the number of attempts that the system makes to send a notification. The default is 1. You can also specify a timeout period.
7. Select the **Email Test Results When complete or on error** check box. Use this check box to quickly enable or disable email notifications.
8. To test your email settings, click **Test Email**.

9. To apply your settings, click **Apply**.
10. Click **Close** when finished.

## Application panels overview

TekExpress D-PHY solution uses panels to group Test Setup Configuration, Results, and Reports settings. Click any button to open the associated panel. A panel may have one or more tabs that list the selections available in that panel. Controls in a tab can change depending on settings made in the same tab or another tab.



**Table 6: Application panels overview**

Panel Name	Purpose
<a href="#">Setup</a> panel	The Setup panel shows the test setup controls. Click the <b>Setup</b> button to open this panel. Use this panel to: <ul style="list-style-type: none"> <li>■ <a href="#">Set DUT tab parameters</a></li> <li>■ <a href="#">Select tests</a></li> <li>■ <a href="#">Set acquisition tab parameters</a></li> <li>■ <a href="#">Set configuration tab parameters</a></li> <li>■ <a href="#">Set preferences tab parameters</a></li> </ul>
<a href="#">Status</a> panel	View the progress and analysis status of the selected tests, and view test logs.
<a href="#">Results</a> panel	View the summary of test results and select result viewing preferences.
<a href="#">Reports</a> panel	Browse for reports, save reports as specific file types, specify report naming conventions, select report content to include (summary information, detailed information, user comments, setup configuration, application configuration, etc.), and select report viewing options.



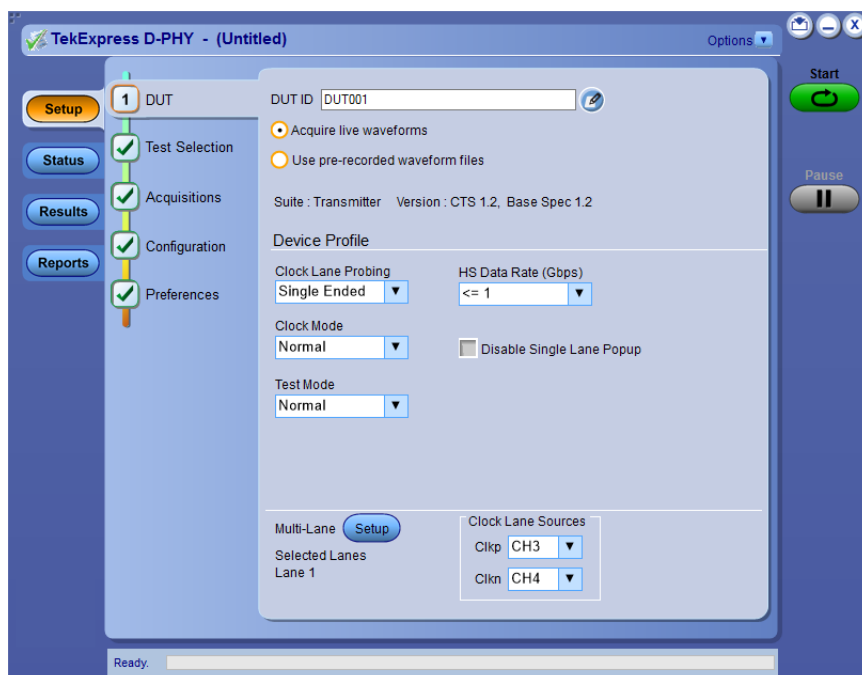
## See also

[Application controls](#)

# Setup panel

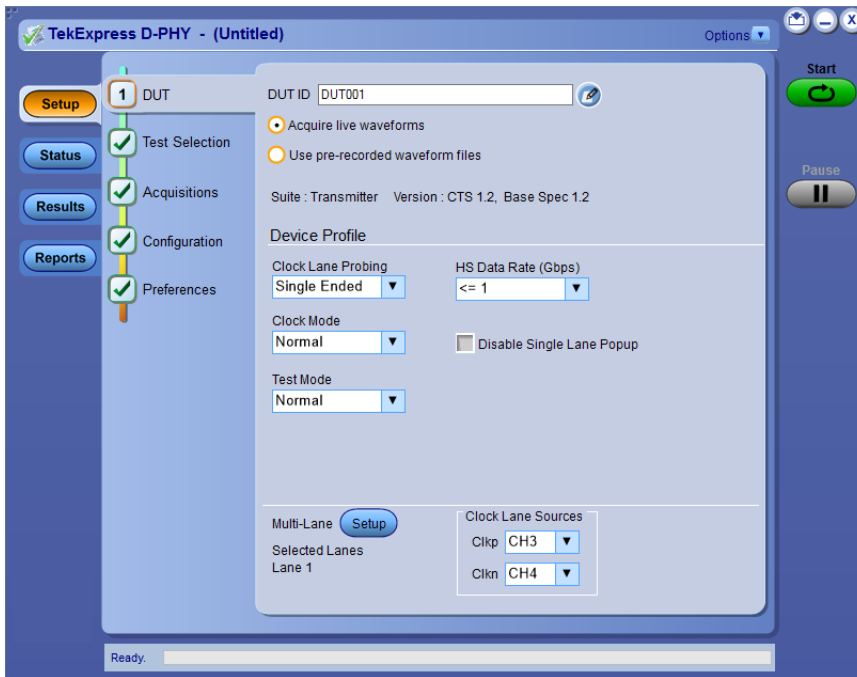
## Setup panel overview

The Setup panel contains sequentially ordered tabs that help you guide through the test setup and execution process.




## Set DUT parameters

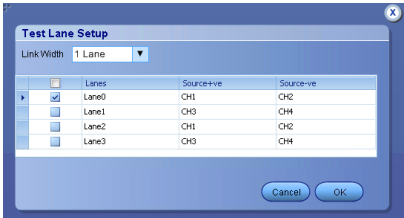
Use the DUT tab to select parameters for the device under test. These settings are global and apply to all tests of current session. DUT settings also affect the list of available tests in the Test Selection tab.



Click **Setup > DUT** to access the DUT parameters:

**Table 7: DUT tab settings**

Setting	Description
DUT ID	Adds an optional text label for the DUT to reports. The default value is DUT001. The maximum number of characters is 32. You cannot use the following characters in an ID name: (,.,,.,.,.,\,/:?> *)
 Comments icon (to the right of the DUT ID field)	Opens Comments dialog box to enter text to add to the report. Maximum size is 256 characters. To enable or disable comments appearing on the test report, see <a href="#">Select report options</a> .
Acquire live waveforms	Acquire active signals from the DUT for measurement and analysis.
Use pre-recorded waveform files	Run tests on a saved waveform. <a href="#">Open (load) a saved test setup</a> <a href="#">Click here</a> for details about the file names for pre-recorded waveform.
Suite	This application is intended for D-PHY Transmitter testing. Version supported is CTS 1.2 Base Spec 1.2.
Version (1.2)	
<b>Device Profile</b>	
Clock Lane Probing	Select either single-ended mode or differential mode for clock lane.  <b>NOTE.</b> <i>When Differential mode is selected, some tests are not available for execution.</i>
Clock Mode	Select the clock mode. The available options are Normal and Continuous. When clock mode is continuous, only normal tests can be performed.  <b>NOTE.</b> <i>Some tests are not supported in continuous clock mode.</i>
Test Mode	Select the test mode. Based on the selection made, the list of tests in the Test Selection panel will change.

Setting	Description
HS Data Rate (Gbps)	<p>Select the HS Data Rate. Available options are</p> <ul style="list-style-type: none"> <li>■ ≤1 Gbps</li> <li>■ &gt;1 and ≤1.5 Gbps</li> <li>■ &gt;1.5 Gbps</li> </ul> <p>The test limits may vary based on the selected data rate.</p> <hr/> <p><b>NOTE.</b> If the selected data rate is &gt; 1.5 Gbps:</p> <ul style="list-style-type: none"> <li>■ 1.4.18 Clock Lane HS Clock Delta UI is disabled.</li> <li>■ 1.5.5 Initial HS Skew Calibration Burst (TSKEWCAL-SYNC, TSKEWCAL) and 1.5.6 Periodic HS Skew Calibration Burst (TSKEWCAL-SYNC, TSKEWCAL) tests are enabled only when the data rate is &gt;1.5 Gbps.</li> </ul>
Disable Single Lane Popup	<p>When selected, disables the connection popup.</p> <hr/> <p><b>NOTE.</b> Before selecting this option ensure that the connections are made as per the selected lane channel configuration.</p>
<p>Multi-Lane Setup</p> 	<p>Displays the test lanes selected for the test session To change lanes selected for testing, click <b>Setup</b>.</p> <ol style="list-style-type: none"> <li>1. In the Test Lane Setup dialog box, select the desired number of lanes from the Link Width drop-down list.</li> <li>2. To select the lanes to use, click the corresponding lane buttons: <ul style="list-style-type: none"> <li>■ To select all four lanes at once, click <b>Select All</b>. If you select this, select <b>4 Lanes</b> from the Link Width drop-down list.</li> <li>■ To deselect all selected lanes, click <b>Deselect All</b>.</li> <li>■ If you select 1 Lane, only one lane can be used.</li> <li>■ If you select 2 Lanes, any one lane or two lanes can be used.</li> <li>■ If you select 4 Lanes, only a lane or two lanes or four lanes can be used.</li> </ul> </li> <li>3. <b>Source selection:</b> Configure the channel sources on which the corresponding data lanes are connected.</li> <li>4. Click <b>OK</b>.</li> </ol> <p>Your selections display in the Lane Setup section of the DUT tab.</p>
Clock Sources	<p>Configure the channel source(s) on which the clock lanes are connected.</p> <hr/> <p><b>NOTE.</b></p> <ul style="list-style-type: none"> <li>■ If the clock probing is differential, only one source need to be configured.</li> <li>■ The channel sources selected for Data lanes should not be same as the sources selected for Clock lane. If the channels are same, the execution will not proceed.</li> </ul>

See also. [Select tests](#)

## Select tests

Use the Test Selection tab to select the tests. The test measurements available depends on the settings selected in the DUT tab.

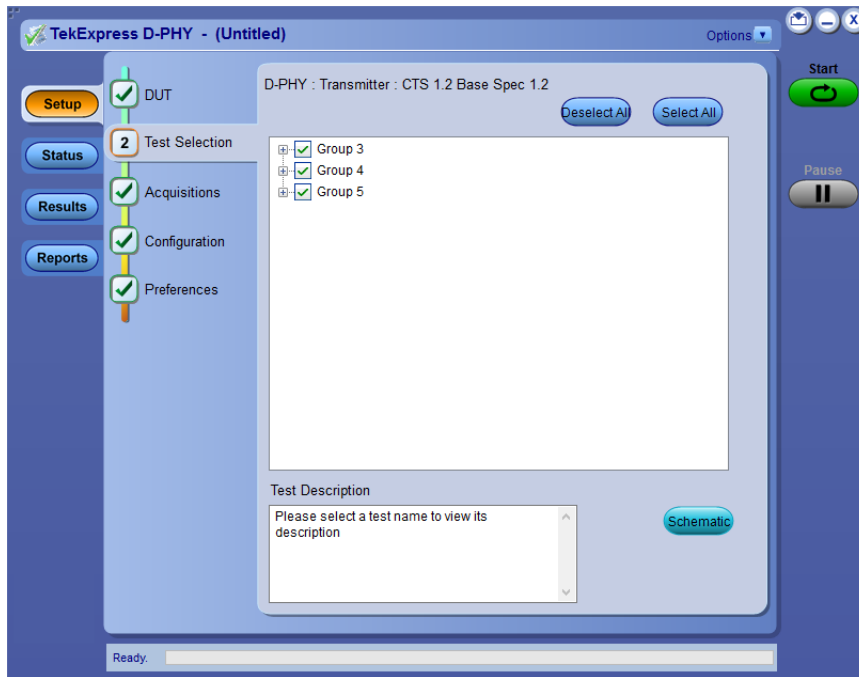


Figure 1: TekExpress D-PHY measurements

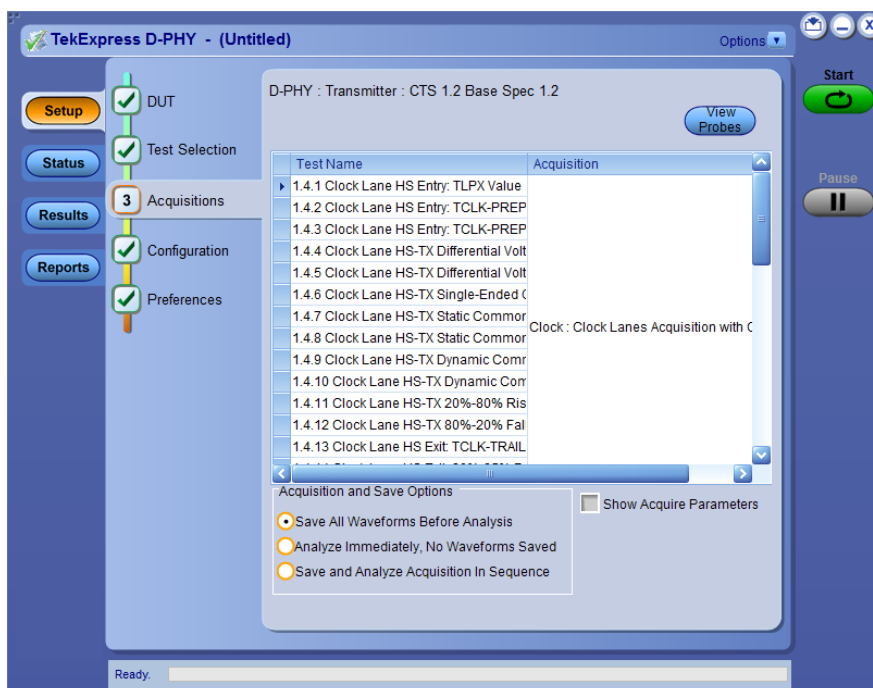
Table 8: Test Selection tab settings

Setting	Description
Deselect All Select All	Deselect or select all tests in the list.
Tests	Click on a test to select or unselect. Highlight a test to show details in the Test Description pane. The application automatically selects all required tests when in Compliance mode.
Test Description	Shows brief description of the highlighted test in the test tree.
Schematic	Shows an equipment and test fixture setup schematic (connection diagram) for the selected test. Use to set up the equipment and fixtures or to verify the setup before running the test.

See also. [Set acquisition tab parameters](#)

## Set acquisition tab parameters

Use Acquisitions tab to view the test acquisition parameters. The contents displayed on this tab depends on the DUT type and the tests selected.



**Table 9: Acquisitions tab settings**

Settings	Description
View Probes	View the detected probe configuration. Use the View Probes dialog box to view the connected probes.
<b>Acquisition and Save options</b>	
Save All Waveforms Before Analysis	Saves all the waveforms before the analysis.
Analyze Immediately, No Waveforms Saved	Analysis the waveforms and no waveforms are saved.
	<b>NOTE:</b> No .wfm files will not be present in the run folder.
Save and Analyze Acquisition In Sequence	Saves and then analyses the acquisition in sequence.
Show Acquire Parameters	Select to view the acquisition parameters.

TekExpress D-PHY saves all acquisition waveforms to files by default. Waveforms are saved in a unique folder for each session (a session is started when you click the Start button). The folder path is `X:\TekExpress D-PHY\Untitled Session \<dutid>\<date>_<time>`. Images created for each analysis, XML files with result values, reports, and other information specific to that particular execution are also saved in this folder.

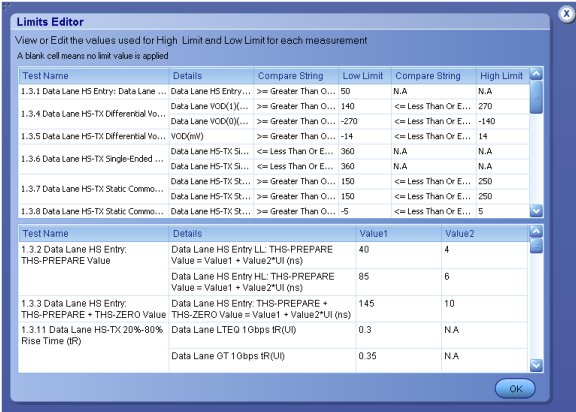
Saving a session moves the session file contents from the Untitled Session folder to the specified folder name, and changes the session name to the specified name.

**See also.** [Pre-recorded waveform file names for test measurements](#)

## Set configuration tab parameters

Use Configuration tab to view and configure the Global Settings and the measurement configurations. The measurement specific configurations available in this tab depends on the selections made in the DUT panel and Test Selection panel.

**Table 10: Configuration tab: Common parameters**

Setting	Description																																																																																				
Mode	<p>Determines whether test parameters are in compliance or can be edited</p> <ul style="list-style-type: none"> <li>Compliance: All the test parameters are editable except <b>Use Cursors</b> in Global Settings.</li> <li>User Defined: All test parameters and global parameters are editable in this mode.</li> </ul>																																																																																				
Limits Editor	<p>Displays the upper and lower limits for the applicable measurement using different types of comparisons. In the Compliance Mode, you can view the measurement high and low limits used for the tests displayed in the tree view of the Measurements tab. When running tests in User Defined Mode, you can edit the limit settings in the Limits Editor.</p> <p>The second table shows the tests with the limits calculated dynamically as per the specification.</p>  <p>The screenshot shows a window titled "Limits Editor" with the following table:</p> <table border="1"> <thead> <tr> <th>Test Name</th> <th>Details</th> <th>Compare String</th> <th>Low Limit</th> <th>Compare String</th> <th>High Limit</th> </tr> </thead> <tbody> <tr> <td>1.3.1 Data Lane HS Entry: Data Lane ...</td> <td>Data Lane HS Entry...</td> <td>&gt;= Greater Than O...</td> <td>150</td> <td>N/A</td> <td>N/A</td> </tr> <tr> <td></td> <td>Data Lane VOD(1)...</td> <td>&gt;= Greater Than O...</td> <td>140</td> <td>&lt;= Less Than Or E...</td> <td>270</td> </tr> <tr> <td>1.3.4 Data Lane HS-TX Differential Vo...</td> <td>Data Lane VOD(0)...</td> <td>&gt;= Greater Than O...</td> <td>-270</td> <td>&lt;= Less Than Or E...</td> <td>-140</td> </tr> <tr> <td>1.3.5 Data Lane HS-TX Differential Vo...</td> <td>VOD(nV)</td> <td>&gt;= Greater Than O...</td> <td>-14</td> <td>&lt;= Less Than Or E...</td> <td>14</td> </tr> <tr> <td>1.3.6 Data Lane HS-TX Single-Ended ...</td> <td>Data Lane HS-TX St...</td> <td>&lt;= Less Than Or E...</td> <td>360</td> <td>N/A</td> <td>N/A</td> </tr> <tr> <td></td> <td>Data Lane HS-TX St...</td> <td>&lt;= Less Than Or E...</td> <td>360</td> <td>N/A</td> <td>N/A</td> </tr> <tr> <td>1.3.7 Data Lane HS-TX Static Commo...</td> <td>Data Lane HS-TX St...</td> <td>&gt;= Greater Than O...</td> <td>150</td> <td>&lt;= Less Than Or E...</td> <td>250</td> </tr> <tr> <td>1.3.8 Data Lane HS-TX Static Commo...</td> <td>Data Lane HS-TX St...</td> <td>&gt;= Greater Than O...</td> <td>150</td> <td>&lt;= Less Than Or E...</td> <td>250</td> </tr> <tr> <td></td> <td>Data Lane HS-TX St...</td> <td>&gt;= Greater Than O...</td> <td>-5</td> <td>&lt;= Less Than Or E...</td> <td>5</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Test Name</th> <th>Details</th> <th>Value1</th> <th>Value2</th> </tr> </thead> <tbody> <tr> <td>1.3.2 Data Lane HS Entry: THS-PREPARE Value</td> <td>Data Lane HS Entry LL: THS-PREPARE Value = Value1 + Value2*UI (ns)</td> <td>40</td> <td>4</td> </tr> <tr> <td></td> <td>Data Lane HS Entry HL: THS-PREPARE Value = Value1 + Value2*UI (ns)</td> <td>85</td> <td>6</td> </tr> <tr> <td>1.3.3 Data Lane HS Entry: THS-PREPARE + THS-ZERO Value</td> <td>Data Lane HS Entry: THS-PREPARE + THS-ZERO Value = Value1 + Value2*UI (ns)</td> <td>145</td> <td>10</td> </tr> <tr> <td>1.3.11 Data Lane HS-TX 20%-80% Rise Time (R)</td> <td>Data Lane LTE0 1 Gbps IR(UI)</td> <td>0.3</td> <td>N/A</td> </tr> <tr> <td></td> <td>Data Lane GT 1 Gbps IR(UI)</td> <td>0.35</td> <td>N/A</td> </tr> </tbody> </table>	Test Name	Details	Compare String	Low Limit	Compare String	High Limit	1.3.1 Data Lane HS Entry: Data Lane ...	Data Lane HS Entry...	>= Greater Than O...	150	N/A	N/A		Data Lane VOD(1)...	>= Greater Than O...	140	<= Less Than Or E...	270	1.3.4 Data Lane HS-TX Differential Vo...	Data Lane VOD(0)...	>= Greater Than O...	-270	<= Less Than Or E...	-140	1.3.5 Data Lane HS-TX Differential Vo...	VOD(nV)	>= Greater Than O...	-14	<= Less Than Or E...	14	1.3.6 Data Lane HS-TX Single-Ended ...	Data Lane HS-TX St...	<= Less Than Or E...	360	N/A	N/A		Data Lane HS-TX St...	<= Less Than Or E...	360	N/A	N/A	1.3.7 Data Lane HS-TX Static Commo...	Data Lane HS-TX St...	>= Greater Than O...	150	<= Less Than Or E...	250	1.3.8 Data Lane HS-TX Static Commo...	Data Lane HS-TX St...	>= Greater Than O...	150	<= Less Than Or E...	250		Data Lane HS-TX St...	>= Greater Than O...	-5	<= Less Than Or E...	5	Test Name	Details	Value1	Value2	1.3.2 Data Lane HS Entry: THS-PREPARE Value	Data Lane HS Entry LL: THS-PREPARE Value = Value1 + Value2*UI (ns)	40	4		Data Lane HS Entry HL: THS-PREPARE Value = Value1 + Value2*UI (ns)	85	6	1.3.3 Data Lane HS Entry: THS-PREPARE + THS-ZERO Value	Data Lane HS Entry: THS-PREPARE + THS-ZERO Value = Value1 + Value2*UI (ns)	145	10	1.3.11 Data Lane HS-TX 20%-80% Rise Time (R)	Data Lane LTE0 1 Gbps IR(UI)	0.3	N/A		Data Lane GT 1 Gbps IR(UI)	0.35	N/A
Test Name	Details	Compare String	Low Limit	Compare String	High Limit																																																																																
1.3.1 Data Lane HS Entry: Data Lane ...	Data Lane HS Entry...	>= Greater Than O...	150	N/A	N/A																																																																																
	Data Lane VOD(1)...	>= Greater Than O...	140	<= Less Than Or E...	270																																																																																
1.3.4 Data Lane HS-TX Differential Vo...	Data Lane VOD(0)...	>= Greater Than O...	-270	<= Less Than Or E...	-140																																																																																
1.3.5 Data Lane HS-TX Differential Vo...	VOD(nV)	>= Greater Than O...	-14	<= Less Than Or E...	14																																																																																
1.3.6 Data Lane HS-TX Single-Ended ...	Data Lane HS-TX St...	<= Less Than Or E...	360	N/A	N/A																																																																																
	Data Lane HS-TX St...	<= Less Than Or E...	360	N/A	N/A																																																																																
1.3.7 Data Lane HS-TX Static Commo...	Data Lane HS-TX St...	>= Greater Than O...	150	<= Less Than Or E...	250																																																																																
1.3.8 Data Lane HS-TX Static Commo...	Data Lane HS-TX St...	>= Greater Than O...	150	<= Less Than Or E...	250																																																																																
	Data Lane HS-TX St...	>= Greater Than O...	-5	<= Less Than Or E...	5																																																																																
Test Name	Details	Value1	Value2																																																																																		
1.3.2 Data Lane HS Entry: THS-PREPARE Value	Data Lane HS Entry LL: THS-PREPARE Value = Value1 + Value2*UI (ns)	40	4																																																																																		
	Data Lane HS Entry HL: THS-PREPARE Value = Value1 + Value2*UI (ns)	85	6																																																																																		
1.3.3 Data Lane HS Entry: THS-PREPARE + THS-ZERO Value	Data Lane HS Entry: THS-PREPARE + THS-ZERO Value = Value1 + Value2*UI (ns)	145	10																																																																																		
1.3.11 Data Lane HS-TX 20%-80% Rise Time (R)	Data Lane LTE0 1 Gbps IR(UI)	0.3	N/A																																																																																		
	Data Lane GT 1 Gbps IR(UI)	0.35	N/A																																																																																		

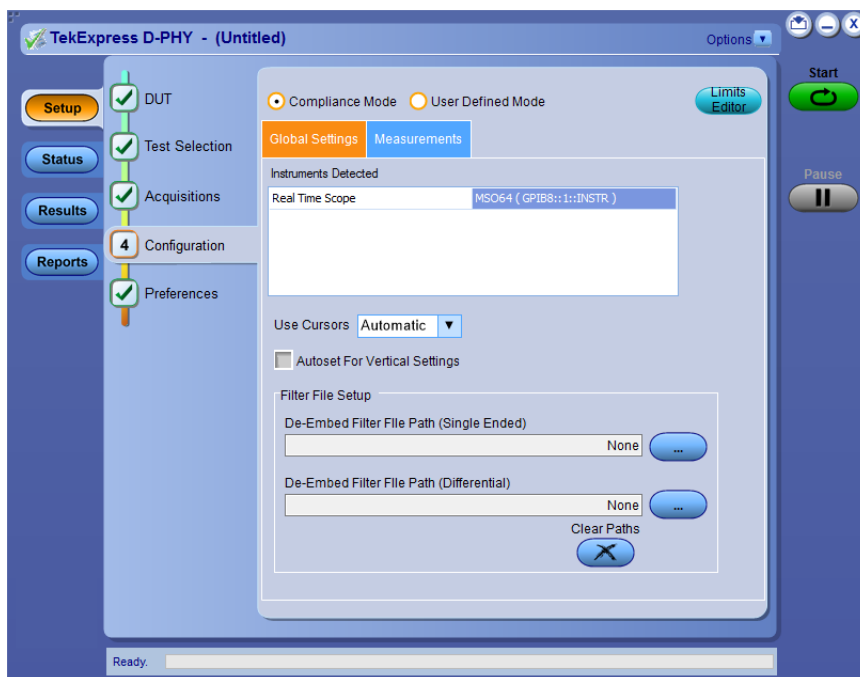


Figure 2: Configuration tab: Global Settings

Table 11: Configuration tab: Global settings

Setting	Description
<b>Global Settings</b>	
Instruments Detected	<p>Displays the instruments connected to this application. Click on the instrument name to open a list of available (detected) instruments.</p> <p>Select <b>Options &gt; Instrument Control Settings</b> and click Refresh to update the instrument list.</p> <p><b>NOTE.</b> Verify that the <b>LAN and GPIB</b> search criteria (default setting) in the Instrument Control Settings is selected when using TekExpress D-PHY application.</p>
Use Cursors	<ul style="list-style-type: none"> <li>■ <b>Automatic</b> mode: the results are computed automatically and are displayed.</li> <li>■ <b>Manual</b> mode: you will be prompted to place cursors at the desired region in the acquired waveform. The cursors will be used as the gating criteria for the measurement and the measured value will be reported.</li> </ul> <p><b>NOTE.</b> You can change the settings to <b>Manual</b> mode only if User Defined Mode is selected in the Configuration tab.</p>
<b>Filter File Setup</b>	
De-Embed Filter File Path (Single Ended)	Browse and select the file for de-embedding the single ended signals.
De-Embed Filter File Path (Differential)	Browse and select the file for de-embedding the differential probe.
Clear Paths	When clicked it resets both the single ended and differential filter file paths.

**NOTE.** The selection of the filter file for de-embedding is based on the sample rate selection on the horizontal settings tab in the measurements panel.

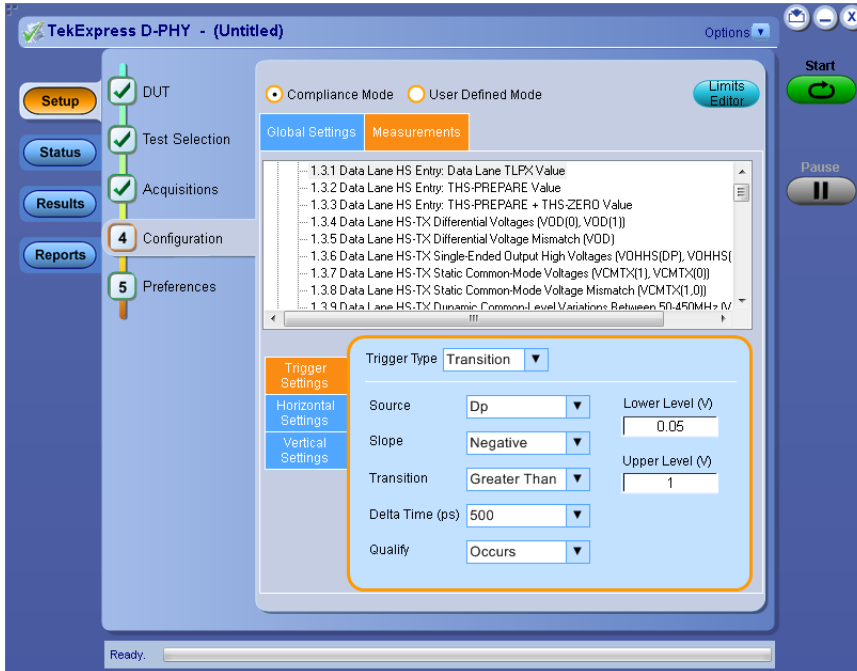


Figure 3: Configuration tab: Measurements

Table 12: Configuration tab: Measurements settings

Setting	Description
Measurements	<p>Displays the measurements which are selected in the Test Selection tab. The tests are grouped with unique acquisition type names.</p> <p><b>NOTE.</b></p> <ul style="list-style-type: none"> <li>When a parent test group is selected and any change is made, the change will be applied to all the tests.</li> <li>Individual test configuration is possible by selecting the test and making the required changes.</li> </ul> <p>If changes are made to individual tests, it becomes a separate acquisition during execution of tests. The difference between the acquisitions can be viewed by clicking <b>Acquisitions &gt; Show Acquire Parameter</b>.</p>
<b>Analyze</b>	
<p>Analysis Mode Available for 1.5.3 HS Clock Rising Edge Alignment to First Payload Bit only.</p>	<ul style="list-style-type: none"> <li>Automatic (default)</li> <li>Visual Verification</li> </ul>
<p>UI INST MIN (ns) Available for 1.4.17 Clock Lane HS Clock Instantaneous (UIINST) only.</p>	<p>Specify the UI INST MIN. The default value is 1.25 ns.</p>



Setting	Description
<b>Trigger Settings</b>	
Trigger Type	Select the trigger type. The configurations may change depending on the Trigger Type selected. <ul style="list-style-type: none"> <li>■ Transition (default)</li> <li>■ Edge</li> <li>■ Width</li> </ul>
Source	<ul style="list-style-type: none"> <li>■ Dp (default)</li> <li>■ Dn</li> <li>■ Clkp</li> <li>■ Clkn</li> </ul>
Slope	<ul style="list-style-type: none"> <li>■ Positive (default)</li> <li>■ Negative</li> <li>■ Either</li> </ul> <p>Available when Trigger Type = Transition or Edge</p>
Transition	<ul style="list-style-type: none"> <li>■ Greater Than (default)</li> <li>■ Less Than</li> </ul>
Delta Time (ps)	<ul style="list-style-type: none"> <li>■ 250</li> <li>■ 500</li> </ul>
Quality	<ul style="list-style-type: none"> <li>■ Occurs</li> <li>■ Logic</li> </ul> <p>Available when Trigger Type = Transition.</p>
Lower Level (V)	Specify the lower level in Volts. The default value is 0.05 Volts. Available when Trigger Type = Transition.
Upper Level (V)	Specify the upper level in Volts. The default value is 1 Volts. Available when Trigger Type = Transition.
Edge Trigger Level (V)	Specify the voltage level. The default value is 0.2 V. Available when Trigger Type = Edge.
Lower Limit (sec)	The default value is 25E-6. Available when Trigger Type = Width.
Upper Limit (sec)	The default value is 300E-6. Available when Trigger Type = Width.
<b>Horizontal Settings</b>	
Record Length	Specify the record length for the selected measurements. The default value is 1000000.
Sample Rate	Select the sample rate. The default value is 12.5.
<b>Vertical Settings</b>	

Setting	Description
Vertical Scale (Data)	Specify the vertical scale for data. The default value is 200 mV.
Vertical Position (Data)	Specify the vertical position for data. The default value is -2.6.
Vertical Scale (Clock)	Specify the vertical scale for clock. The default value is 200 mV.
Vertical Position (Clock)	Specify the vertical position for clock. The default value is -2.6.
Vertical Offset	Specify the vertical offset. The default value is 0 V.

### Set preferences tab parameters

Use Preferences tab to set the application action on completion of a measurement.

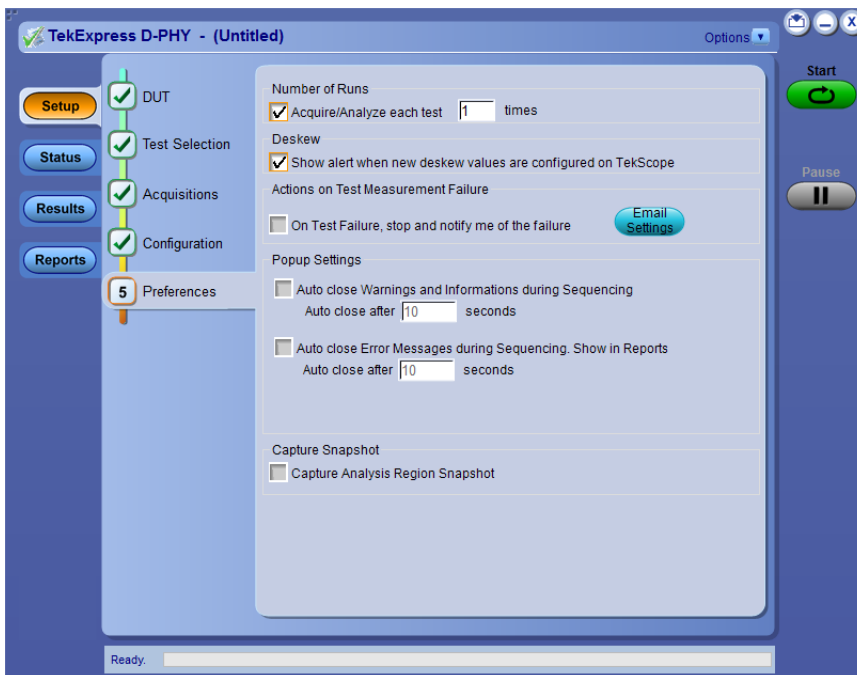


Figure 4: Preferences tab

Table 13: Preferences tab settings

Setting	Description
<b>Number of Runs</b>	
Acquire/Analyze each test <no> times (not applicable to Custom Tests)	Select to repeat the test run by setting the number of times. By default, it is selected with 1 run.
<b>Deskew</b>	
Show alert when new deskew values are configured on TekScope	Select to show alert when deskew values are configured on TekScope.
	<b>NOTE.</b> The alert is displayed when the values set on the TekScope is different from the stored values of Deskew or Attenuation.
<b>Actions on Test Measurement Failure</b>	

Setting	Description
On Test Failure, stop and notify me of the failure	Select to stop the test run on Test Failure, and to get notified via email. By default, it is unselected. Click <a href="#">Email Settings</a> to configure.  <b>NOTE.</b> Ensure that the email settings are configured correctly. If the settings are not done correctly, the test will be stopped and no other notifications will be sent.
<b>Popup Settings</b>	
Auto close Warnings and Informations during Sequencing Auto close after <no> Seconds	Select to auto close warnings/informations during sequencing. Set the Auto close time. By default it is unselected.
Auto close Error Messages during Sequencing. Show in Reports Auto close after <no> Seconds	Select to auto close Error Messages during Sequencing. Set the Auto close time. By default it is unselected.
<b>Capture Snapshot</b>	
Capture Analysis Region Snapshot	Select to capture the screenshots of the region of analysis. Select <b>Include plot images</b> in the <i>Reports</i> panel to include the captured screenshots in the reports generated.

## Status panel overview

The Status panel accesses the Test Status and Log View tabs, which provide status on test acquisition and analysis (Test Status) and a listing of test tasks performed (Log View tab). The application opens the Test Status tab when you start a test run. You can select the Test Status or the Log View tab to view these items while tests are running.

**Test Status:** The tests are grouped and displayed based on the Clock and Data lane. It displays the tests along with the Acquisition type, Acquire, and Analysis status of the tests. In pre-recorded mode, Acquire status is not valid.

**Log View:** It displays the detailed execution status of the tests.

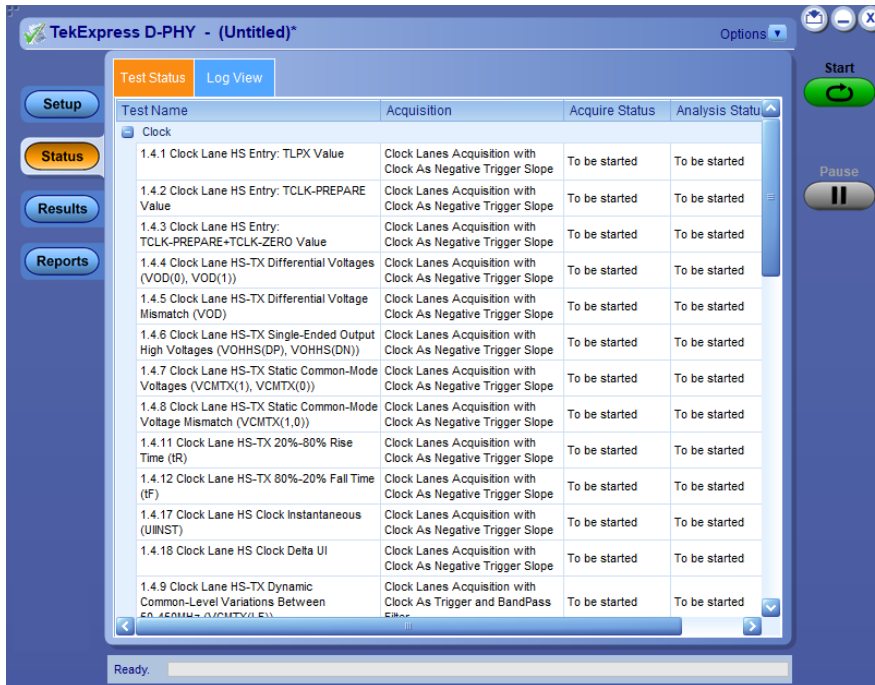


Figure 5: Test status view in Status panel

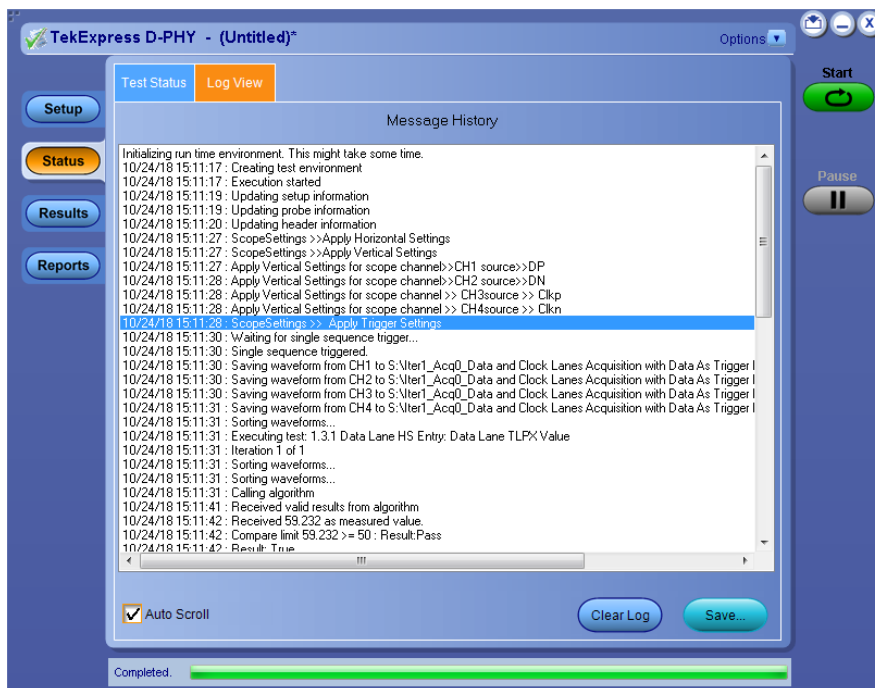


Figure 6: Log view in Status panel

Table 14: Status panel settings

Control	Description
Message History	Lists all executed test operations and timestamp information.
Auto Scroll	Enables automatic scrolling of the log view as information is added to the log during the test execution.
Clear Log	Clears all messages from the log view.
Save	Saves the log file to a text file. Use the standard Save File window to navigate to and specify the folder and file name to which to save the log text.

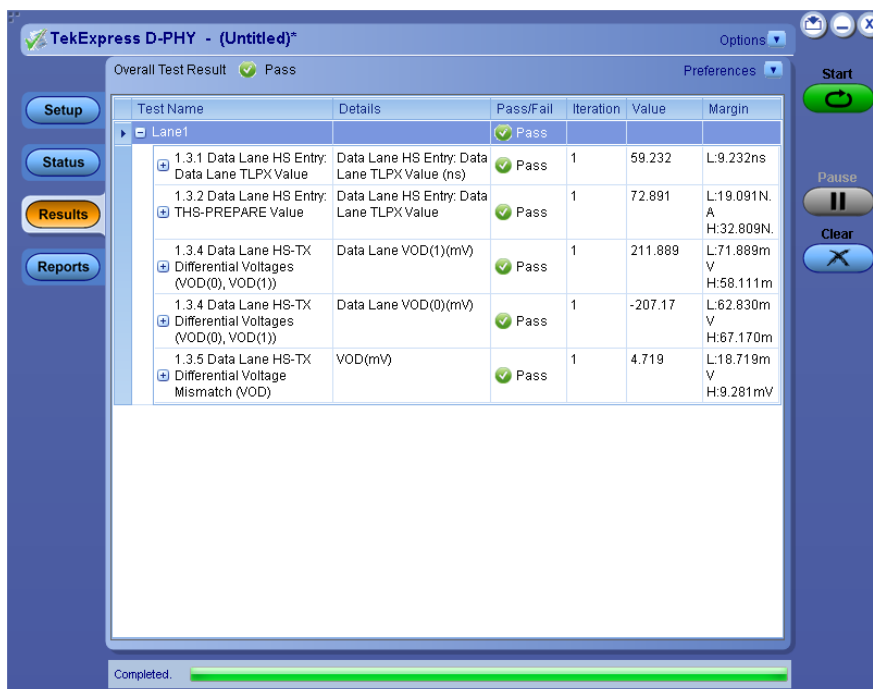
### See also

[Application panel overview](#)

## Results panel

### Results panel overview

When a test execution is complete, the application automatically opens the **Results** panel to display a summary of test results.



When a test finishes, the application switches to the Results panel, which displays a summary of test results.

Each test result occupies a row in the Results table. By default, results are displayed in summary format with the measurement details collapsed and with the Pass/Fail column visible. Change the view in the following ways:

- To remove or restore the Pass/Fail column, select Preferences > **Show Pass/Fail**.
- To collapse all expanded tests, select Preferences > **View Results Summary**.
- To expand all tests listed, select **View Results Details** from the **Preferences** menu in the upper right corner.

- To enable or disable the wordwrap feature, select Preferences > **Enable Wordwrap**.
- To view the results grouped by lane or test, select the corresponding item from the Preferences menu.
- To expand the width of a column, place the cursor over the vertical line that separates the column from the column to the right. When the cursor changes to a double-ended arrow, hold down the mouse button and drag the column to the desired width.
- To clear all test results displayed, click **Clear**.

**See also.** [View a report](#)

[Application panel overview](#)

## Preferences menu

The Preferences menu is part of the Results panel display. Use the Preferences menu to change how some items display in the Results panel.

- To include pass/fail details info in the details table, select **Show Pass/Fail**.
- To view the results summary, select **View Results Summary**.
- To expand all tests listed, select **View Results Details**
- To enable or disable the wordwrap feature, select **Enable Wordwrap**.
- To group the tests by lane, select **Group by Lane**.
- To group the tests, select **Group by Test**.

**See also.** [Results panel overview](#)

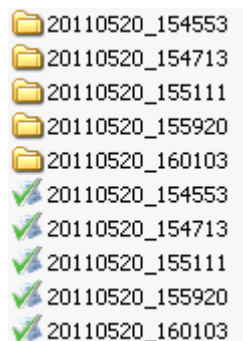
## View test-related files

Files related to tests are stored in My TekExpress\D-PHY\Untitled session folder. Each test setup in this folder has both a test setup *file* and a test setup folder, both with the test setup name.

The test setup file is preceded by the TekExpress icon and usually has no visible file name extension.

Inside the test setup folder is another folder named for the DUT ID used in the test sessions. The default is DUT001.

Inside the DUT001 folder are the session folders and files. Each session also has a folder and file pair, both named for the test session using the naming convention (date)\_(time). Each session file is stored outside its matching session folder:



Each session folder contains image files of any plots generated from running the test session. If you selected to save all waveforms or ran tests using prerecorded waveform files, these are included here.

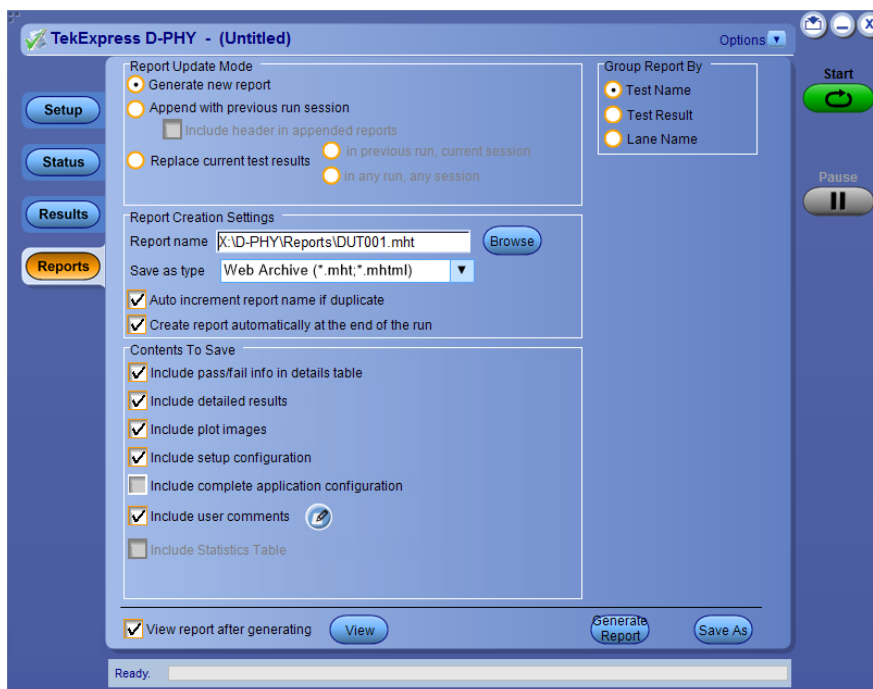
The first time you run a new, unsaved session, the session files are stored in the Untitled Session folder located at X:\D-PHY. When you name and save the session, the files are placed in a folder with the name that you specify. A copy of the test files stay in the Untitled Session folder until you run a new test or until you close the D-PHY application.

See also. [File name extensions](#)

## Reports panel

### Reports panel overview

Use Reports panel to configure report generation settings parameters, view the report, generate the report, browse for reports, name and save reports, select test content to include in reports, and select report viewing options.



For information on setting up reports, see [Select report options](#). For information on viewing reports, see [View a report](#).

See also. [View a report](#)

[Application panels overview](#)


### Select report options

Click Reports panel and use the Reports panel controls to select which test result information to include in the report, and the naming conventions to use for the report. For example, always give the report a unique name or select to have the same name increment each time you run a particular test.

Select report options before running a test or when creating and saving test setups. Report settings are included in saved test setups.

In the Reports panel, select from the following report options:

**Table 15: Report options**

Setting	Description
<b>Report Update Mode</b>	
Generate new report	Creates a new report. The report can be in either .mht or .pdf file formats.
Append with previous run session	Appends the latest test results to the end of the current test results report.
Include header in appended reports	Select to include header in appended reports
Replace current test in previous run session	Replaces the previous test results with the latest test results. Results from newly added tests are appended to the end of the report.
In previous run, current session	Select to replace current test results in the report with the test result(s) of previous run in current session.
In any run, any session	Select to replace current test results in the report with the test result(s) in selected run session's report. Click  and select test result of any other run session.
<b>Report Creation Settings</b>	
Report name	<p>Displays the name and location from which to open a D-PHY report. The default location is at <i>My TekExpress\D-PHY\Untitled Session</i>. The report file in this folder gets overwritten each time you run a test unless you specify a unique name or select to auto increment the report name.</p> <p>Change the report name or location.</p> <p>Do one of the following:</p> <ul style="list-style-type: none"> <li>■ In the Report Path field, type over the current folder path and name.</li> <li>■ Double-click in the Report Path field and then make selections from the popup keyboard and click the <b>Enter</b> button.</li> </ul> <p>Be sure to include the entire folder path, the file name, and the file extension. For example: C:\Documents and Settings\your user name\My Documents\My TekExpress\D-PHY\DUT001.mht.</p> <p><b>NOTE.</b> You cannot set the file location using the Browse button.</p> <p>Open an existing report.</p> <p>Click <b>Browse</b>, locate and select the report file and then click <b>View</b> at the bottom of the panel.</p>
Save as type	<p>Saves a report in the specified file type, selected from the drop-down list. The report are saved in .csv, .pdf or .mht.</p> <p><b>NOTE.</b> If you select a file type different from the default, be sure to change the report file name extension in the Report Name field to match.</p>
Auto increment report name if duplicate	Sets the application to automatically increment the name of the report file if the application finds a file with the same name as the one being generated. For example: DUT001, DUT002, DUT003. This option is enabled by default.
Create report automatically at the end of the run	Creates report at the end of the run.



Setting	Description
<b>Contents To Save</b>	
Include pass/fail info in details table	Includes pass/fail info in the details table of the report.
Include plot images	Includes the plot images in the report.
Include detailed results	Includes detailed results in the report.
Include setup configuration	Sets the application to include hardware and software information in the summary box at the top of the report. Information includes: the oscilloscope model and serial number, the oscilloscope firmware version, and software versions for applications used in the measurements.
Include complete application configuration	Select to complete application configuration.
Include user comments	Select to include any comments about the test that you or another user added in the DUT tab of the Setup panel. Comments appear in the Comments section, under the summary box at the beginning of each report.
Include statics table	Select to include test run statistics in the report. This is enabled when you run any test for more than once. Set <b>Acquire/Analyze each test</b> in the Preferences tab to more than one to run any test for multiple times.
<b>Group Report By</b>	
Test Name	Select to group the tests in the report by test name.
Test Result	Select to group the tests in the report by test results
Lane Name	Select to group the tests by the lane name.
View report after generating	Automatically opens the report in a Web browser when the test completes. This option is selected by default.
View	Click to view the most current report.
Generate Report	Generates a new report based on the current analysis results.
Save As	Specify a name for the report.

## View a report

The application automatically generates a report when test execution is complete and displays the report in your default Web browser (unless you cleared the **View Report After Generating** check box in the Reports panel before running the test). If you cleared this check box, or to view a different test report, do the following:

1. Click the **Reports** button.
2. Click the **Browse** button and locate and select the report file to view.
3. In the Reports panel, click **View**.

---

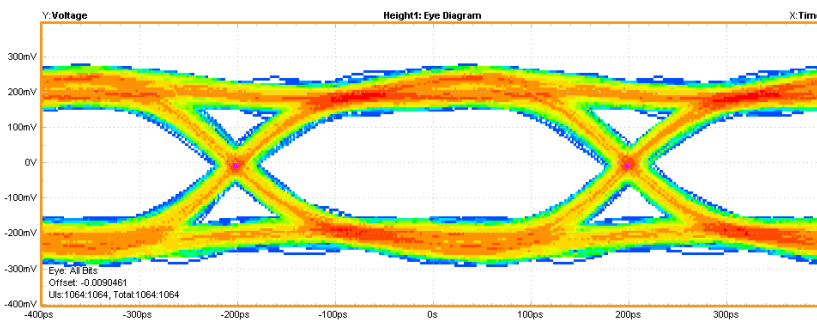
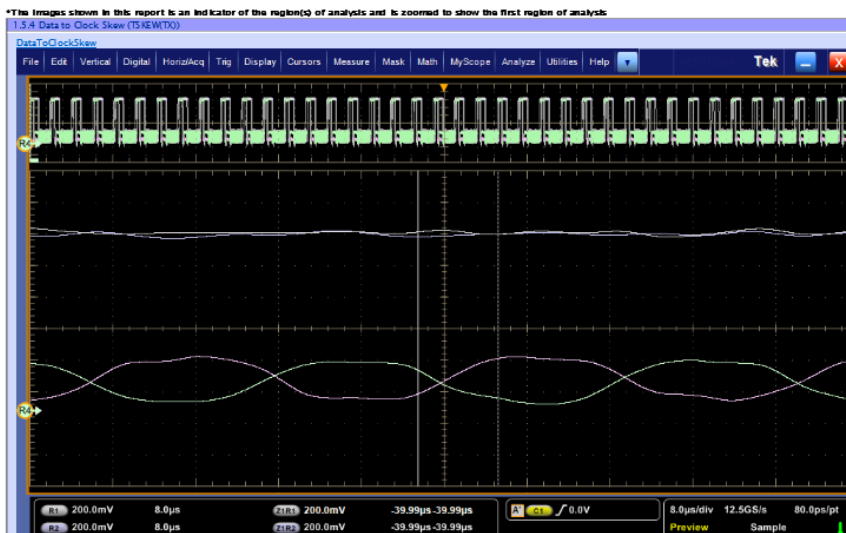
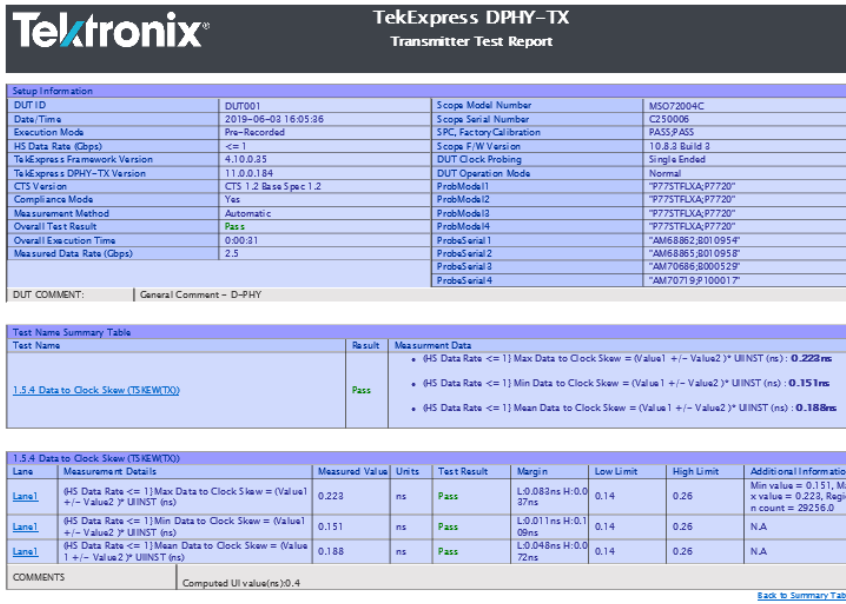
**NOTE.** *Iteration column will be displayed only when more than one test run is selected.*

---

For information on changing the file type, file name, and other report options, see [Select report options](#).

## Report content

A report shows detailed results and plots, as set in the Reports panel.



Setup configuration information

The summary box at the beginning of the report lists setup configuration information. This information includes the oscilloscope model and serial number, optical module model and serial number, and software version numbers of all associated applications.

To exclude this information from a report, clear the **Include Setup Configuration** check box in the Reports panel before running the test.

#### User comments

If you selected to include comments in the test report, any comments you added in the DUT tab are shown at the top of the report.

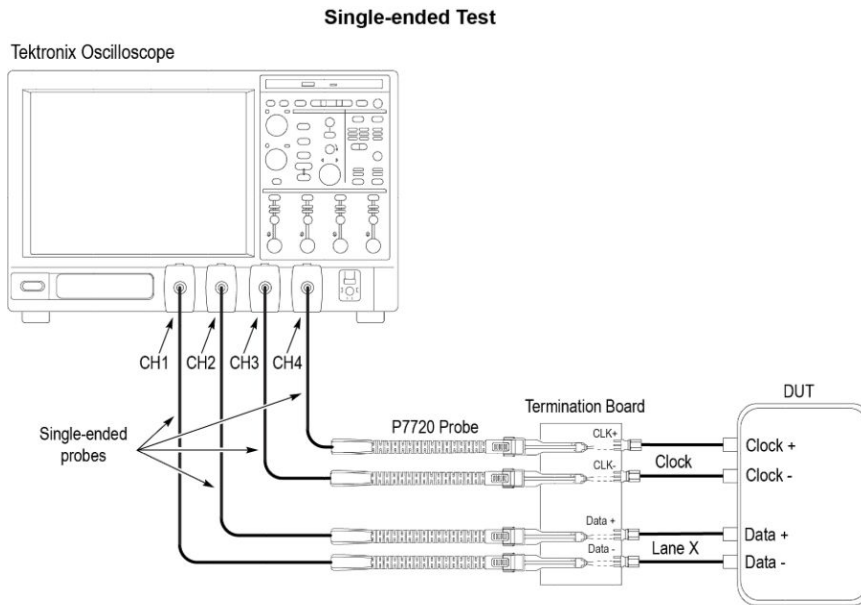
**See also.** [Results panel overview](#)

[View test-related files](#)



# Running tests

## Equipment connection diagram

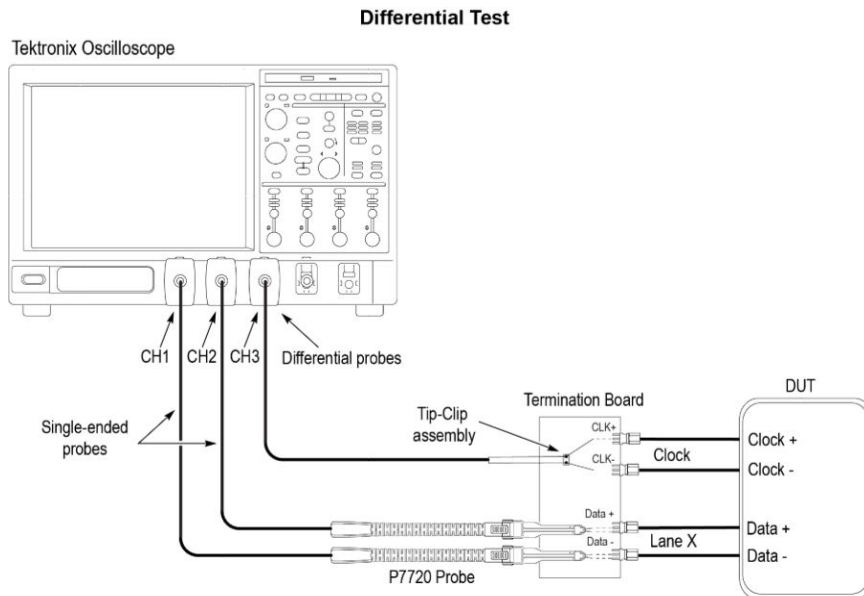


---- Indicates connection wire

X- The lane number of the terminator board selected.

Note: If you have a partial setup, such as only Tx and no Rx, then the probing board should be connected to the termination board, which provides proper termination for the LP and HS signals. If you have a full setup, then there is no need for the termination board.

0514-006



---- Indicates connection wire

X - The lane number of the terminator board selected.

Note: If you have a partial setup, such as only Tx and no Rx, then the probing board should be connected to the termination board, which provides proper termination for the LP and HS signals. If you have a full setup, then there is no need for the termination board.

0514-007

## Prerequisite

### Compensate the signal path

Use the following procedure to compensate the internal signal acquisition path. Perform this procedure if the ambient temperature has changed more than 5 °C (9 °F) since you performed the last signal path compensation. Perform the signal path compensation once a week. Failure to do so may result in the instrument not meeting warranted performance levels.

1. Power on and wait for the instrument to complete its warm up period before continuing with this procedure.
2. Disconnect any probes you have connected to the input channels.
3. Set the instrument to Menu mode.
4. Select Instrument Calibration from the Utilities menu.
5. Note any instructions that appear in the resulting control window.
6. Click Run SPC to begin the procedure. The procedure may take several minutes to complete.
7. Verify that the Status changes to Compensated after the procedure is complete. If the Calibration Status field indicates anything other than Compensated, see Signal Path Compensation Status for information on the readout and recommended action.

---

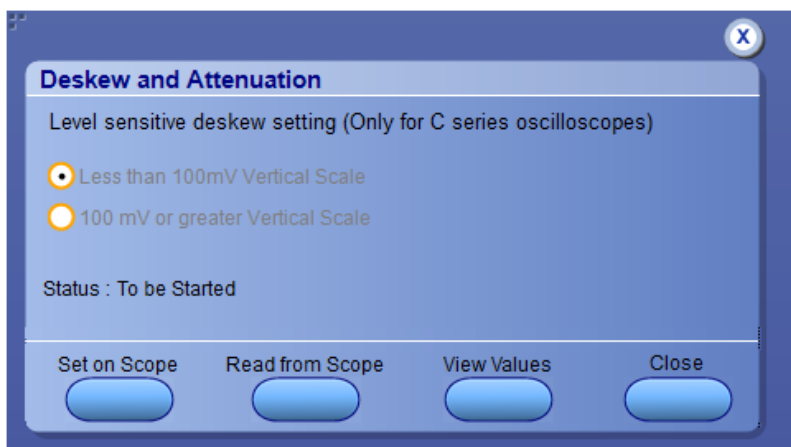
**NOTE.** When making measurements at vertical scale settings less than or equal to 5 mV, you should perform the signal path compensation at least once a week. Failure to do so may result in the instrument not meeting warranted performance levels at those volts/div settings.

---

## Deskew

If skew is present between positive and negative channels, then the channels need to be deskewed before being used for waveform measurements. TekExpress D-PHY provides support for channel deskew and attenuation using the following method:

1. Determine what the skew is for each channel.
2. From the TekScope menu, select Vertical > Deskew.
3. In the Deskew/Attenuation window, click the channel (1 – 4) button for the first channel to be deskewed.
4. Click in the Ch(x) Deskew Time entry field and enter the skew. The skew can be +ve or –ve.
5. Click the channel button for the next channel and repeat step 1.
6. After entering the skew for all the channels that require it, from the Options menu in TekExpress D-PHY, select Deskew.
7. In the Deskew dialog box, select the desired level (applicable for C series oscilloscopes only):
  - Less than 100 mV signal amplitude: Select this if the signal amplitude is such that the oscilloscope's vertical setting is less than 100 mV/division.
  - 100 mV or greater signal amplitude: Select this if the signal amplitude is such that the oscilloscope's vertical setting is greater than 100 mV/division.



**Figure 7: Deskew**

8. Click **Read from Scope**.
9. When the status in the dialog box indicates the deskew is finished, click Close.

Each input channel has its own deskew settings. Deskew compensates individual channels for probes or cables of different lengths. The instrument applies the delay values after each completed acquisition. The deskew values are saved as part of the instrument setup. The deskew values for the selected channel are retained until you change the probe, you restore a saved setup, or you recall the factory setup.

## Running tests

*Set DUT parameters, select tests, set acquisition parameters, set configuration parameters, set preferences parameters,* and click **Start** to run the tests. While tests are running, you cannot access the Setup or Reports panels. To monitor the test progress, switch between the Status panel and the Results panel.

While tests are running, the other applications will be displayed at the background. If you want the TekExpress D-PHY application to run in the foreground select **Keep On Top** from the TekExpress Options menu.

The application displays report when the tests execution is complete.


### Prerun checklist

1. Make sure that the instruments are warmed up (approximately 20 minutes) and stabilized.
2. Perform compensation: In the oscilloscope main menu, select **Utilities > Instrument Compensation**. Click **Help** in the compensation window for steps to perform instrument compensation.

## View test results

When a test completes, the application switches to the Results panel, which shows a summary of test results.

Each test result occupies a row in the Results table. By default, results are displayed in summary format, with the measurement details collapsed. You can change the view in the following ways:

- To view the results grouped by lane, test, or data rate, select the corresponding item from the Preferences menu.
- To expand all tests listed, select **View Results Details** from the Preferences menu.
- To expand and collapse tests, use the plus and minus buttons to the left of the test rows.
- To collapse all expanded tests, select **Preferences > View Results Summary**.
- To enable or disable the wordwrap feature, select **Preferences > Enable Wordwrap**.
- To expand the width of a column, place the cursor over the vertical line that separates the column from the one to the right. When the cursor changes to a double-ended arrow, hold down the mouse button and drag the column to the desired width.
- To sort the test information by column, click the column head. When sorted in ascending order, a small up arrow is displayed. When sorted in descending order, a small down arrow is displayed.
- To clear all test results displayed, click **Clear** ()



---

# Saving and recalling test setup

## Test setup files overview

Saved test setup information (such as the selected oscilloscope, general parameters, acquisition parameters, measurement limits, waveforms (if applicable), and other configuration settings) are saved under the setup name at **X:\D-PHY**.

Use test setups to:

- Run a new session, acquiring live waveforms, using a saved test configuration.
- Create a new test setup using an existing one.
- View all the information associated with a saved test, including the log file, the history of the test status as it executed, and the results summary.
- Run a saved test using saved waveforms.

### See also

[Save a test setup](#)

[Open \(load\) a saved test setup](#)

## Save a test setup

You can save a test setup before or after running a test. You can create a test setup from [already created test setup](#), or using [default test setup](#). When you select the default test setup, the parameters are set to the application's default value.

Select **Options > Save Test Setup** to save the opened setup.

Select **Options > Save Test Setup As** to save the setup with different name.

## Open load a saved test setup

To Open (load) a saved test setup, do the following:

1. Select **Options > Open Test Setup**.
2. Select the setup from the list and click **Open**. Setup files are located at **X:\D-PHY\**.

### See also

[About test setups](#)

[Create a test setup using an existing one](#)

[Create a test setup from default settings](#)

## Create a test setup from default settings

To create a test setup using default settings, follow the steps:

1. Select **Options > Default Test Setup**. For default test setup, the parameters are set to the application's default value.
2. Click application [Setup](#) and set the parameters
3. Click application [Reports](#) and set the report options
4. Optional: Click **Start** to run the test and verify that it runs correctly and captures the specified test information and reports. If it does not, then edit the parameters and repeat this step until the test runs to your satisfaction
5. Select **Options > Save Test Setup**. Enter the file name and click Save. The application saves the file to X:\D-PHY \*session\_name*>

## Create a test setup using an existing one

To create a test setup using an existing one, follow the steps:

1. Select **Options > Open Test Setup**
2. Select a setup from the list and then click **Open**
3. Click application setup and modify the parameters
4. Click application reports and modify the report options
5. Select **Options > Save Test Setup As**
6. Enter test setup name, and click **Save**.

# TekExpress programmatic interface

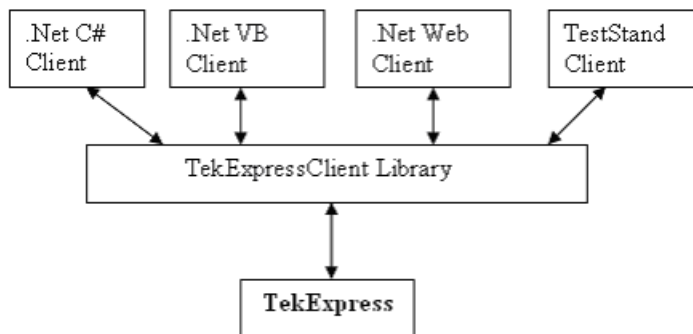
## About the programmatic interface

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. See [Server and client proxy objects](#).

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 *Idlglib.dll* and *IRemoteInterface.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:

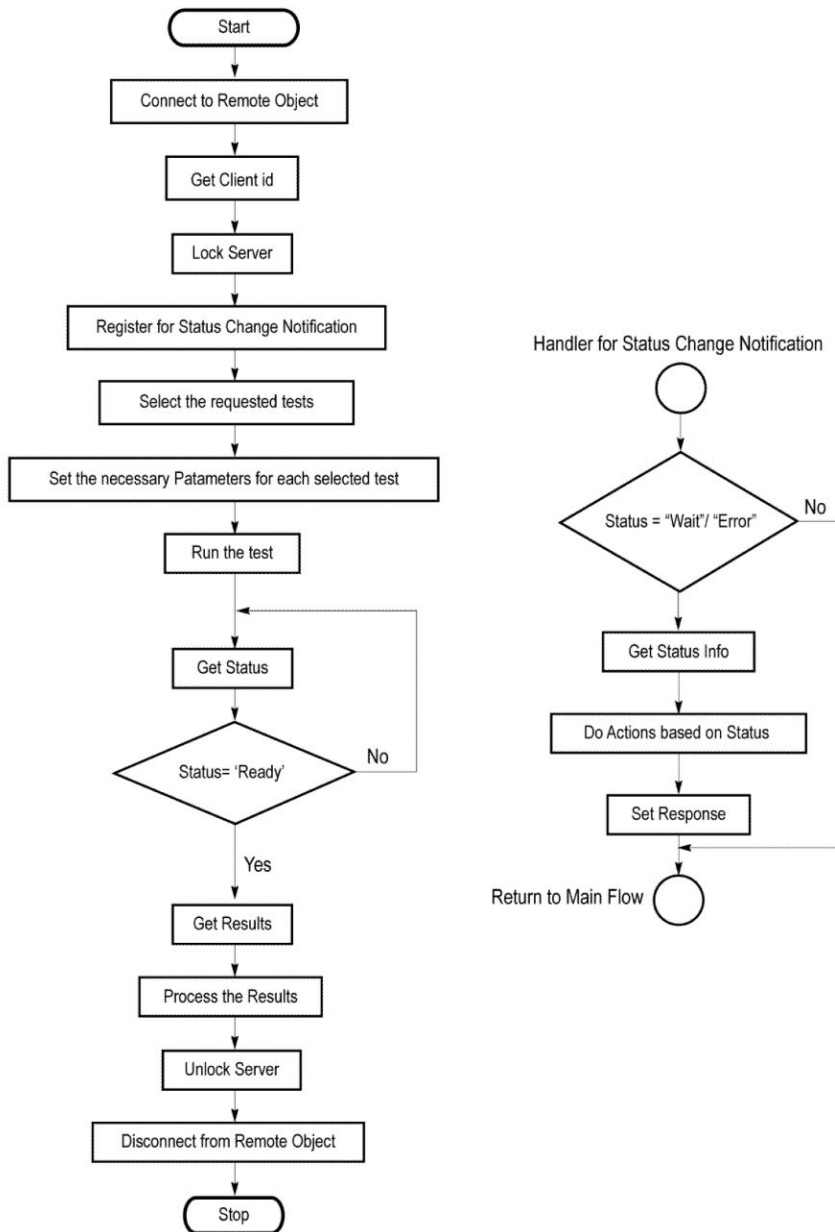
A client UI must be developed to access the interfaces exposed through the server. This client needs to load `TekExpressClient.dll` to access the interfaces. Once the `TekExpressClient.dll` is loaded, the client UI can call the specific functions to run the operations requested by the client. Once the 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.

## 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](#)

[Lock the server](#)

[Disable the popups](#)

[Set or get the DUT ID](#)

[Set the configuration parameters for a suite or measurement](#)

[Query the configuration parameters for a suite or measurement](#)

[Select a measurement](#)

Select a suite

Select a channel

Configure the selected measurement

Run with set configurations or stop the run operation

Get or set the timeout value

Wait for the measurement to complete

After the measurement is complete

Save, recall, or check if a session is saved

Unlock the server

Disconnect from the server

Select a lane

Assign input signals to scope channels

Handle error codes

### Connect through an IP address

Command name	Parameters	Description	Return Value	Example
Connect()	string ipAddress out string clientID	<p>This method connects the client to the server.</p> <p>Note</p> <p><b>NOTE.</b> 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.</p> <p>The client provides the IP address to connect to the server. 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>

**string ipAddress**

Name	Type	Direction	Description
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.

**out string clientID**

Name	Type	Direction	Description
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.** The Fail condition for this command occurs in the following conditions:

*If the server is LOCKED the command returns "Server is locked by another client".*

*If the session is UNLOCKED the command returns "Lock Session to execute the command".*

*If the server is NOTFOUND the command returns "Server not found...Disconnect!".*

*If none of these fail conditions occur the command returns "Failed...".*

## Lock the server

Command name	Parameters	Description	Return Value	Example
LockSession()	string clientID	<p>This method locks the server. Note</p> <p><b>NOTE.</b> 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.</p> <p>The client must call this method before running any of the remote automations. The server can be locked by only one client.</p>	<p>String value that gives the status of the operation after it has been performed. The return value is "Session Locked..." on success.</p>	<pre>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)</pre>

### out string clientID

Name	Type	Direction	Description
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.** The Fail condition for this command occurs in the following conditions:

If the server is **LOCKED** the command returns "Server is locked by another client".

If the session is **UNLOCKED** the command returns "Lock Session to execute the command".

If the server is **NOTFOUND** the command returns "Server not found...Disconnect!".

If none of these fail conditions occur the command returns "Failed...".



## Disable the popups

Command name	Parameters	Description	Return Value	Example
SetVerboseMode()	string clientID bool _verbose	This method sets the verbose mode to either true or false. When the value is set to true, you need to handle the pop-ups. When the value is set to false, then the application handles the popup automatically. The default selections are considered as response and the execution proceeds.	String that gives the status of the operation after it has been performed. When Verbose mode is set to true, the return value is "Verbose mode turned on. All dialog box will be shown to client ...". When Verbose mode is set to false, the return value is "Verbose mode turned off. All dialog box will be shown to server ...".	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string <b>Verbose mode is turned on</b> return=m_Client.SetVerboseMode(clientID, true) <b>Verbose mode is turned off</b> returnval=m_Client.SetVerboseMode(clientID, false)

### out string clientID

Name	Type	Direction	Description
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

### bool \_verbose

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

**NOTE.** The Fail condition for this command occurs in the following conditions:

If the server is **LOCKED** the command returns "Server is locked by another client".

If the session is **UNLOCKED** the command returns "Lock Session to execute the command".

If the server is **NOTFOUND** the command returns "Server not found...Disconnect!".

If none of these fail conditions occur the command returns "Failed...".

## Set or get the DUT ID

Command name	Parameters	Description	Return Value	Example
SetDutId()	string clientID string dutName	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) Note</pre> <p><b>NOTE.</b> If the <i>dutName</i> parameter is null, the client is prompted to provide a valid DUT ID.</p>
GetDutId()	string clientID string dutId	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>

### out string clientID

Name	Type	Direction	Description
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

### string dutName

Name	Type	Direction	Description
dutName	string	IN	The new DUT ID of the setup.

### string dutId

Name	Type	Direction	Description
dutId	string	OUT	The DUT ID of the setup.

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

---

**NOTE.** The Fail condition for this command occurs in the following conditions:

If the server is LOCKED the command returns "Server is locked by another client".

If the session is UNLOCKED the command returns "Lock Session to execute the command".

If the server is NOTFOUND the command returns "Server not found...Disconnect!".

If none of these fail conditions occur the command returns "Failed...".

---

## Set the configuration parameters for a suite or measurement

---

**NOTE.** The strings required for these examples can be referred from the [Command parameters list](#).

---

Command name	Parameters	Description	Return Value	Example
SetGeneralParameter	string clientID string device string suite string test string parameterString	This method sets the number of video lanes for the selected measurement.  <i><b>NOTE.</b> Using this command we can select a lane, channel, or source type.</i>	String that gives the status of the operation after it has been performed. 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 returnval as string Select Channel  <b>Select Channel Example</b>  returnval = m_Client.SetGeneralParameter(clientID, "D-PHY", "Transmitter", "", "Lane1:Lane1\$Included") Select Measurement Method  <b>Select Measurement Method Example</b>  returnval=mClient.SetGeneralParameter(clientID, "D-PHY", "Transmitter", "", "Measurement Method\$Automatic")
SetAnalyzeParameter()	string clientID string device string suite string test string parameterString	This method sets the configuration parameters in the Configuration Panel, Measurements tab for each 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 returnval as string
SetAcquireParameter()	string clientID string device string suite string test string parameterString	This method sets the configuration parameters in the Acquire panel of the Configuration Panel dialog box for a given suite or measurement.	returnVal = remoteObject.SetAcquireParameter(id, device, suite, test, parameterString) if ((OP_STATUS) returnVal != OP_STATUS.SUCCESS) return CommandFailed(returnVal)	Configure Acquire Parameters for Data Lane Rise Time <b>Example</b> string returnval = m_Client.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Vertical Scale (mV)\$400")

**out string clientID**

Name	Type	Direction	Description
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

**string device**

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device.

**string suite**

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite.

**string test**

Name	Type	Direction	Description
test	string	IN	Specifies the name of the test for acquire parameter.

**string parameterString**

Name	Type	Direction	Description
parameterString	string	IN	Specifies the acquire type, acquire parameter name, and value.

**NOTE.** The Fail condition for this command occurs in the following conditions:

*If the server is LOCKED the command returns "Server is locked by another client".*

*If the session is UNLOCKED the command returns "Lock Session to execute the command".*

*If the server is NOTFOUND the command returns "Server not found...Disconnect!".*

*If none of these fail conditions occur the command returns "Failed...".*

**Configure Parameter example for Acquisition Parameters for 1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)**

Parameter	Example
Trigger Type	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Trigger Type\$Transition")
Trigger Source	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Trigger Source\$Dp")
Trigger Slope	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$"Trigger Slope\$Positive)
Trigger Transition	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Trigger Transition\$Greater Than")
Delta Time (ps)	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Trigger Time (pS)\$500")
Qualify	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Quaify\$Occurs")
Lower Level (V)	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Lower Level (V)\$0.05")
Upper Level (V)	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Upper Level (V)\$1")
Trigger If Violation	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Trigger If Violation\$Occurs")
Record Length	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Record Length\$1000000")
Sample Rate (GS/s)	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode withData As Trigger\$Sample Rate (GS/s)\$12.5")

Parameter	Example
Vertical Scale (mV)	<code>returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode withData As Trigger\$Vertical Scale (mV)\$200")</code>
Vertical Position (div)	<code>returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode withData As Trigger\$Vertical Position (div)\$-2.6")</code>
Vertical Offset (V)	<code>returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode withData As Trigger\$Vertical Offset (V)\$0")</code>

## Query the configuration parameters for a suite or measurement

Command name	Parameters	Description	Return Value	Example
GetGeneralParameter()	string clientID string device string suite string test string parameterString	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.	<p>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string Query Channel</p> <p><b>Query Channel Example</b></p> <p>returnval=mClient.GetGeneralParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Lane5 Connected to:Data:Dn")</p> <p><b>Query Measurement Method</b></p> <p><b>Query Measurement Method Example</b></p> <p>returnval=mClient.GetGeneralParameter(clientID, "D-PHY", "Transmitter", "Data Lane LP-TX Thevenin Output High Level Voltage (VOH)")</p>
GetAnalyzeParameter()	string clientID string device string suite string test string parameterString	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.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string
GetAcquireParameter()	string clientID string device string suite string test string parameterString	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.	Query Acquire Parameters for Data Lane LP-TX Thevenin Output High Level Voltage (VOH)



**out string clientID**

Name	Type	Direction	Description
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

**string device**

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device.

**string suite**

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite.

**string test**

Name	Type	Direction	Description
test	string	IN	Specifies the name of the test for acquire parameter.

**string parameterString**

Name	Type	Direction	Description
parameterString	string	IN	Specifies the acquire parameter for acquire type, acquire parameter name, and value.

**NOTE.** The Fail condition for this command occurs in the following conditions:

If the server is **LOCKED** the command returns "Server is locked by another client".

If the session is **UNLOCKED** the command returns "Lock Session to execute the command".

If the server is **NOTFOUND** the command returns "Server not found...Disconnect!".

If none of these fail conditions occur the command returns "Failed...".

**Query Data Lane LP-TX Thevenin Output High Level Voltage (VOH) Acquire Parameter Examples**

Parameter	Example
Trigger Type	returnval = mClient.GetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Trigger Type")
Trigger Source	returnval = mClient.GetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Trigger Source")
Slope	returnval = mClient.GetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Slope")
Transition	returnval = mClient.GetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Transition")
Delta Time (ps)	returnval = mClient.GetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Delta Time (pS)")
Qualify	returnval = mClient.GetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Qualify")
Lower Level (V)	returnval = mClient.GetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Lower Level (V)")
Upper Level (V)	returnval = mClient.GetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Upper Level (V)")
Record Length	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Record Length\$")
Sample Rate (GS/s)	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Sample Rate (GS/s)")
Vertical Scale (mV)	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Vertical Scale (mV)")

Parameter	Example
Vertical Position (div)	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lane LP-TX Thevenin Output High Level Voltage", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Vertical Position (div)")
Vertical Offset (V)	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Vertical Offset (V)")

### Select a measurement

Command name	Parameters	Description	Return Value	Example
SelectTest()	string clientID string device string suite string test bool isSelected	This method selects or deselects a given test. Setting parameter isSelected to true, you can select a measurement. Setting parameter isSelected to false, you can deselect a measurement.	String that displays the status of the operation after it has been performed. 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 returnval as string <b>Select Measurement (Data Lane Rise Time):</b> returnval=m_Client.SelectTest(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", true)

#### out string clientID

Name	Type	Direction	Description
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

#### string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device.

#### string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite.

**string test**

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

**bool isSelected**

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

**NOTE.** The Fail condition for this command occurs in the following conditions:

If the server is **LOCKED** the command returns "Server is locked by another client".

If the session is **UNLOCKED** the command returns "Lock Session to execute the command".

If the server is **NOTFOUND** the command returns "Server not found...Disconnect!".

If none of these fail conditions occur the command returns "Failed...".

**Select a single measurement**

Command name	Parameters	Description	Return Value	Example
SelectSingleTest()	string clientID string device string suite string version	This method will deselect all tests and only the test mentioned in the command will get selected.	String that displays the status of the operation after it has been performed. 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 returnval as string <b>Select Single Measurement (1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)):</b> Example :m_client=new Client//m-client is a reference to the Client class in the Client DLL Retrunval=m_Client.selectSingleTest(clientID,"D-PHY","Transmitter","CTS 1.2 Base Spec 1.2","1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)")

**out string clientID**

Name	Type	Direction	Description
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

**string device**

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device.

**string suite**

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite.

**string version**

Name	Type	Direction	Description
version	string	IN	Specifies the CTS version of the application.

**NOTE.** The Fail condition for this command occurs in the following conditions:

If the server is LOCKED the command returns "Server is locked by another client".

If the session is UNLOCKED the command returns "Lock Session to execute the command".

If the server is NOTFOUND the command returns "Server not found...Disconnect!".

If none of these fail conditions occur the command returns "Failed...".

**Select a suite**

Command name	Parameters	Description	Return Value	Example
SelectSuite()	string clientID string device string suite bool isSelected	This method selects or deselects a given suite. Setting parameter isSelected to true, you can select a suite. Setting parameter isSelected to false, you can deselect a suite.	String that gives the status of the operation after it has been performed. 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 returnval as string <b>Select Suite (Default):</b> returnval=m_Client.SelectTest(clientID, "D-PHY", "Transmitter", true)

**out string clientID**

Name	Type	Direction	Description
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

**string device**

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device.

**string suite**

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite.

**bool isSelected**

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

**NOTE.** The Fail condition for this command occurs in the following conditions:

If the server is **LOCKED** the command returns "Server is locked by another client".

If the session is **UNLOCKED** the command returns "Lock Session to execute the command".

If the server is **NOTFOUND** the command returns "Server not found...Disconnect!".

If none of these fail conditions occur the command returns "Failed...".

## Select a channel

Command name	Parameters	Description	Return Value	Example
SetGeneralParameter()	string clientID string device string suite string test string parameterString	This method sets the parameters that are not specific to any given test.  <i><b>NOTE.</b> Using this command we can select a lane, channel, or source type.</i>	String that gives the status of the operation after it has been performed. 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 returnval as string Select Channel Select Measurement Method
SetAnalyzeParameter()	string clientID string device string suite string test string parameterString	This method sets the configuration parameters in the Configuration Panel, Measurements tab for each 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 returnval as string
SetAcquireParameter()	string clientID string device string suite string test string parameterString	This method sets the configuration parameters in the Acquire panel of the Configuration Panel dialog box for a given suite or measurement.	returnVal = remoteObject.SetAcquireParameter(id, device, suite, test, parameterString) if ((OP_STATUS) returnVal != OP_STATUS.SUCCESS) return CommandFailed(returnVal)	Configure Acquire Parameters for Data Lane Rise Time

### out string clientID

Name	Type	Direction	Description
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

### string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device.

### string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite.

**string test**

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

**string parameterString**

Name	Type	Direction	Description
parameterString	string	IN	Specifies the acquire type, acquire parameter name, and value.

**Select Channel Example**

```
returnval = m_Client.SetGeneralParameter(clientID, "D-PHY", "Transmitter", "", "Lane1:Lane 1$Included")
```

**Select Measurement Method Example**

```
returnval=mClient.SetGeneralParameter(clientID, "D-PHY", "Transmitter", "" "Measurement Method$Automatic")
```

**Configure Parameter example for Acquisition Parameters for 1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)**

Parameter	Example
Trigger Type	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode withData As Trigger\$Trigger Type\$Transition")
Trigger Source	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode withData As Trigger\$Trigger Source\$Dp")
Trigger Slope	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode withData As Trigger\$Trigger Slope\$Positive)
Trigger Transition	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode withData As Trigger\$Trigger Transition\$Greater Than")
Delta Time (ps)	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode withData As Trigger\$Trigger Time (pS)\$500")
Qualify	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode withData As Trigger\$Quaify\$Occurs")
Lower Level (V)	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Lower Level (V)\$0.05")



Parameter	Example
Upper Level (V)	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Upper Level (V)\$1")
Record Length	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Record Length\$1000000")
Sample Rate (GS/s)	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Sample Rate (GS/s)\$12.5")
Vertical Scale (mV)	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Vertical Scale (mV)\$200")
Vertical Position (div)	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Vertical Position (div)\$-2.6")
Vertical Offset (V)	returnval = mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape Mode with Data As Trigger\$Vertical Offset (V)\$0")

**NOTE.** The Fail condition for this command occurs in the following conditions:

*If the server is LOCKED the command returns "Server is locked by another client".*

*If the session is UNLOCKED the command returns "Lock Session to execute the command".*

*If the server is NOTFOUND the command returns "Server not found...Disconnect!".*

*If none of these fail conditions occur the command returns "Failed...".*

## Configure the selected measurement

Command name	Parameters	Description	Return Value	Example
SetAnalyzeParameter()	string clientID string device string suite string test string parameterString	This method sets the Analyze parameters (Configuration parameters) for a 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 returnval as string
SetAcquireParameter()	string clientID string device string suite string test string parameterString	This method sets the Acquire parameters in the Acquire panel of the Configure Dialog box for a given test.	returnVal = remoteObject.SetAcquireParameter(id, device, suite, test, parameterString) if ((OP_STATUS) returnVal != OP_STATUS.SUCCESS) return CommandFailed(returnVal)	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string

## out string clientID

Name	Type	Direction	Description
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

## string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device.

## string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite.

## string test

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

**string parameterString**

Name	Type	Direction	Description
parameterString	string	IN	Specifies the acquire type, acquire parameter name, and value.

**NOTE.** The Fail condition for this command occurs in the following conditions:

If the server is LOCKED the command returns "Server is locked by another client".

If the session is UNLOCKED the command returns "Lock Session to execute the command".

If the server is NOTFOUND the command returns "Server not found...Disconnect!".

If none of these fail conditions occur the command returns "Failed...".

**Run with set configurations or stop the run operation**

Command name	Parameters	Description	Return Value	Example
Run()	string clientID	Runs the selected tests. Note	String that gives the status of the operation after it has been performed. The return value is "Run started..." 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.Run(clientID)
		<b>NOTE.</b> When the run is performed, the status of the run is updated periodically using a timer.		
		Once the server is set up and is configured, it can be run remotely using this function.		
Stop()	string clientID	Stops the currently running tests. Note	String that gives the status of the operation after it has been performed. The return value is "Stopped..." 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.Stop(clientID)
		<b>NOTE.</b> When the session is stopped, the client is prompted to stop the session and is stopped at the consent.		

**out string clientID**

Name	Type	Direction	Description
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.** The Fail condition for this command occurs in the following conditions:

If the server is **LOCKED** the command returns "Server is locked by another client".

If the session is **UNLOCKED** the command returns "Lock Session to execute the command".

If the server is **NOTFOUND** the command returns "Server not found...Disconnect!".

If none of these fail conditions occur the command returns "Failed...".

### Get or set the timeout value

Command name	Parameters	Description	Return Value	Example
GetTimeOut()	string clientID	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=m_Client.GetTimeOut()
SetTimeOut()	string clientID string time	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=m_Client.SetTimeOut(clientID, desiredTimeOut)

#### out string clientID

Name	Type	Direction	Description
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

#### 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.

**NOTE.** *The Fail condition for this command occurs in the following conditions:*

*If the server is LOCKED the command returns "Server is locked by another client".*

*If the session is UNLOCKED the command returns "Lock Session to execute the command".*

*If the server is NOTFOUND the command returns "Server not found...Disconnect!".*

*If none of these fail conditions occur the command returns "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()	string clientID	<p>This method gets the status of the server application. The states at a given time are Ready, Running, Paused, Wait, or Error.</p> <p>Ready: Test configured and ready to start.</p> <p>Running: Test running.</p> <p>Paused: Test paused.</p> <p>Wait: A popup that needs your inputs.</p> <p>Error: An error is occurred.</p>	String value that gives the status of the server application.	<pre>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)</pre>
QueryStatus()	string clientID out string[] status	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()	string clientID out string caption out string message out string[] buttonTexts	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>
<p><b>NOTE.</b> This command is used when the application is running and is in the wait or error state.</p>				

Command name	Parameters	Description	Return Value	Example
SendResponse()	string clientID out string caption out string message string response	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.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL m_Client.SendResponse(clientID, caption,message, response)
<p><b>NOTE.</b> This command is used when the application is running and is in the wait or error state.</p>				

**out string clientID**

Name	Type	Direction	Description
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

**out string[] status**

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

**out string caption**

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

**out string message**

Name	Type	Direction	Description
message	String	OUT	The wait state/error state message to you.

**out string[] buttonTexts**

Name	Type	Direction	Description
buttonTexts	string array	OUT	An array of strings containing the possible response types that you can send.

**string response**

Name	Type	Direction	Description
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).

**NOTE.** The Fail condition for this command occurs in the following conditions:

*If the server is LOCKED the command returns "Server is locked by another client".*

*If the session is UNLOCKED the command returns "Lock Session to execute the command".*

*If the server is NOTFOUND the command returns "Server not found...Disconnect!".*

*If none of these fail conditions occur the command returns "Failed...".*



## After the test is complete

Command name	Parameters	Description	Return Value	Example
GetPassFailStatus()	string clientID string device string suite string test	This method gets the pass or fail status of the measurement after test completion.  <i><b>NOTE.</b> Execute this command after completing the measurement.</i>	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.	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, "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)") //Pass or Fail
GetResultsValue()	string clientID string device string suite string test string parameterString	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.	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", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Measured Value")

Command name	Parameters	Description	Return Value	Example
GetResultsValueForSubMeasurements()	string clientID string device string suite string test string parameterString int rowNr	This method gets the result values for individual sub-measurements, 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 <b>Data Lane Rise Time DP (nS)</b> returnval=m_Client.GetResultsValueForSubMeasurements(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Measured Value",0) // For DP wfm returnval=m_Client.GetResultsValueForSubMeasurements(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)", "Measured Value",1) // For DN wfm</pre>
GetReportParameter()	string clientID string device string suite string test string parameterString	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.GetReportParameter(clientID, "Scope Model") <b>TekExpress Version</b> returnval=m_Client.GetReportParameter(clientID, "TekExpress Version") <b>D-PHY Version</b> returnval=m_Client.GetReportParameter(clientID, "Application Version")</pre>

Command name	Parameters	Description	Return Value	Example
TransferReport()	string clientID string filePath	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.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string returnval=m_Client.TransferReport(clientID,"C:\Report")
TransferWaveforms()	string clientID string filePath	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...".	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string returnval=m_Client.TransferWaveforms(clientID,"C:\Waveforms")
TransferImages()	string clientID string filePath	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.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string returnval=m_Client.TransferImages(clientID, "C:\Waveforms")

**out string clientID**

Name	Type	Direction	Description
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

**string device**

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device.

**string suite**

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite.

**string test**

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

**string parameterString**

Name	Type	Direction	Description
parameterString	string	IN	Specifies the acquire type, acquire parameter name, and value.

**int rowNr**

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

**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.

**string parameterString**

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

**NOTE.** The Fail condition for this command occurs in the following conditions:

If the server is **LOCKED** the command returns "Server is locked by another client".

If the session is **UNLOCKED** the command returns "Lock Session to execute the command".

If the server is **NOTFOUND** the command returns "Server not found...Disconnect!".

If none of these fail conditions occur the command returns "Failed...".

## Save, recall, or check if a session is saved

Command name	Parameters	Description	Return Value	Example
CheckSessionSaved()	string clientID out bool saved	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()	string clientID string name	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()	string clientID string name	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..."/"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()	string clientID string name	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)

### out string clientID

Name	Type	Direction	Description
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

### 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.

**out bool saved**

Name	Type	Direction	Description
saved	bool	OUT	Boolean representing whether the current session is saved.

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

**NOTE.** The Fail condition for this command occurs in the following conditions:

If the server is LOCKED the command returns "Server is locked by another client".

If the session is UNLOCKED the command returns "Lock Session to execute the command".

If the server is NOTFOUND the command returns "Server not found...Disconnect!".

If none of these fail conditions occur the command returns "Failed...".

**Unlock the server**

Command name	Parameters	Description	Return Value	Example
UnlockSession()	string clientID	This method unlocks the server from the client. The ID of the client to be unlocked must be provided. Note  <b>NOTE.</b> When the client is disconnected, it is unlocked from the server and then disconnected. The id is reused.	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.UnlockServer(clientID)

**out string clientID**

Name	Type	Direction	Description
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.** The Fail condition for this command occurs in the following conditions:

If the server is LOCKED the command returns "Server is locked by another client".

If the session is UNLOCKED the command returns "Lock Session to execute the command".

If the server is NOTFOUND the command returns "Server not found...Disconnect!".

If none of these fail conditions occur the command returns "Failed...".

## Disconnect from the server

Command name	Parameters	Description	Return Value	Example
Disconnect()	string clientID	This method disconnects the client from the server it is connected to. Note  <b>NOTE.</b> When the client is disconnected, it is unlocked from the server and then disconnected. The id is reused.	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.Disconnect(m_clientID)

### out string clientID

Name	Type	Direction	Description
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.** The Fail condition for this command occurs in the following conditions:

If the server is LOCKED the command returns "Server is locked by another client".

If the session is UNLOCKED the command returns "Lock Session to execute the command".

If the server is NOTFOUND the command returns "Server not found...Disconnect!".

If none of these fail conditions occur the command returns "Failed...".

## Select a lane

Command name	Parameters	Description	Return Value	Example
SetGeneralParameter()	string clientID string device string suite string test string parameterString	This method sets the parameters that are not specific to any given test. Note. Using this command you 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.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string Select/Deselect a Lane Deselect a Lane

## out string clientID

Name	Type	Direction	Description
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

## string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device.

## string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite.

## string test

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

## string parameterString

Name	Type	Direction	Description
parameterString	string	IN	Specifies the acquire type, acquire parameter name, and value.



**NOTE.** To select Lane 1, the parameterString is "Lane1:Lane 1\$Included", To deselect it – "Lane1:Lane 1\$Excluded"

To select Lane 2, the parameterString is "Lane2:Lane 1\$Included", To deselect it – "Lane2:Lane 1\$Excluded"

To select Lane 3, the parameterString is "Lane3:Lane 1\$Included", To deselect it – "Lane3:Lane 1\$Excluded"

To select Lane 4, the parameterString is "Lane4:Lane 1\$Included", To deselect it – "Lane4:Lane 1\$Excluded"

Example: `returnval=mClient.SetGeneralParameter(clientID,"D-PHY","Transmitter", "1.1.3 Data Lane Rise Time", "Lane1:Lane 1$Included")`

## Assign input signals to scope channels

Command name	Parameters	Description	Return Value	Example
SetGeneralParameter()	string clientID string device string suite string test string parameterString	This method sets the parameters that are not specific to any given test. Note. Using this command you 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.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string Assign input signals to scope channels

### out string clientID

Name	Type	Direction	Description
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

### string device

Name	Type	Direction	Description
device	string	IN	Specifies the name of the device.

### string suite

Name	Type	Direction	Description
suite	string	IN	Specifies the name of the suite.

### string test

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

**string parameterString**

Name	Type	Direction	Description
parameterString	string	IN	Specifies the acquire type, acquire parameter name, and value.

**NOTE.** To assign the Dp signal to Channel 1, the parameterString is "Lane5 Connected to:Data:Dn\$CH1"

To assign the Dp signal to Channel 1, the parameterString is "Lane6 Connected to:Data:Dn\$CH1"

To assign the Clkpp signal to Channel 1, the parameterString is "Lane7 Connected to:Clock:Clkp\$CH1"

To assign the Clknn signal to Channel 1, the parameterString is "Lane8 Connected to:Clock:Clkn\$CH1"

To assign the Clkp-Clkn signal to channel 1 if a differential probe is used for the clock signals, the parameterString is "Lane9 Connected to:Clkp-clkn\$CH1"

You can assign the signal to CH1, CH2, CH3, or CH4. These values must replace the text after the '\$' symbol in parameterString.

Example:`returnval=mClient.SetGeneralParameter(clientID,"D-PHY","Transmitter", "1.1.3 Data Lane Rise Time", "Lane5 Connected to:Data:Dn$CH1")`

**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.

```
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", "1.1.1 Data Lane LP-TX  
Thevenin Output High Level Voltage (VOH)", true)
```

7. Select a channel.

```
mClient.SetGeneralParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-  
TX Thevenin Output High Level Voltage (VOH)", "Lane1 Connected to:Lane1+(Dp):Single  
Ended$CH1")
```

```
mClient.SetGeneralParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-  
TX Thevenin Output High Level Voltage (VOH)", "Lane1 Connected to:Lane1-(Dn):Single  
Ended$CH2")
```

8. Configure the selected measurement.

```
mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-  
TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape  
Mode with Data As Trigger$Trigger Type$Transition")
```

```
mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-  
TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape  
Mode with Data As Trigger$Trigger Source$Ch1")
```

```
mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-  
TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape  
Mode with Data As Trigger$Trigger Upper Level(V)$1")
```

```
mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-  
TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape  
Mode with Data As Trigger$Trigger Lower Level(V)$0")
```

```
mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-  
TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape  
Mode with Data As Trigger$Trigger Time(ps)$500")
```

```
mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-  
TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape  
Mode with Data As Trigger$Trigger Transition$Greater Than")
```

```
mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-  
TX Thevenin Output High Level Voltage (VOH)", "Trigger Slope$Positive")
```

```
mClient.SetAcquireParameter(clientID, "D-PHY", "Transmitter", "1.1.1 Data Lane LP-  
TX Thevenin Output High Level Voltage (VOH)", "Data Lanes Acquisition In Escape  
Mode with Data As Trigger$Trigger If Violation$Occurs")
```

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, waitingMsbBxCaption, waitingMsbBxMessage,  
waitingMsbBxResponse)
```

```
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.

### Python PI Client example

The following is an example for Python PI Client available in the path, <C:\Program Files\Tektronix\TekExpress  
\TekExpress D-PHY\Examples\Programatic Interface

# SCPI commands

## About SCPI command

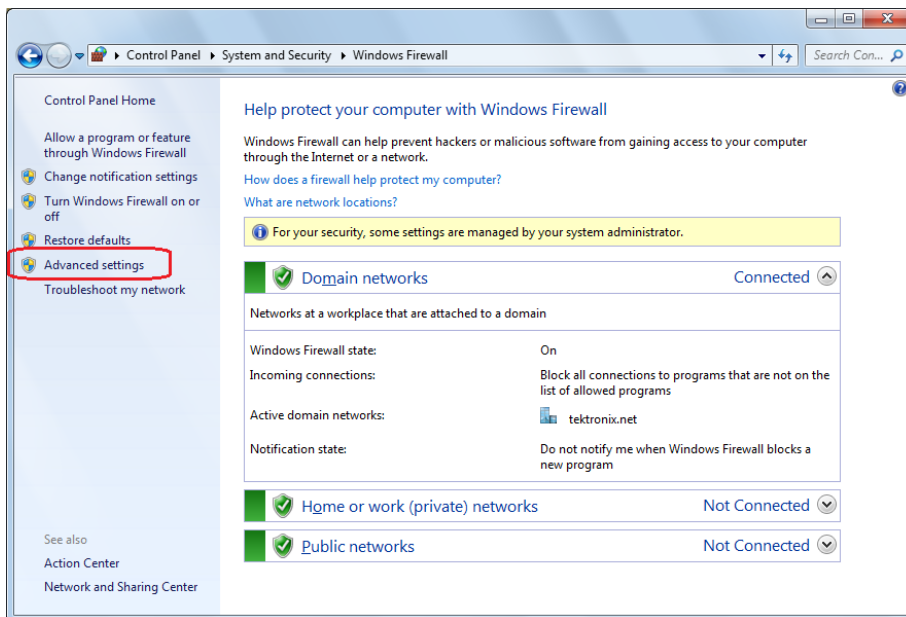
You can use Standard Commands for Programmable Instruments (SCPI) to communicate with the TekExpress application.

## Socket configuration for SCPI commands

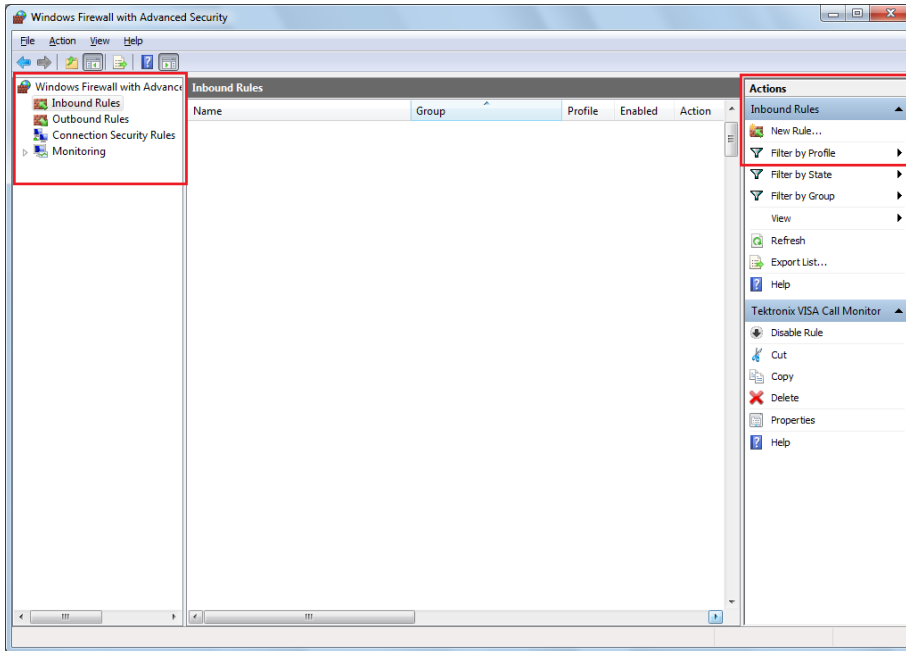
This section describes the steps for TCP/IP socket configuration and TekVISA configuration to execute the SCPI commands.

### TCP/IP socket configuration

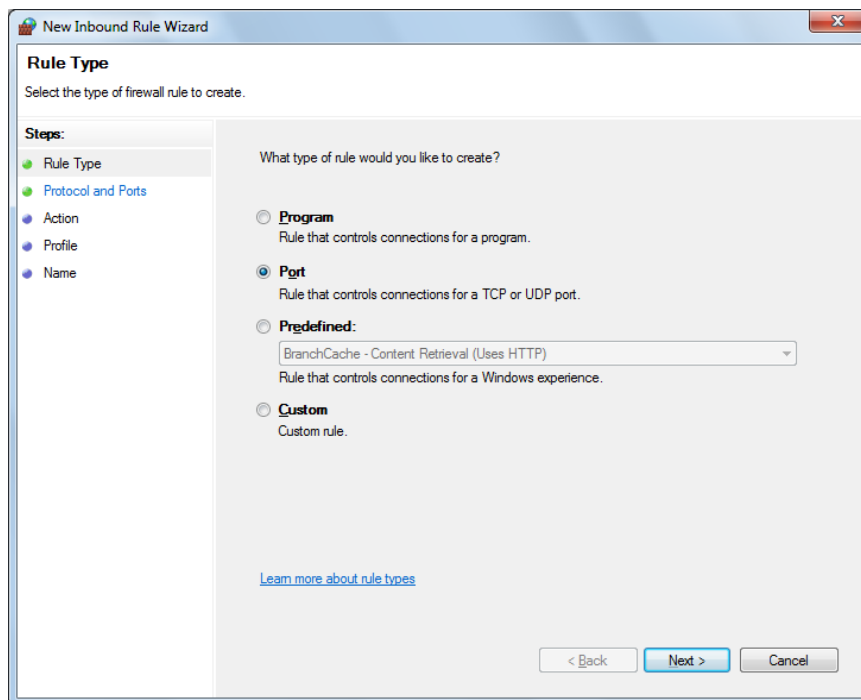
1. Click **Start > Control Panel > System and Security > Windows Firewall > Advanced settings**.



2. In Windows Firewall with Advanced Security menu, select **Windows Firewall with Advanced Security on Local Computer > Inbound Rules** and click **New Rule...**



3. In New Inbound Rule Wizard menu
  - a. Select **Port** and click **Next**.



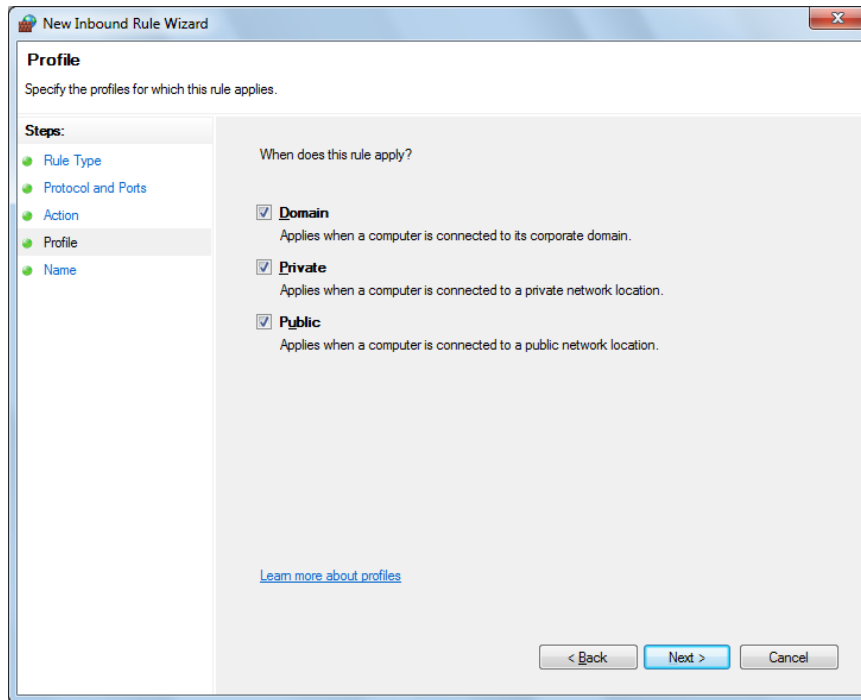
- b. Select **TCP** as rule apply and enter 5000 for **Specific local ports** and click **Next**.

The screenshot shows the 'New Inbound Rule Wizard' dialog box, specifically the 'Protocol and Ports' step. The title bar reads 'New Inbound Rule Wizard'. The main heading is 'Protocol and Ports' with the instruction 'Specify the protocols and ports to which this rule applies.' On the left, a 'Steps:' pane lists 'Rule Type', 'Protocol and Ports', 'Action', 'Profile', and 'Name', with 'Protocol and Ports' selected. The main area contains two questions: 'Does this rule apply to TCP or UDP?' with radio buttons for 'TCP' (selected) and 'UDP'; and 'Does this rule apply to all local ports or specific local ports?' with radio buttons for 'All local ports' and 'Specific local ports:' (selected). The 'Specific local ports:' field contains the text '5000' and has an example below it: 'Example: 80, 443, 5000-5010'. At the bottom, there are buttons for '< Back', 'Next >', and 'Cancel'. A link 'Learn more about protocol and ports' is also present.

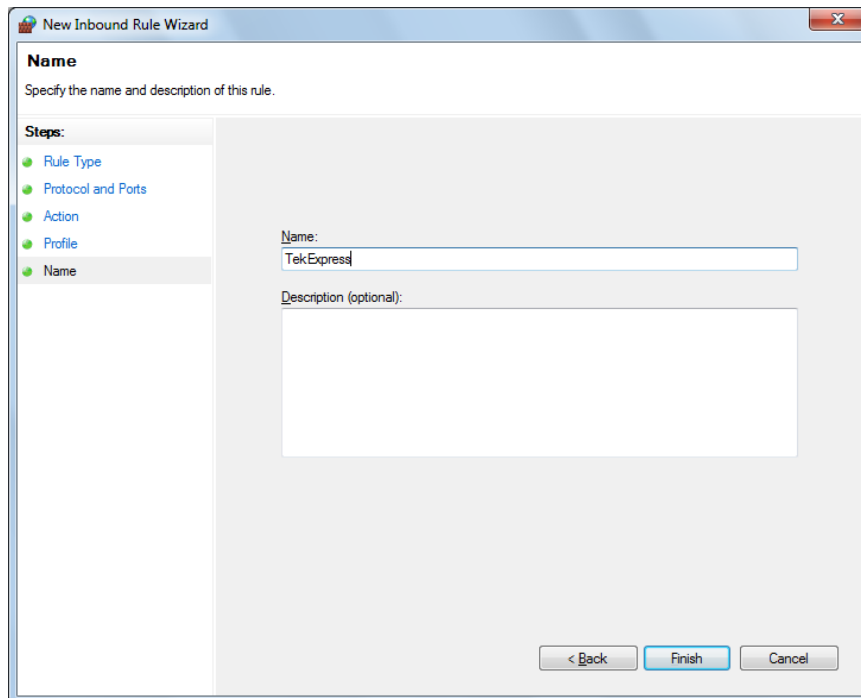
- c. Select **Allow the connection** and click **Next**.

The screenshot shows the 'New Inbound Rule Wizard' dialog box, specifically the 'Action' step. The title bar reads 'New Inbound Rule Wizard'. The main heading is 'Action' with the instruction 'Specify the action to be taken when a connection matches the conditions specified in the rule.' On the left, a 'Steps:' pane lists 'Rule Type', 'Protocol and Ports', 'Action', 'Profile', and 'Name', with 'Action' selected. The main area contains the question 'What action should be taken when a connection matches the specified conditions?' with three radio button options: 'Allow the connection' (selected), 'Allow the connection if it is secure', and 'Block the connection'. The 'Allow the connection if it is secure' option has a description: 'This includes only connections that have been authenticated by using IPsec. Connections will be secured using the settings in IPsec properties and rules in the Connection Security Rule node.' and a 'Customize...' button. At the bottom, there are buttons for '< Back', 'Next >', and 'Cancel'. A link 'Learn more about actions' is also present.

- d. Select **Domain**, **Private**, **Public** and click **Next**.

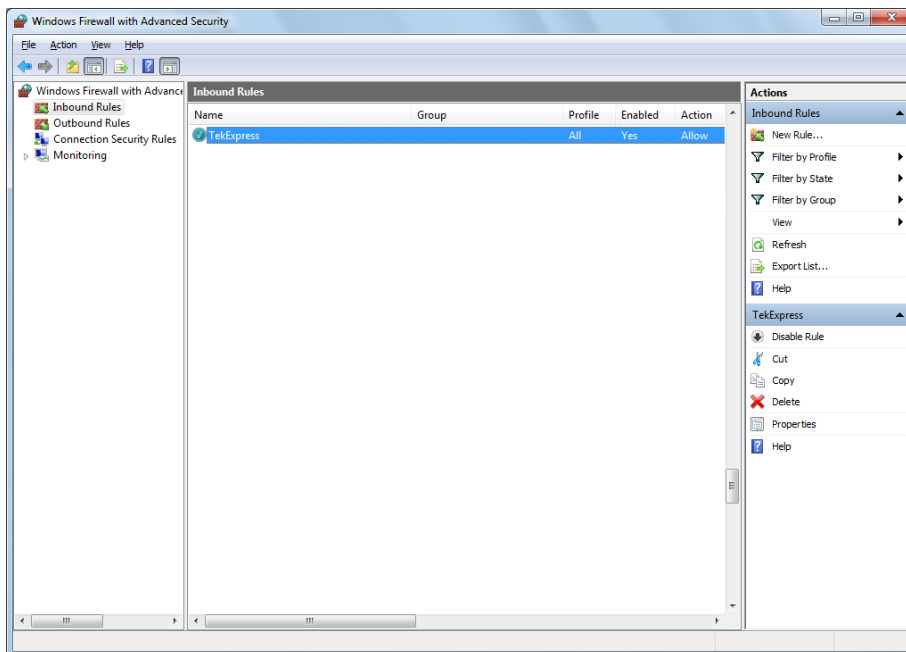


- e. Enter **Name**, Description (optional), and click **Finish**.



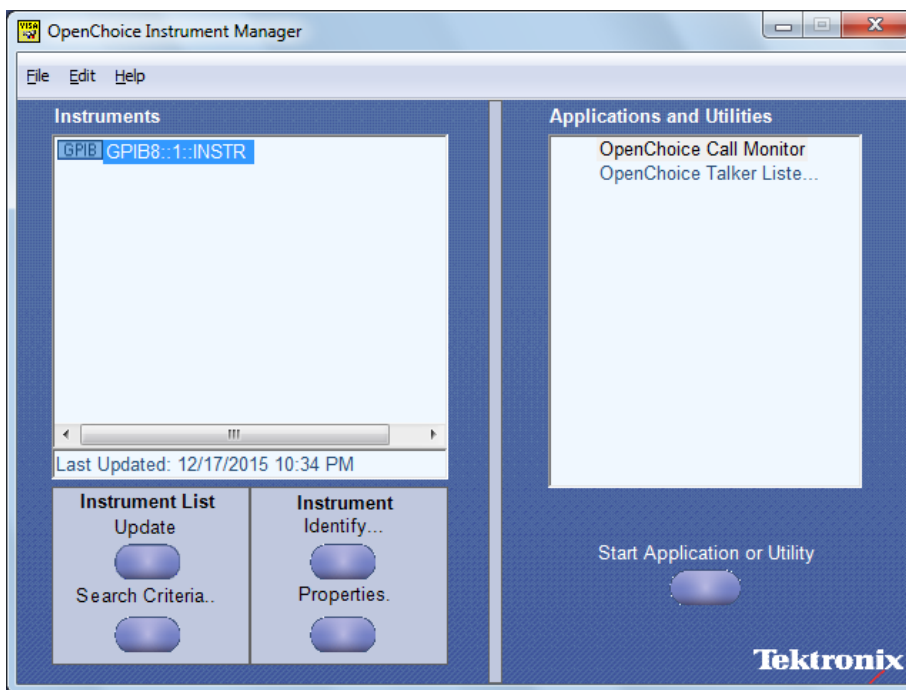



4. Check whether the Rule name is displayed in **Windows Firewall with Advanced Security** menu > **Inbound Rules**.



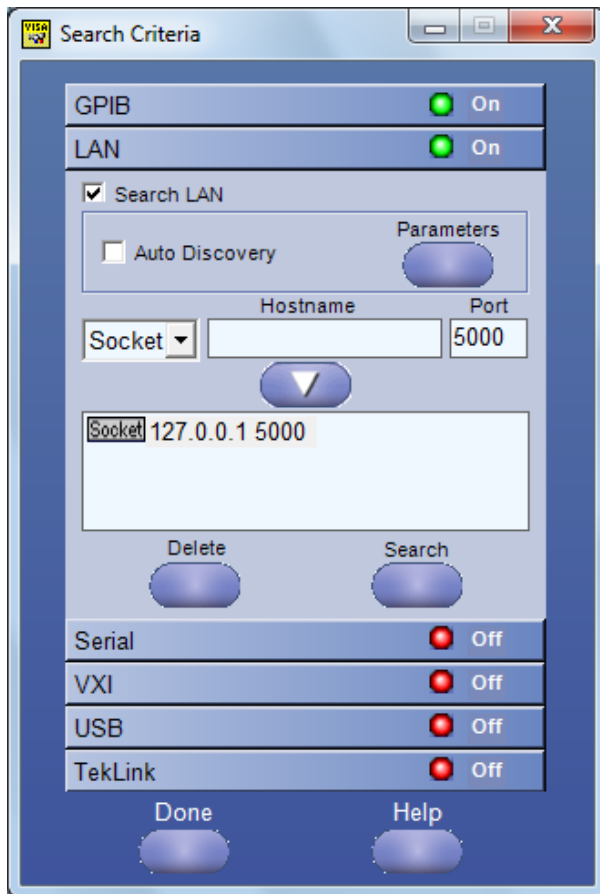
### TekVISA configuration

1. Click **Start > All Programs > TekVISA > OpenChoice Instrument Manager**.

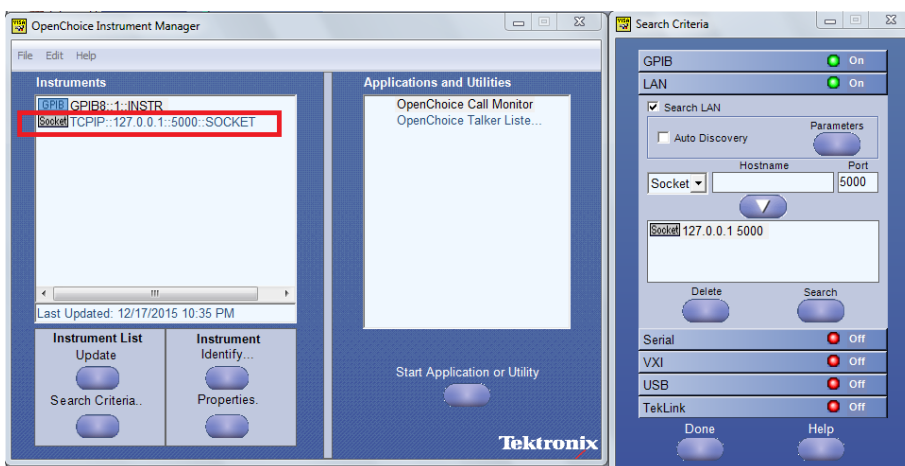


2. Click **Search Criteria**. In Search Criteria menu, click **LAN** to Turn-on. Select **Socket** from the drop-down list, enter the IP address of the TekExpress device in **Hostname** and type **Port** as 5000. Click  to configure the IP address with Port.

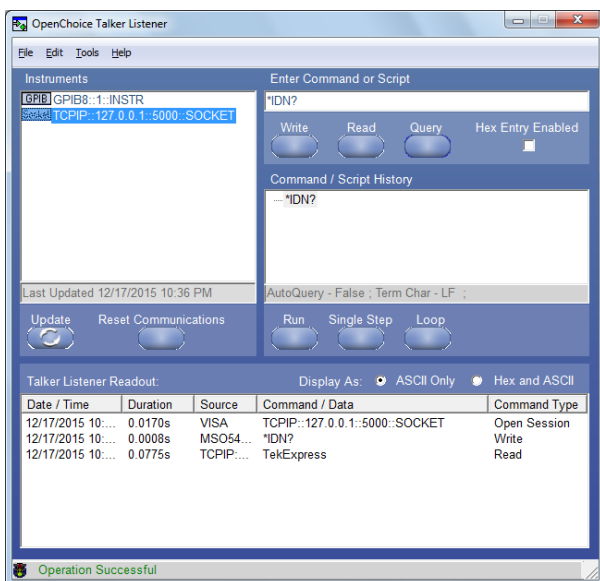
Enter the Hostname as 127.0.0.1 if the TekVISA and TekExpress application are in the same system, else enter the IP address of the TekExpress application system.



- Click **Search** to setup the TCPIP connection with the host. Check whether the TCPIP host name is displayed in **OpenChoice Instrument Manager > Instruments**.



- Double-click **OpenChoice Talker Listener** and enter the Command `*IDN?` in command entry field and click **Query**. Check that the Operation is successful and Talker Listener Readout displays the Command / Data.



## TEKEXP:\*IDN?

This command queries the active TekExpress application name running on the oscilloscope.

### Syntax

```
TEKEXP:*IDN?\n
```

### Inputs

NA

### Outputs

Returns active TekExpress application name running on the oscilloscope.



---

**TIP.** [Click here](#) for examples.

---

## TEKEXP:\*OPC?

This command queries the execution status of the last executed command.

### Syntax

```
TEKEXP:*OPC?\n
```

### Inputs

NA

### Outputs

0 - last command execution is not complete

1 - last command execution is complete



---

**TIP.** [Click here](#) for examples.

---

## TEKEXP:ACQUIRE\_MODE

This command sets the acquire mode as live or pre-recorded.

### Syntax

```
TEKEXP:ACQUIRE_MODE {LIVE | PRE-RECORDED}\n
```

### Inputs

```
{LIVE | PRE-RECORDED}
```

### Outputs

NA



---

**TIP.** [Click here](#) for examples.

---

## TEKEXP:ACQUIRE\_MODE?

This command queries the acquire mode type.

### Syntax

```
TEKEXP:ACQUIRE_MODE?\n
```

### Inputs

NA

### Outputs

```
{LIVE | PRE-RECORDED}
```



---

**TIP.** [Click here](#) for examples.

---

## TEKEXP:EXPORT

This command returns all the bytes of data to the specified file.

Syntax	Outputs
TEKEXP:EXPORT REPORT\n	Returns the report file in bytes
TEKEXP:EXPORT WFM, "<FileName>"\n	Returns the specified waveform file in bytes
TEKEXP:EXPORT IMAGE, "<FileName>"\n	Returns the specified image file in bytes

### Inputs

FileName - Specifies the file name



**TIP.** [Click here](#) for examples.

## TEKEXP:INFO?

This command queries the information about the file(s).

Syntax	Outputs
TEKEXP:INFO? REPORT\n	<ReportFileSize>,"<ReportFileName.mht>"
TEKEXP:INFO? WFM\n	<WfmFile1Size>,"<WfmFileName1.wfm>";<WfmFile2Size>,"<WfmFileName2.wfm>";...
TEKEXP:INFO? IMAGE\n	<Image1FileSize>,"<Image1FileName>";<Image2FileSize>,"<Image2FileName>" ;...



**TIP.** [Click here](#) for examples.

## TEKEXP:INSTRUMENT

This command sets the value for the selected instrument type.

### Syntax

TEKEXP:INSTRUMENT "<InstrumentType>",<Value>"\n

## Inputs

InstrumentType

value



---

**TIP.** Check [Command parameters list](#) for InstrumentType and Value parameters.

---

## Outputs

NA



---

**TIP.** [Click here](#) for examples.

---

## TEKEXP:INSTRUMENT?

This command queries the instrument selected for the specified instrument type.

### Syntax

```
TEKEXP:INSTRUMENT? "<InstrumentType>"\n
```

### Inputs

InstrumentType



---

**TIP.** Check [Command parameters list](#) for InstrumentType parameters.

---

### Outputs

Returns the instrument selected for the specified instrument type



---

**TIP.** [Click here](#) for examples.

---

## TEKEXP:LASTERROR?

This command queries the last error string occurred for the current TCP session. If there are no errors since startup, or since the last call to TEKEXP:LASTERROR?\n, this command returns an empty string.

### Syntax

TEKEXP:LASTERROR?\n

### Inputs

NA

### Outputs

<string>



**TIP.** [Click here](#) for examples.

## TEKEXP:LIST?

This command queries the list of available device, suite, test, version or instrument.

Syntax	Outputs
TEKEXP:LIST? DEVICE\n	Returns the list of available device(s) as comma separated values.
TEKEXP:LIST? SUITE\n	Returns the list of available suite(s) as comma separated values.
TEKEXP:LIST? TEST\n	Returns the list of available test(s) as comma separated values.
TEKEXP:LIST? VERSION\n	Returns the list of available version(s) as comma separated values.
TEKEXP:LIST? INSTRUMENT, "<InstrumentType>"\n	Returns the list of available instruments' for the given Instrument type as comma separated values.

**NOTE.** This command returns the list of items within double quotes ("). Iterate the receive procedure until the list ends with double quotes otherwise the next query commands won't work as expected.



## Inputs

InstrumentType



---

**TIP.** Check [Command parameters list](#) for InstrumentType parameters.

---



---

**TIP.** [Click here](#) for examples.

---

## TEKEXP:MODE

This command sets the execution mode as compliance or user defined.

### Syntax

```
TEKEXP:MODE {COMPLIANCE | USER-DEFINED}\n
```

### Inputs

{COMPLIANCE | USER-DEFINED}

### Outputs

NA



---

**TIP.** [Click here](#) for examples.

---

## TEKEXP:MODE?

This command queries the execution mode type.

### Syntax

```
TEKEXP:MODE?\n
```

### Inputs

NA

### Outputs

{COMPLIANCE | USER-DEFINED}



---

**TIP.** [Click here](#) for examples.

---

## TEKEXP:POPUP

This command sets the response to the active popup shown in the application.

### Syntax

```
TEKEXP:POPUP "<PopupResponse>"\n
```

### Inputs

PopupResponse

### Outputs

NA



---

**TIP.** [Click here](#) for examples.

---

## TEKEXP:POPUP?

This command queries the active popup information shown in the application.

### Syntax

```
TEKEXP:POPUP?\n
```

### Inputs

NA

### Outputs

Returns the active popup information in the application.



---

**TIP.** [Click here](#) for examples.

---

## TEKEXP:REPORT

This command generates the report for the current session.

### Syntax

```
TEKEXP:REPORT GENERATE\n
```

### Inputs

GENERATE

### Outputs

NA



---

**TIP.** [Click here](#) for examples.

---

## TEKEXP:REPORT?

This command queries the queried header field value in the report.

### Syntax

```
TEKEXP:REPORT? "<HeaderField>"\n
```

### Inputs

HeaderField - Specifies to return the measured value for the indicated test.



---

**TIP.** Check **Report** for HeaderField parameters.

---

### Outputs

Returns the queried header field value in the report



---

**TIP.** [Click here](#) for examples.

---

## TEKEXP:RESULT?

This command queries the result available in report summary/details table.

Syntax	Outputs
TEKEXP:RESULT? "<TestName>"\n	Return Pass/Fail status of the test.
TEKEXP:RESULT? "<TestName>", "<ColumnName>"\n	Returns all the row values of the specified column for the test.
TEKEXP:RESULT? "<TestName>", "<ColumnName>", <RowNumber>\n	Returns the column value for the specified row number <sup>1</sup>

### Inputs

TestName - Specifies the name of the test for which to obtain the test result value.

ColumnName - Specifies the column name for the measurement

RowNumber - Specifies the row number of the measurement



**TIP.** Check **Results** panel for TestName, ColumnName, and RowNumber parameters.



**TIP.** [Click here](#) for examples.

## TEKEXP:SELECT

This command selects the device, suite, version, or test.

### Syntax

```
TEKEXP:SELECT <string1>,<string2>,<string4>\n
```

```
TEKEXP:SELECT TEST,<string3>,<string4>\n
```

### Inputs

<string1> = {DEVICE | SUITE | VERSION}

<string2> = {DeviceName | SuiteName | VersionName}

<string3> = {"<TestName>" | ALL | REQUIRED }

<string4> = {TRUE | FALSE}



**TIP.** Check [Command parameters list](#) for DeviceName, SuiteName, VersionName, and TestName parameters.

<sup>1</sup> Row number starts from zero.




---

**TIP.** [Click here](#) for examples.

---

## Outputs

NA

## TEKEXP:SELECT?

This command queries the name of the selected device, suite, version, or test.

### Syntax

TEKEXP:SELECT? {DEVICE | SUITE | TEST | VERSION}\n

### Inputs

{DEVICE | SUITE | TEST | VERSION}

### Outputs

Returns the name of the selected device, suite, version, or test.




---

**TIP.** [Click here](#) for examples.

---

## TEKEXP:SETUP

This command sets the value of the current setup.

Syntax	Outputs
TEKEXP:SETUP DEFAULT\n	Restore to default Setup
TEKEXP:SETUP OPEN,"<SessionName>"\n	Open the session
TEKEXP:SETUP SAVE\n	Save the session
TEKEXP:SETUP SAVE,"<SessionName>"\n	Save the session

### Inputs

sessionName - The name of the session




---

**TIP.** [Click here](#) for examples.

---

## TEKEXP:STATE

This command sets the execution state of the application.

### Syntax

```
TEKEXP:STATE {RUN | STOP | PAUSE | RESUME}\n
```

### Inputs

```
{RUN | STOP | PAUSE | RESUME}
```

### Outputs

NA



**TIP.** [Click here](#) for examples.

## TEKEXP:STATE?

This command queries the current setup state.

Syntax	Outputs
TEKEXP:STATE?	RUNNING   PAUSED   WAIT   ERROR   READY
TEKEXP:STATE? SETUP	SAVED   NOT_SAVED



**TIP.** [Click here](#) for examples.

## TEKEXP:VALUE

This command sets the value of parameters of type General, Acquire, Analyze, or DUTID.

### Syntax

```
TEKEXP:VALUE GENERAL, "<ParameterName>", "<Value>"\n
```

```
TEKEXP:VALUE ACQUIRE, "<TestName>", "<AcquireType>", "<ParameterName>", "<Value>"\n
```

```
TEKEXP:VALUE ANALYZE, "<TestName>", "<ParameterName>". "<Value>"\n
```

```
TEKEXP:VALUE DUTID, "<Value>"\n
```

```
TEKEXP:VALUE WFMFILE, <Test_Name>, <Acquire_Type>, <FileName1$FileName2>\n
```

## Inputs

ParameterName - Specifies the parameter name

TestName - Specifies the test name

AcquireType - Specifies the acquire type

Value - Specifies the value to set

FileName1\$FileName2 - Specifies the waveform file name



**TIP.** Check [Command parameters list](#) for ParameterName, AcquireType, and Value parameters.

## Outputs

NA



**TIP.** [Click here](#) for examples.

## TEKEXP:VALUE?

This command queries the value of the parameter for type General, Acquire, Analyze, or DUTID.

Syntax	Outputs
TEKEXP:VALUE? GENERAL,"<ParameterName>"\n	Returns the value of Parameter for type GENERAL
TEKEXP:VALUE? ACQUIRE,"<TestName>", "<AcquireType>","<ParameterName>"\n	Returns the value of Parameter for type ACQUIRE
TEKEXP:VALUE? ANALYZE, "<TestName>","<ParameterName>"\n	Returns the value of Parameter for type ANALYZE
TEKEXP:VALUE? DUTID\n	Returns the DUTID value
TEKEXP:VALUE? WFMMFILE,<Test_Name>,<Acquire_Type>\n	Returns the waveform file name

## Inputs

ParameterName - Specifies the parameter name

TestName - Specifies the test name

AcquireType - Specifies the acquire type



**TIP.** Check [Command parameters list](#) for ParameterName and AcquireType parameters.

## Outputs

Returns the value of Parameter for type GENERAL | ACQUIRE | ANALYZE | DUTID.



**TIP.** [Click here](#) for examples.

## Command parameters list

This section provides the parameters list for the SCPI commands.

### TekExpress D-PHY command parameters

#### ParameterName and Value for DUT, Test selection, Acquisition, Configuration and Preferences tabs

Specifies the ParameterName and Value for DUT, Test selection, Acquisition, Configuration, and Preferences tabs

**Table 16: ParameterName and Value for DUT tab**

Parameters	Description
DUT ID	Specifies the value parameters. For DUTID, valid value is: Comment
Acquiremode	Specifies the acquire mode parameter: Acquire live waveforms
Use pre-recorded mode	Specifies the pre-recorded parameter: Use pre-recorded waveform files
View	Specifies the view mode Compliance
Test Mode	Specifies the test mode Valid values are: <ul style="list-style-type: none"> <li>■ Normal</li> <li>■ Escape</li> <li>■ Both</li> </ul>
Clock Mode	Specifies the clock mode Valid values are: <ul style="list-style-type: none"> <li>■ Normal</li> <li>■ Continuous</li> </ul>
Use Cursors	Specifies the cursor values Valid values are: <ul style="list-style-type: none"> <li>■ Manual</li> <li>■ Automatic</li> </ul>
HS Data Rate	<ul style="list-style-type: none"> <li>■ &lt;=1</li> <li>■ &gt;1 and &lt;=1.5</li> <li>■ &gt;1.5</li> </ul>



Parameters	Description
Multi-Lane	Configures the lanes and channels
Clock-Lane Probing	Specifies the selected source. Valid values are <ul style="list-style-type: none"><li data-bbox="678 394 860 426">■ Single-ended</li><li data-bbox="678 443 834 474">■ Differential</li></ul>

**Table 17: ParameterName and Value for Test Selection tab**

Parameters	Description
Test Name	<p>Specifies the test measurement name. Valid values are:</p> <ul style="list-style-type: none"> <li>■ 1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)</li> <li>■ 1.1.2 Data Lane LP-TX Thevenin Output Low Level Voltage (VOL)</li> <li>■ 1.1.3 Data Lane LP-TX 15%-85% Rise Time (TRLP)</li> <li>■ 1.1.4 Data Lane LP-TX 15%-85% Fall Time (TFLP)</li> <li>■ 1.1.5 Data Lane LP-TX Slew Rate vs. CLOAD</li> <li>■ 1.1.6 Data Lane LP-TX Pulse Width of Exclusive-OR Clock (TLP-PULSE-TX)</li> <li>■ 1.1.7 Data Lane LP-TX Period of Exclusive-OR Clock (TLP-PER-TX)</li> <li>■ 1.2.1 Clock Lane LP-TX Thevenin Output High Level Voltage (VOH)</li> <li>■ 1.2.2 Clock Lane LP-TX Thevenin Output Low Level Voltage (VOL)</li> <li>■ 1.2.3 Clock Lane LP-TX 15%-85% Rise Time (TRLP)</li> <li>■ 1.2.4 Clock Lane LP-TX 15%-85% Fall Time (TFLP)</li> <li>■ 1.2.5 Clock Lane LP-TX Slew Rate vs. CLOAD</li> <li>■ 1.3.1 Data Lane HS Entry: Data Lane TLPX Value</li> <li>■ 1.3.2 Data Lane HS Entry: THS-PREPARE Value</li> <li>■ 1.3.3 Data Lane HS Entry: THS-PREPARE + THS-ZERO Value</li> <li>■ 1.3.4 Data Lane HS-TX Differential Voltages (VOD(0), VOD(1))</li> <li>■ 1.3.5 Data Lane HS-TX Differential Voltage Mismatch (VOD)</li> <li>■ 1.3.6 Data Lane HS-TX Single-Ended Output High Voltages (VOHHS(DP), VOHHS(DN))</li> <li>■ 1.3.7 Data Lane HS-TX Static Common-Mode Voltages (VCMTX(1), VCMTX(0))</li> <li>■ 1.3.8 Data Lane HS-TX Static Common-Mode Voltage Mismatch (VCMTX(1,0))</li> <li>■ 1.3.9 Data Lane HS-TX Dynamic Common-Level Variations Between 50-450MHz (VCMTX(LF))</li> <li>■ 1.3.10 Data Lane HS-TX Dynamic Common-Level Variations Above 450MHz (VCMTX(HF))</li> <li>■ 1.3.11 Data Lane HS-TX 20%-80% Rise Time (tR)</li> <li>■ 1.3.12 Data Lane HS-TX 80%-20% Fall Time (tF)</li> <li>■ 1.3.13 Data Lane HS Exit: THS-TRAIL Value</li> <li>■ 1.3.14 Data Lane HS Exit: 30%-85% Post-EoT Rise Time (TREET)</li> <li>■ 1.3.15 Data Lane HS Exit: TEOT Value</li> <li>■ 1.3.16 Data Lane HS Exit: THS-EXIT Value</li> <li>■ 1.4.1 Clock Lane HS Entry: TLPX Value</li> <li>■ 1.4.2 Clock Lane HS Entry: TCLK-PREPARE Value</li> <li>■ 1.4.3 Clock Lane HS Entry: TCLK-PREPARE+TCLK-ZERO Value</li> </ul>

Parameters	Description
	<ul style="list-style-type: none"> <li>■ 1.4.4 Clock Lane HS-TX Differential Voltages (VOD(0), VOD(1))</li> <li>■ 1.4.5 Clock Lane HS-TX Differential Voltage Mismatch (VOD)</li> <li>■ 1.4.6 Clock Lane HS-TX Single-Ended Output High Voltages (VOHHS(DP), VOHHS(DN))</li> <li>■ 1.4.7 Clock Lane HS-TX Static Common-Mode Voltages (VCMTX(1), VCMTX(0))</li> <li>■ 1.4.8 Clock Lane HS-TX Static Common-Mode Voltage Mismatch (VCMTX(1,0))</li> <li>■ 1.4.9 Clock Lane HS-TX Dynamic Common-Level Variations Between 50-450MHz (VCMTX(LF))</li> <li>■ 1.4.10 Clock Lane HS-TX Dynamic Common-Level Variations Above 450MHz (VCMTX(HF))</li> <li>■ 1.4.11 Clock Lane HS-TX 20%-80% Rise Time (tR)</li> <li>■ 1.4.12 Clock Lane HS-TX 80%-20% Fall Time (tF)</li> <li>■ 1.4.13 Clock Lane HS Exit: TCLK-TRAIL Value</li> <li>■ 1.4.14 Clock Lane HS Exit: 30%-85% Post-EoT Rise Time (TREOT)</li> <li>■ 1.4.15 Clock Lane HS Exit: TEOT Value</li> <li>■ 1.4.16 Clock Lane HS Exit: THS-EXIT Value</li> <li>■ 1.4.17 Clock Lane HS Clock Instantaneous (UIINST)</li> <li>■ 1.4.18 Clock Lane HS Clock Delta UI</li> <li>■ 1.5.1 HS Entry TCLK PREValue</li> <li>■ 1.5.2 HS Exit TCLK POSTValue</li> <li>■ 1.5.3 HS Clock Rising Edge Alignment to First Payload Bit</li> <li>■ 1.5.4 Data to Clock Skew (TSKEW(TX))</li> <li>■ 1.5.5 Initial HS Skew Calibration Burst (TSKEWCAL-SYNC, TSKEWCAL)</li> <li>■ 1.5.6 Periodic HS Skew Calibration Burst (TSKEWCAL-SYNC, TSKEWCAL)</li> <li>■ 1.6.1 INIT LP-TX Initialization Period (TINIT, MASTER)</li> <li>■ 1.6.2 ULPS Entry Verification of Clock Lane LP-TX ULPS support</li> <li>■ 1.6.3 ULPS Exit Transmitted TWAKEUP Interval</li> <li>■ 1.6.4 BTA TX-Side TTA-GO Interval Value</li> <li>■ 1.6.5 BTA RX-Side TTA-SURE Interval Value</li> <li>■ 1.6.6 BTA RX-Side TTA-GET Interval Value</li> </ul>

**Table 18: ParameterName and Value for Acquisition tab**

Parameters	Description
Acquisition and Save Options	Saves the acquisitions and waveforms <ul style="list-style-type: none"> <li>■ Save All Waveforms Before Analysis</li> <li>■ Analyze immediately, No Waveforms Saved</li> <li>■ Save and Analyze Acquisition in Sequence</li> </ul>
Show Acquire Parameters	<ul style="list-style-type: none"> <li>■ True</li> <li>■ False</li> </ul>

**Table 19: ParameterName and Value for Preferences tab**

Parameters	Description
Number of Runs	1 to 250
On Test Failure, stop and notify me of the failure	TRUE or FALSE
Email Settings	Recipient e-mail Address Sender's Address

**Table 20: ParameterName and Value for Analyze**

TestName	ParameterName	Value
1.4.17 Clock Lane HS Clock Instantaneous (UIINST)	UI INST Min (ns)	0.01 to 12.5
1.5.3 HS Clock Rising Edge Alignment to First Payload Bit	Analysis Mode	<ul style="list-style-type: none"> <li>■ Automatic</li> <li>■ Visual Verification</li> </ul>

**Table 21: ParameterName and Value for General**

Parameters	Description
Measurement Method	Automatic Manual
HS Data Rate	<= 1 >1 and <=1.5 >1.5
Lane1:Lane 1	Included Excluded
Lane2:Lane 2	Included Excluded
Lane3:Lane 3	Included Excluded
Lane4:Lane 4	Included Excluded

Parameters	Description
Clock Lane Probing	Single-ended Differential
Clock Mode	Normal Continuous
Test Mode	Normal Escape
Lane5:Data:Dp	Included Excluded
Lane6:Data:Dn	Included Excluded
Lane7:Clock:Clkp	Included Excluded
Lane8:Clock:Clkn	Included Excluded
Lane9:Clock:Clkp-Clkn	Included Excluded
Lane5 Connected to:Data:Dp	CH1 CH2 CH3 CH4
Lane6 Connected to:Data:Dn	CH1 CH2 CH3 CH4
Lane7 Connected to:Clock:Clkp	CH1 CH2 CH3 CH4
Lane8 Connected to:Clock:Clkn	CH1 CH2 CH3 CH4
Lane9 Connected to:Clock:Clkp-Clkn	CH1 CH2 CH3 CH4
Save Report As Type	Web Archive (*.mht;*.mhtml) PDF (*.pdf;) CSV (*.csv;)
Disable Single Lane Pop-up	TRUE FALSE

**Table 22: ParameterName and Value for Acquire**

Test Name	Acquire Type	Parameter Name	Values
1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH) 1.1.2 Data Lane LP-TX Thevenin Output Low Level Voltage (VOL) 1.1.3 Data Lane LP-TX 15%-85% Rise Time (TRLP) 1.1.4 Data Lane LP-TX 15%-85% Fall Time (TFLP) 1.1.5 Data Lane LP-TX Slew Rate vs. CLOAD 1.1.6 Data Lane LP-TX Pulse Width of Exclusive-OR Clock (TLP-PULSE-TX) 1.1.7 Data Lane LP-TX Period of Exclusive-OR Clock (TLP-PER-TX)	Data Lanes Acquisition In Escape Mode with Data As Trigger	Vertical Scale Data (mV)	<ul style="list-style-type: none"> <li>■ 10</li> <li>■ 500</li> </ul>
		Vertical Scale Clock (mV)	<ul style="list-style-type: none"> <li>■ 10</li> <li>■ 500</li> </ul>
		Vertical Offset (V)	0
		Horizontal Scale (us)	<ul style="list-style-type: none"> <li>■ 8</li> <li>■ 10</li> <li>■ 12</li> </ul>
		Sample Rate (Gs/s)	<ul style="list-style-type: none"> <li>■ 12.5</li> <li>■ 25</li> </ul>
		Record Length	<ul style="list-style-type: none"> <li>■ 100000</li> <li>■ 10000000</li> </ul>
		Trigger Type: Transition	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Transition: Greater Than, Less Than, Occurs</li> <li>■ Delta Time: 500, 250</li> <li>■ Qualify: Occurs, Logic</li> <li>■ Lower Level: 0.05</li> <li>■ Upper Level: 1</li> </ul>
		Trigger Type: Edge	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Level: 0.1 V to 1 V</li> </ul>
		Trigger Type: Width	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Polarity: Positive, Negative</li> <li>■ Lower Limit: 25.0E-6</li> <li>■ Upper Limit: 300.0E-6</li> </ul>
		1.2.1 Clock Lane LP-TX Thevenin Output High Level Voltage (VOH)	Clock Lanes Acquisition In Escape Mode with Clock As Trigger
Vertical Scale Clock (mV)	<ul style="list-style-type: none"> <li>■ 500</li> </ul>		
Vertical Offset (V)	0		

Test Name	Acquire Type	Parameter Name	Values
1.2.2 Clock Lane LP-TX Thevenin Output Low Level Voltage (VOL)  1.2.3 Clock Lane LP-TX 15%-85% Rise Time (TRLP)  1.2.4 Clock Lane LP-TX 15%-85% Fall Time (TFLP)  1.2.5 Clock Lane LP-TX Slew Rate vs. CLOAD		Horizontal Scale (us)	<ul style="list-style-type: none"> <li>■ 8</li> <li>■ 10</li> <li>■ 12</li> </ul>
		Sample Rate (GS/s)	<ul style="list-style-type: none"> <li>■ 12.5</li> <li>■ 25</li> </ul>
		Record Length	<ul style="list-style-type: none"> <li>■ 100000</li> <li>■ 1000000</li> </ul>
		Trigger Type: Transition	<ul style="list-style-type: none"> <li>■ Source: Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Transition: Greater Than, Less Than, Occurs</li> <li>■ Delta Time: 500, 250</li> <li>■ Qualify: Occurs, Logic</li> <li>■ Lower Level: 0.05</li> <li>■ Upper Level: 1</li> </ul>
		Trigger Type: Edge	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Level: 0.1 V to 1 V</li> </ul>
		Trigger Type: Width	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Polarity: Positive, Negative</li> <li>■ Lower Limit: 25.0E-6</li> <li>■ Upper Limit: 300.0E-6</li> </ul>
1.3.1 Data Lane HS Entry: Data Lane TLPX Value  1.3.2 Data Lane HS Entry: THS-PREPARE Value  1.3.3 Data Lane HS Entry: THS-PREPARE + THS-ZERO Value  1.3.4 Data Lane HS-TX Differential Voltages (VOD(0), VOD(1))	Data and Clock Lanes Acquisition with Data As Trigger	Vertical Scale Data (mV)	<ul style="list-style-type: none"> <li>■ 10</li> </ul>
		Vertical Scale Clock (mV)	<ul style="list-style-type: none"> <li>■ 500</li> </ul>
		Vertical Offset (V)	0
		Horizontal Scale (us)	<ul style="list-style-type: none"> <li>■ 8</li> <li>■ 10</li> <li>■ 12</li> </ul>
		Sample Rate (GS/s)	<ul style="list-style-type: none"> <li>■ 12.5</li> <li>■ 25</li> </ul>

Test Name	Acquire Type	Parameter Name	Values
1.3.5 Data Lane HS-TX Differential Voltage Mismatch (VOD)		Record Length	<ul style="list-style-type: none"> <li>■ 100000</li> <li>■ 10000000</li> </ul>
1.3.6 Data Lane HS-TX Single-Ended Output High Voltages (VOHHS(DP), VOHHS(DN))		Trigger Type: Transition	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Transition: Greater Than, Less Than, Occurs</li> <li>■ Delta Time: 500, 250</li> <li>■ Qualify: Occurs, Logic</li> <li>■ Lower Level: 0.05</li> <li>■ Upper Level: 1</li> </ul>
1.3.7 Data Lane HS-TX Static Common-Mode Voltages (VCMTX(1), VCMTX(0))		Trigger Type: Edge	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Level: 0.1 V to 1 V</li> </ul>
1.3.8 Data Lane HS-TX Static Common-Mode Voltage Mismatch (VCMTX(1,0))			
1.3.9 Data Lane HS-TX Dynamic Common-Level Variations Between 50-450MHz (VCMTX(LF))		Trigger Type: Width	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Polarity: Positive, Negative</li> <li>■ Lower Limit: 25.0E-6</li> <li>■ Upper Limit: 300.0E-6</li> </ul>
1.3.10 Data Lane HS-TX Dynamic Common-Level Variations Above 450MHz (VCMTX(HF))			
1.3.11 Data Lane HS-TX 20%-80% Rise Time (tR)			
1.3.12 Data Lane HS-TX 80%-20% Fall Time (tF)			
1.3.13 Data Lane HS Exit: THS-TRAIL Value			
1.3.14 Data Lane HS Exit: 30%-85% Post-EoT Rise Time (TREOT)			
1.3.15 Data Lane HS Exit: TEOT Value	Clock Lanes Acquisition	Vertical Scale Data (mV)	<ul style="list-style-type: none"> <li>■ 10</li> </ul>
1.3.16 Data Lane HS Exit: THS-EXIT Value		Vertical Scale Clock (mV)	<ul style="list-style-type: none"> <li>■ 500</li> </ul>
1.4.1 Clock Lane HS Entry: TLPX Value			



Test Name	Acquire Type	Parameter Name	Values	
1.4.2 Clock Lane HS Entry: TCLK-PREPARE Value	with Clock As Trigger	Vertical Offset (V)	0	
1.4.3 Clock Lane HS Entry: TCLK-PREPARE+TCLK- ZERO Value		Horizontal Scale (us)	<ul style="list-style-type: none"> <li>■ 8</li> <li>■ 10</li> <li>■ 12</li> </ul>	
1.4.4 Clock Lane HS-TX Differential Voltages (VOD(0), VOD(1))		Sample Rate (GS/s)	<ul style="list-style-type: none"> <li>■ 12.5</li> <li>■ 25</li> </ul>	
1.4.5 Clock Lane HS-TX Differential Voltage Mismatch (VOD)		Record Length	<ul style="list-style-type: none"> <li>■ 100000</li> <li>■ 10000000</li> </ul>	
1.4.6 Clock Lane HS-TX Single-Ended Output High Voltages (VOHHS(DP), VOHHS(DN))		Trigger Type: Transition	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Transition: Greater Than, Less Than, Occurs</li> <li>■ Delta Time: 500, 250</li> <li>■ Qualify: Occurs, Logic</li> <li>■ Lower Level: 0.05</li> <li>■ Upper Level: 1</li> </ul>	
1.4.7 Clock Lane HS-TX Static Common-Mode Voltages (VCMTX(1), VCMTX(0))		Transition: Edge		<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Level: 0.1 V to 1 V</li> </ul>
1.4.8 Clock Lane HS-TX Static Common-Mode Voltage Mismatch (VCMTX(1,0))				
1.4.9 Clock Lane HS-TX Dynamic Common-Level Variations Between 50-450MHz (VCMTX(LF))				
1.4.10 Clock Lane HS-TX Dynamic Common-Level				

Test Name	Acquire Type	Parameter Name	Values
Variations Above 450MHz (VCMTX(HF)) 1.4.11 Clock Lane HS-TX 20%-80% Rise Time (tR)  1.4.12 Clock Lane HS-TX 80%-20% Fall Time (tF)  1.4.13 Clock Lane HS Exit: TCLK-TRAIL Value  1.4.14 Clock Lane HS Exit: 30%-85% Post-EoT Rise Time (TREOT)  1.4.15 Clock Lane HS Exit: TEOT Value  1.4.16 Clock Lane HS Exit: THS-EXIT Value  1.4.18 Clock Lane HS Clock Delta UI  1.4.17 Clock Lane HS Clock Instantaneous (UIINST)		Transition: Width	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Polarity: Positive, Negative</li> <li>■ Lower Limit: 25.0E-6</li> <li>■ Upper Limit: 300.0E-6</li> </ul>
1.5.1 HS Entry TCLK PREValue  1.5.2 HS Exit TCLK POSTValue  1.5.4 Data to Clock Skew (TSKEW(TX))	Data and Clock Lanes Acquisition with Data As Trigger	Vertical Scale Data (mV) Vertical Scale Clock (mV)  Vertical Offset (V)  Horizontal Scale (us)  Sample Rate (GS/s)  Record Length	<ul style="list-style-type: none"> <li>■ 10</li> <li>■ 500</li> <li>0</li> <li>■ 8</li> <li>■ 10</li> <li>■ 12</li> <li>■ 12.5</li> <li>■ 25</li> <li>■ 100000</li> <li>■ 10000000</li> </ul>

Test Name	Acquire Type	Parameter Name	Values
		Trigger Type: Transition	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Transition: Greater Than, Less Than, Occurs</li> <li>■ Delta Time: 500, 250</li> <li>■ Qualify: Occurs, Logic</li> <li>■ Lower Level: 0.05</li> <li>■ Upper Level: 1</li> </ul>
		Trigger Type: Edge	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Level: 0.1 V to 1 V</li> </ul>
		Trigger Type: Width	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Polarity: Positive, Negative</li> <li>■ Lower Limit: 25.0E-6</li> <li>■ Upper Limit: 300.0E-6</li> </ul>
1.5.3 HS Clock Rising Edge Alignment to First Payload Bit	Data and Clock Lanes Acquisition with Data As Trigger	Analysis Mode	<ul style="list-style-type: none"> <li>■ Automatic</li> <li>■ Visual Verification</li> </ul>
		Vertical Scale Data (mV)	<ul style="list-style-type: none"> <li>■ 10</li> </ul>
		Vertical Scale Clock (mV)	<ul style="list-style-type: none"> <li>■ 500</li> </ul>
		Vertical Offset (V)	0
		Horizontal Scale (us)	<ul style="list-style-type: none"> <li>■ 8</li> <li>■ 10</li> <li>■ 12</li> </ul>
		Sample Rate (GS/s)	<ul style="list-style-type: none"> <li>■ 12.5</li> <li>■ 25</li> </ul>
		Record Length	<ul style="list-style-type: none"> <li>■ 100000</li> <li>■ 10000000</li> </ul>

Test Name	Acquire Type	Parameter Name	Values
		Trigger Type: Transition	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Transition: Greater Than, Less Than, Occurs</li> <li>■ Delta Time: 500, 250</li> <li>■ Qualify: Occurs, Logic</li> <li>■ Lower Level: 0.05</li> <li>■ Upper Level: 1</li> </ul>
		Trigger Type: Edge	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Level: 0.1 V to 1 V</li> </ul>
		Trigger Type: Width	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Polarity: Positive, Negative</li> <li>■ Lower Limit: 25.0E-6</li> <li>■ Upper Limit: 300.0E-6</li> </ul>
1.5.5 Initial HS Skew Calibration Burst (TSKEWCAL-SYNC, TSKEWCAL)	Data and Clock Lanes Acquisition with Data As Trigger Negative Slope_Init	Vertical Scale Data (mV)	<ul style="list-style-type: none"> <li>■ 10</li> </ul>
		Vertical Scale Clock (mV)	<ul style="list-style-type: none"> <li>■ 500</li> </ul>
		Vertical Offset (V)	0
		Horizontal Scale (us)	<ul style="list-style-type: none"> <li>■ 8</li> <li>■ 10</li> <li>■ 12</li> </ul>
		Sample Rate (GS/s)	<ul style="list-style-type: none"> <li>■ 12.5</li> <li>■ 25</li> </ul>
		Record Length	<ul style="list-style-type: none"> <li>■ 100000</li> <li>■ 10000000</li> </ul>

Test Name	Acquire Type	Parameter Name	Values
		Trigger Type: Transition	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Transition: Greater Than, Less Than, Occurs</li> <li>■ Delta Time: 500, 250</li> <li>■ Qualify: Occurs, Logic</li> <li>■ Lower Level: 0.05</li> <li>■ Upper Level: 1</li> </ul>
		Trigger Type: Edge	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Level: 0.1 V to 1 V</li> </ul>
		Trigger Type: Width	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Polarity: Positive, Negative</li> <li>■ Lower Limit: 25.0E-6</li> <li>■ Upper Limit: 300.0E-6</li> </ul>
1.5.6 Periodic HS Skew Calibration Burst (TSKEWCAL-SYNC, TSKEWCAL)	Data and Clock Lanes Acquisition with Data As Trigger Negative Slope	Vertical Scale Data (mV)	<ul style="list-style-type: none"> <li>■ 10</li> </ul>
		Vertical Scale Clock (mV)	<ul style="list-style-type: none"> <li>■ 500</li> </ul>
		Vertical Offset (V)	0
		Horizontal Scale (us)	<ul style="list-style-type: none"> <li>■ 8</li> <li>■ 10</li> <li>■ 12</li> </ul>
		Sample Rate (GS/s)	<ul style="list-style-type: none"> <li>■ 12.5</li> <li>■ 25</li> </ul>
		Record Length	<ul style="list-style-type: none"> <li>■ 100000</li> <li>■ 10000000</li> </ul>

Test Name	Acquire Type	Parameter Name	Values
		Trigger Type: Transition	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Transition: Greater Than, Less Than, Occurs</li> <li>■ Delta Time: 500, 250</li> <li>■ Qualify: Occurs, Logic</li> <li>■ Lower Level: 0.05</li> <li>■ Upper Level: 1</li> </ul>
		Trigger Type: Edge	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Level: 0.1 V to 1 V</li> </ul>
		Trigger Type: Width	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Polarity: Positive, Negative</li> <li>■ Lower Limit: 25.0E-6</li> <li>■ Upper Limit: 300.0E-6</li> </ul>
1.6.1 INIT LP-TX Initialization Period (TINIT,MASTER)	Data and Clock Lanes Acquisition In Escape Mode with Data As Trigger6_1	Vertical Scale Data (mV)	<ul style="list-style-type: none"> <li>■ 10</li> </ul>
		Vertical Scale Clock (mV)	<ul style="list-style-type: none"> <li>■ 500</li> </ul>
		Vertical Offset (V)	0
		Horizontal Scale (us)	<ul style="list-style-type: none"> <li>■ 8</li> <li>■ 10</li> <li>■ 12</li> </ul>
		Sample Rate (GS/s)	<ul style="list-style-type: none"> <li>■ 12.5</li> <li>■ 25</li> </ul>
		Record Length	<ul style="list-style-type: none"> <li>■ 100000</li> <li>■ 10000000</li> </ul>

Test Name	Acquire Type	Parameter Name	Values
		Trigger Type: Transition	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Transition: Greater Than, Less Than, Occurs</li> <li>■ Delta Time: 500, 250</li> <li>■ Qualify: Occurs, Logic</li> <li>■ Lower Level: 0.05</li> <li>■ Upper Level: 1</li> </ul>
		Trigger Type: Edge	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Level: 0.1 V to 1 V</li> </ul>
		Trigger Type: Width	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Polarity: Positive, Negative</li> <li>■ Lower Limit: 25.0E-6</li> <li>■ Upper Limit: 300.0E-6</li> </ul>
1.6.2 ULPS Entry Verification of Clock Lane LP-TX ULPS support	Clock Lanes Acquisition In Escape Mode with Clock As Trigger	Vertical Scale Data (mV)	<ul style="list-style-type: none"> <li>■ 10</li> </ul>
		Vertical Scale Clock (mV)	<ul style="list-style-type: none"> <li>■ 500</li> </ul>
		Vertical Offset (V)	0
		Horizontal Scale (us)	<ul style="list-style-type: none"> <li>■ 8</li> <li>■ 10</li> <li>■ 12</li> </ul>
		Sample Rate (GS/s)	<ul style="list-style-type: none"> <li>■ 12.5</li> <li>■ 25</li> </ul>
		Record Length	<ul style="list-style-type: none"> <li>■ 100000</li> <li>■ 10000000</li> </ul>

Test Name	Acquire Type	Parameter Name	Values
		Trigger Type: Transition	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Transition: Greater Than, Less Than, Occurs</li> <li>■ Delta Time: 500, 250</li> <li>■ Qualify: Occurs, Logic</li> <li>■ Lower Level: 0.05</li> <li>■ Upper Level: 1</li> </ul>
		Trigger Type: Edge	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Level: 0.1 V to 1 V</li> </ul>
		Trigger Type: Width	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Polarity: Positive, Negative</li> <li>■ Lower Limit: 25.0E-6</li> <li>■ Upper Limit: 300.0E-6</li> </ul>
1.6.3 ULPS Exit Transmitted TWAKEUP Interval	Data and Clock Lanes Acquisition in Escape Mode with Data As Trigger6_3	Vertical Scale Data (mV)	<ul style="list-style-type: none"> <li>■ 10</li> </ul>
		Vertical Scale Clock (mV)	<ul style="list-style-type: none"> <li>■ 500</li> </ul>
		Vertical Offset (V)	0
		Horizontal Scale (us)	<ul style="list-style-type: none"> <li>■ 8</li> <li>■ 10</li> <li>■ 12</li> </ul>
		Sample Rate (GS/s)	<ul style="list-style-type: none"> <li>■ 12.5</li> <li>■ 25</li> </ul>
		Record Length	<ul style="list-style-type: none"> <li>■ 100000</li> <li>■ 10000000</li> </ul>



Test Name	Acquire Type	Parameter Name	Values
		Trigger Type: Transition	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Transition: Greater Than, Less Than, Occurs</li> <li>■ Delta Time: 500, 250</li> <li>■ Qualify: Occurs, Logic</li> <li>■ Lower Level: 0.05</li> <li>■ Upper Level: 1</li> </ul>
		Trigger Type: Edge	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Level: 0.1 V to 1 V</li> </ul>
		Trigger Type: Width	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Polarity: Positive, Negative</li> <li>■ Lower Limit: 25.0E-6</li> <li>■ Upper Limit: 300.0E-6</li> </ul>
1.6.4 BTA TX-Side TTA-GO Interval Value	Data and Clock Lanes Acquisition in Escape Mode with Data As Trigger6_4to6	Vertical Scale Data (mV)	<ul style="list-style-type: none"> <li>■ 10</li> </ul>
1.6.5 BTA RX-Side TTA-SURE Interval Value		Vertical Scale Clock (mV)	<ul style="list-style-type: none"> <li>■ 500</li> </ul>
1.6.6 BTA RX-Side TTA-GET Interval Value		Vertical Offset (V)	0
		Horizontal Scale (us)	<ul style="list-style-type: none"> <li>■ 8</li> <li>■ 10</li> <li>■ 12</li> </ul>
		Sample Rate (GS/s)	<ul style="list-style-type: none"> <li>■ 12.5</li> <li>■ 25</li> </ul>
		Record Length	<ul style="list-style-type: none"> <li>■ 100000</li> <li>■ 10000000</li> </ul>

Test Name	Acquire Type	Parameter Name	Values
		Trigger Type: Transition	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Transition: Greater Than, Less Than, Occurs</li> <li>■ Delta Time: 500, 250</li> <li>■ Qualify: Occurs, Logic</li> <li>■ Lower Level: 0.05</li> <li>■ Upper Level: 1</li> </ul>
		Trigger Type: Edge	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Slope: Positive, Negative, Either</li> <li>■ Level: 0.1 V to 1 V</li> </ul>
		Trigger Type: Width	<ul style="list-style-type: none"> <li>■ Source: Dp, Dn, Clkp, Clkn</li> <li>■ Polarity: Positive, Negative</li> <li>■ Lower Limit: 25.0E-6</li> <li>■ Upper Limit: 300.0E-6</li> </ul>

## Examples

This section provides the examples for the SCPI commands.

Example	Description
TEKEXP:*IDN?\n	It returns the active TekExpress application name running on the scope.
TEKEXP:*OPC?\n	It returns the last command execution status.
TEKEXP:ACQUIRE_MODE PRE-RECORDED\n	It sets the acquire mode as pre-recorded.
TEKEXP:ACQUIRE_MODE?\n	It returns LIVE when acquire mode is set to live.
TEKEXP:EXPORT REPORT\n	It returns the report file in bytes. This can be written into another file for further analysis.
TEKEXP:INFO? REPORT\n	It returns "100,"ReportFileName.mht", when 100 is the file size in bytes for the filename ReportFileName.
TEKEXP:INFO? WFM\n	It returns "100,"WfmFileName1.wfm";"200,"WfmFileName2.wfm" when 100 is the filesize in bytes for the filename WfmFileName1.wfm and 200 is the file size in bytes for the filename WfmFileName2.wfm.
TEKEXP:INSTRUMENT "Real Time Scope",MSO58 ( GPIB8::1::INSTR )\n	It sets the instrument value as MSO58 ( GPIB8::1::INSTR ) for the selected instrument type Real Time Scope.
TEKEXP:INSTRUMENT? "Real Time Scope"\n	It returns "MSO56 ( GPIB8::1::INSTR ), when MSO56 ( GPIB8::1::INSTR )" is the selected instrument for the instrument type Real Time Scope.

Example	Description
TEKEXP:LASTERROR?\n	It returns ERROR: INSTRUMENT_NOT_FOUND, when no instrument is found.
TEKEXP:LIST? DEVICE\n	It returns "TX-Device,RX-Device" when TX-Device, RX-Device are the available device.
TEKEXP:LIST? INSTRUMENT,"Real Time Scope"\n	It returns "MSO58 ( GPIB8::1::INSTR ),MSO56 ( TCPIP::134.64.248.91::INSTR )" when MSO58 ( GPIB8::1::INSTR ), MSO56 ( TCPIP::134.64.248.91::INSTR ) are the list of available instruments.
TEKEXP:MODE COMPLIANCE\n	It sets the execution mode as compliance.
TEKEXP:MODE?\n	It returns COMPLIANCE when the execution mode is compliance.
TEKEXP:POPOPUP "OK"\n	It sets OK as the response to active popup in the application.
TEKEXP:POPOPUP?\n	It returns "OK", when OK is the active popup information shown in the application.
TEKEXP:REPORT GENERATE\n	It generates report for the current session.
TEKEXP:REPORT? "Scope Model"\n	It returns "MSO54" when MSO54 is the scope model.
TEKEXP:REPORT? "DUT ID"\n	It returns "DUT001" when DNI_DUT001 is the DUT ID.
TEKEXP:RESULT? "1.1.1 Data Lane LP-TX Thevenin Output High Level voltage (VOH)"\n	It returns Pass when the test result is Pass.
TEKEXP:RESULT? "1.1.1 Data Lane LP-TX Thevenin Output High Level voltage (VOH)","Margin",1\n	It returns "L:-50.000ps H:2000.000ps" when L:-50.000ps H:2000.000ps is the value.
TEKEXP:SELECT DEVICE, TX_Device, TRUE\n	It selects TX_Device
TEKEXP:SELECT? DEVICE\n	It returns "TX-Device" when TX-Device is the selected device type.
TEKEXP:SETUP DEFAULT\n	It restores the application to default setup.
TEKEXP:STATE STOP\n	It stops the test execution.
TEKEXP:STATE?\n	It returns as READY when the application is ready to run next measurement.
TEKEXP:STATE? SETUP\n	It returns as NOT_SAVED when the current setup is not saved.
TEKEXP:VALUE GENERAL," Test Mode", "Escape"\n	It sets the Test Mode parameter value to Escape
TEKEXP:VALUE? GENERAL,"Test Mode"\n	It returns "Escape" when Escape is the Test Mode value.
TEKEXP:SELECT TEST ,"1.1.1 Data Lane LP-TX Thevenin Output High Level voltage (VOH)",True	Execute this command to select an individual test. This command will select "1.1.1 Data Lane LP-TX Thevenin Output High Level Voltage (VOH)" test in the Signal Test tab.
TEKEXP:VALUE ACQUIRE," 1.1.1 Data Lane LP-TX Thevenin Output High Level voltage (VOH)","Data Lanes Acquisition In Escape Mode with Data As Trigger","Sample Rate (GS/s)",12.5	It sets Sample Rate Parameter value to 12.5
TEKEXP:VALUE ACQUIRE?" 1.1.1 Data Lane LP-TX Thevenin Output High Level voltage (VOH)","Data Lanes Acquisition In Escape Mode with Data As Trigger"," Sample Rate (GS/s)"	It returns the value of Sample Rate



# References

## Pre-recorded waveform file names for test measurements

The following table specifies the waveforms to load for the selected Acquire Type.

Acquire Type	Waveforms to be loaded
Data Lanes Acquisition In Escape Mode with Data As Trigger	DP and DN waveforms captured in escape mode. <ul style="list-style-type: none"> <li>■ Lane X..._DP.wfm</li> <li>■ Lane X..._DN.wfm</li> </ul>
Clock Lanes Acquisition In Escape Mode with Clock As Trigger	Clkp (or CP) and Clkn (or CN) waveforms captured in escape mode. <ul style="list-style-type: none"> <li>■ Clock..._Clkp.wfm</li> <li>■ Clock..._Clkn.wfm</li> </ul>
Data and Clock Lanes Acquisition with Data As Trigger	DP, DN, Clkp (or CP) and Clkn (or CN) waveforms captured in normal mode <ul style="list-style-type: none"> <li>■ Lane X..._DP.wfm</li> <li>■ Lane X..._DN.wfm</li> <li>■ Lane X..._Clkp.wfm</li> <li>■ Lane X..._Clkn.wfm</li> </ul>
Clock Lanes Acquisition with Clock As Trigger	Clkp (or CP) and Clkn (or CN) waveforms captured in normal mode. <ul style="list-style-type: none"> <li>■ Clock..._Clkp.wfm</li> <li>■ Clock..._Clkn.wfm</li> </ul>
Data and Clock Lanes Acquisition In Escape Mode with Data As Trigger6_1	DP, DN, Clkp (or CP) and Clkn (or CN) waveforms <ul style="list-style-type: none"> <li>■ Lane X..._DP.wfm</li> <li>■ Lane X..._DN.wfm</li> <li>■ Lane X..._Clkp.wfm</li> <li>■ Lane X..._Clkn.wfm</li> </ul>
Data and Clock Lanes Acquisition in Escape Mode with Data As Trigger6_3	DP, DN, Clkp (or CP) and Clkn (or CN) waveforms <ul style="list-style-type: none"> <li>■ Lane X..._DP.wfm</li> <li>■ Lane X..._DN.wfm</li> <li>■ Lane X..._Clkp.wfm</li> <li>■ Lane X..._Clkn.wfm</li> </ul>

Acquire Type	Waveforms to be loaded
Data and Clock Lanes Acquisition in Escape Mode with Data As Trigger6_4to6	DP, DN, Clkp (or CP) and Clkn (or CN) waveforms <ul style="list-style-type: none"><li>■ Lane X..._DP.wfm</li><li>■ Lane X..._DN.wfm</li><li>■ Lane X..._Clkp.wfm</li><li>■ Lane X..._Clkn.wfm</li></ul>
Data and Clock Lanes Acquisition with Data As Trigger Negative	<ul style="list-style-type: none"><li>■ Slope_int</li><li>■ Slope</li></ul>

---

# Index

## A

- Acquire parameters
  - including in test reports, 29
  - viewing in reports, 31
- Acquire Step by Step, 19
- Acquisition tab, 19
- Activate the TekExpress D-PHY license, 4
- Analysis options, 24
- Application panels overview, 14
- ApplicationStatus(), 68

## B

- Bandwidth Limit
  - clock channels, 16
  - data channels, 16
- Button
  - clear log, 25
  - Email settings, 24
  - save, 25

## C

- CheckSessionSaved(), 75
- Clearing test results, 38
- Compensate the signal path, 36
- Configuration tab parameter
  - instruments detected, 20
- Configuration tab parameters
  - global settings, 20
- Connect(), 44
- Connected instruments
  - searching for, 11, 12
- Contacting Tektronix, 2
- Conventions, 1
- Create a test setup from default settings, 40
- Create a test setup using an existing one, 40

## D

- Default directory usage, 5
- Directories and usage, 5
- Disconnect(), 77–79
- DUT ID, 16

- DUT parameter
  - device, 16
  - device profile, 16
- DUT type
  - device, 16

## E

- Email notification and setup, 13
- Exiting the application, 8

## F

- File name extensions, 5
- Flowchart for Client Programmatic Interface, 42

## G

- GetAcquireParameter(), 54
- GetAnalyzeParameter(), 54
- GetCurrentStateInfo(), 68
- GetDutId(), 48
- GetGeneralParameter(), 54
- GetPassFailStatus(), 71
- GetReportParameter(), 72
- GetResultsValue(), 71
- GetResultsValueForSubMeasurements(), 72
- GetTimeOut(), 66
- Global controls, 10
- GPIO, 11

## I

- Installing the software
  - switch matrix application, 3
- Instruments
  - discovering connected, 11
  - viewing connected, 12
- Instruments detected, 20
- Interface error codes, 80

## L

- LAN, 11

- Link Lane Setup, 17
- Live waveforms, 16
- Loading a test setup, 39
- LockSession(), 46
- Log view
  - save file, 25

## M

- Menus
  - Options, 10
  - Preferences, 38
- My TekExpress folder
  - files stored in, 28

## N

- Non-VISA, 11

## O

- Opening a saved test setup, 39
- Option Installation wizard, 4
- Options menu
  - Instrument control settings, 11
- Oscilloscope compensation, 36

## P

- Panels, 14
- Pattern, 16
- Pattern type, 16
- Pre-recorded waveform file names for test measurements, 123
- Preferences menu, 27, 29
- Preferences tab
  - send an Email, 24
  - setup panel, 24
- Program example, 81

## Q

- QueryStatus(), 68

## R

- Reactivate the TekExpress D-PHY license, 4

- Recalling a test setup, 39
- RecallSession(), 75
- Recommended accessories, 3
- Recommended probes, 3
- Related Documentation, 1
- Report contents, 32
- Report name, 30
- Report options, 29
- Report sections, 31
- Reports
  - receiving in email notifications, 13
- Reports panel, 14
- Results panel
  - summary of test results, 27, 29
  - test name, 27, 29
- Results Panel, 38
- Run button, 10
- Run(), 65
- Running tests, 38

## S

- Save log file, 25
- SaveSession(), 75
- SaveSessionAs(), 75
- Saving tests, 28
- SCPI commands
  - Command parameters list, 102
  - Examples, 120
  - TEKEXP:\*IDN?, 90
  - TEKEXP:\*OPC?, 90
  - TEKEXP:ACQUIRE\_MODE, 91
  - TEKEXP:ACQUIRE\_MODE?, 91
  - TEKEXP:EXPORT, 92
  - TEKEXP:INFO?, 92
  - TEKEXP:INSTRUMENT, 93
  - TEKEXP:INSTRUMENT?, 92
  - TEKEXP:LASTERROR?, 94
  - TEKEXP:LIST?, 94
  - TEKEXP:MODE, 95
  - TEKEXP:MODE?, 95
  - TEKEXP:POPOP, 96
  - TEKEXP:POPOP?, 96
  - TEKEXP:REPORT, 97



- TEKEXP:REPORT?, 97
- TEKEXP:RESULT?, 98
- TEKEXP:SELECT, 98
- TEKEXP:SELECT?, 99
- TEKEXP:SETUP, 99
- TEKEXP:STATE, 100
- TEKEXP:STATE?, 100
- TEKEXP:VALUE, 100
- TEKEXP:VALUE?, 101
- Search for connected instruments, 12
- Selecting test report contents, 29
- Selecting tests, 18
- SelectSingleTest(), 58
- SelectTest(), 57, 59
- SendResponse(), 69
- Session folders and files, 28
- SetAcquireParameter(), 64
- SetAnalyzeParameter(), 64
- SetDutId(), 48
- SetGeneralParameter(), 50, 61
- SetTimeOut(), 66
- Setup
  - acquisition tab, 19
- Setup files, 39
- Setup panel, 14
- SetVerboseMode(), 47
- Show acquire parameters, 19
- Software installation
  - activate TekExpress D-PHY license, 4
  - switch matrix application, 3
- Source, 16
- Status panel
  - log view, 25
  - message history, 25
  - test status tab, 25
- Stop(), 65
- Support, 2
- Supported oscilloscopes, 3

**T**

- Technical support, 2
- Tek Link, 11

- TekExpress D-PHY license activation, 4
- Test Name, 19
- Test reports, 31
- Test results
  - send by email, 13
- Test Results
  - clearing displayed, 38
- Test selection
  - TekExpress D-PHY, 18
  - test description, 18
- Test selection controls, 18
- Test setup files, 28, 39
- Test setups
  - load, 39
  - open, 39
  - recalling, 39
- Test status
  - acquire status, 25
  - analysis status, 25
  - auto scroll, 25
- Test-related files, 28
- Tests
  - running, 38
  - selecting, 18
- TransferImages(), 73
- TransferReport(), 73
- TransferWaveforms(), 73

## U

- UnlockSession(), 76
- User Comments
  - including in reports, 31

## V

- View a report, 31

## W

- Waveform files
  - locating and storing, 28
- Wavelength, 16

