TekExpress®
Thunderbolt Transmitter Automated Solution

Printable Online Help

077-0692-01

Tektronix

TekExpress®
Thunderbolt Transmitter Automated Solution

Printable Online Help

www.tektronix.com
077-0692-01

Tektronix

Contacting Tektronix

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

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

- In North America, call 1-800-833-9200.
- Worldwide, visit www.tektronix.com to find contacts in your area.

Table of Contents

Thunderbolt tests

Reference

Index

Related documentation

The following manuals are available as part of the TekExpress Thunderbolt Automated Solution documentation set.

Table 1: Product documentation

Item	Purpose	Location
Online Help	In-depth operation and UI help	Application Help menu
PDF of the Online Help	In-depth operation and UI help	Application Help menu Downloadable file from www.tektronix.com

See also

Technical Support (see page 2)

Conventions used in help

Online Help uses the following conventions:

- The term “DUT” is an abbreviation for Device Under Test.
- The terms “click” and “select” are generic terms that mean to select an item in the application user interface (UI). You can use a mouse or the touch screen to select UI items.

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 for more 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
- Probes used
- Your name, company, mailing address, phone number, FAX number
- Please indicate if you would like to be contacted by Tektronix about your suggestion or comments.

Application specific information

- Software version number
- Description of the problem such that technical support can duplicate the problem
- If possible, save the setup files for all the instruments used and the application.
- If possible, save the TekExpress setup files, log.xml, *.TekX (session files and folders), and status messages text file.

Install the Thunderbolt software

The TekExpress Thunderbolt web installer page provides a link to the software package Tbolt WebInstaller 1.0.1.X.exe used to install the application on a supported oscilloscope. For a list of compatible instruments, see Supported oscilloscopes and probes (see page 6).

To download and install the files:

1. Close the TekScope application.
2. Go to the www.tek.com Web site and search for Thunderbolt to locate the installation file. Download the file `Tbolt WebInstaller 1.0.1.X.exe`.
3. If you downloaded the file to a PC, copy the file to the oscilloscope.
4. Double-click the executable file to extract the installation files.

   After extraction, the installer launches and the software automatically installs in the following location:

   - Windows 7 location: `C:\Program Files (x86)\Tektronix\TekExpress\TekExpress Thunderbolt`
   - Windows XP location: `C:\Program Files\Tektronix\TekExpress\TekExpress Thunderbolt`
5. To run the application, open the TekScope application and select **Analyze > TekExpress Thunderbolt**.

See also

Minimum system requirements (see page 7)

Supported oscilloscopes and probes (see page 6)

Activate the license

Activate the license using the Option Installation wizard on the oscilloscope. The oscilloscope Online Help has instructions for using the Options Installation window to activate licenses for installed applications. Follow these steps to activate the TekExpress Thunderbolt license:

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

See also

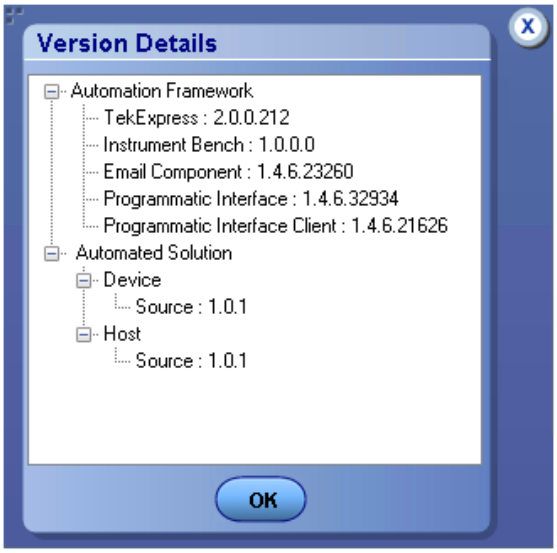
View software version and license information (see page 4)

View software version and license information

The following instructions show how to view version and license information for the application and associated modules such as the Programmatic Interface and the Programmatic Interface Client.

To view version information:

1. From the Options menu, select **About TekExpress**.
2. Click the **View Version Details** link to check the version numbers of the installed test suites. Close the dialog box when finished.



To view license information:

1. In the TekScope application, select **Help > About TekScope**.

   The Options section in the dialog box displays a list of installed options, including Thunderbolt: Transmitter solution.
2. Look in the **Option Installation Key** section for option key information.

See also

Activate the license (see page 3)

Options menu (see page 13)

Install the TenLira software and test scripts

TenLira is a software utility (provided by the Intel corporation) used to control the DUT. The electrical test scripts are used to control the Device and Host DUT.

NOTE. *Obtaining the TenLira software and compliance scripts requires that you have access to the Intel Business Link (IBL) portal to download the necessary files.*

TenLira software

1. Download the latest TenLira folder from the IBL library. Save the folder to your C: drive.
2. In the TenLira folder, open the SETUP_CTS_EDITION folder.
3. Double click TenLiraSetup_x86 bat file to start the installation.
4. Click NEXT and Close as needed.
5. When the window closes, go to the CMD window and follow the instructions until the installation is complete.
6. Upon completion, the oscilloscope automatically restarts. (Restart manually if necessary.)

Thunderbolt electrical compliance scripts

1. Download the latest Thunderbolt Electrical Compliance Scripts from the IBL library.
2. Navigate to the TenLira program folder.
   - Windows 7: C:\Program Files x86\Intel Corporation\TenLira
   - Windows XP: C:\Program Files\Intel Corporation\TenLira
3. Create a folder named “TBT Electrical”.
4. Place the Thunderbolt Electrical Compliance Scripts in the TBT Electrical folder.

The Thunderbolt application uses the following executables to control the Device DUT when using the Custom mode.

- TBT_Device_TX_PRBS11.exe
- TBT_Device_TX_PRBS31.exe
- TBT_Device_TX_SQ32.exe

Thunderbolt key features

Welcome to the TekExpress® Thunderbolt Transmitter Automated Solution application (Option TBT-TX). ThunderboltTX provides an automated, simple, and efficient way to test Thunderbolt Transmitter interfaces and devices consistent to the requirements of the Thunderbolt Specification v0.7.

Thunderbolt is based on TekExpress version 2, the Tektronix Test Automation Framework developed to support your current and future test automation needs. TekExpress uses a highly modular architecture that lets you quickly deploy automated test solutions for various standards.

Key Thunderbolt transmitter testing features include:

- Automated testing reduces the complexity of executing transmitter tests and enables you to test devices faster
- Seamless debug allows pause on each test in automation, and switch to DPOJET analysis tool for detailed debug

Supported oscilloscopes

The TekExpress Thunderbolt application runs on the following Tektronix oscilloscopes:

- DPO/DSA/MSO71604/B/C
- DPO/DSA/MSO72004/B/C
- DPO/DSA72504D
- DPO/DSA73304D

See also

Minimum system requirements (see page 7)

Minimum system requirements

The following table shows the minimum system requirements for an oscilloscope to run TekExpress.

Table 2: System requirements

Oscilloscope	■ Supported oscilloscopes (see page 6)
Processor	Same as the oscilloscope
Operating System	Same as the oscilloscope: ■ Windows XP (32-bit) SP2 and higher ■ Windows 7 32-bit or 64-bit
Memory	Same as the oscilloscope
Hard Disk	Same as the oscilloscope
Display	Same as the oscilloscope [1]
Firmware	TekScope v5.3.4.25 (Windows XP), TekScope v6.2.0.26 and above (Windows 7)
Software	■ National Instruments LabVIEW Runtime 9.0.1 or later ■ National Instruments TestStand engine 4.2.1 ■ Matlab Runtime engine v7.6 (Windows XP), Matlab Runtime engine v7.14 (Windows 7) ■ Microsoft .NET 4.0 Framework ■ Microsoft Excel 2002 or above ■ Microsoft Internet Explorer 6.0 SP1 or later ■ Adobe Reader 7.0 or equivalent software for viewing portable document format (PDF) files
DPOJET	v3.6.0.32 and above
Other Devices	■ Matched pair of SMA cables, one-set minimum for single lane ■ Microsoft compatible mouse or compatible pointing device

1 If TekExpress is running on an instrument having a video resolution lower than 800x600 (for example, sampling oscilloscope), it is recommended that you connect a secondary monitor. The secondary monitor must be configured and active before launching the application.

2 If TekExpress is installed on a Tektronix oscilloscope, TekExpress uses a virtual GPIB port to communicate with oscilloscope applications. If external GPIB communication devices such as USB-GPIB-HS or equivalent are used for instrument connectivity, make sure that the Talker Listener utility is enabled in the DPO/DSA/MSO oscilloscope GPIB menu. For ease of use, connect to an external (secondary) monitor.

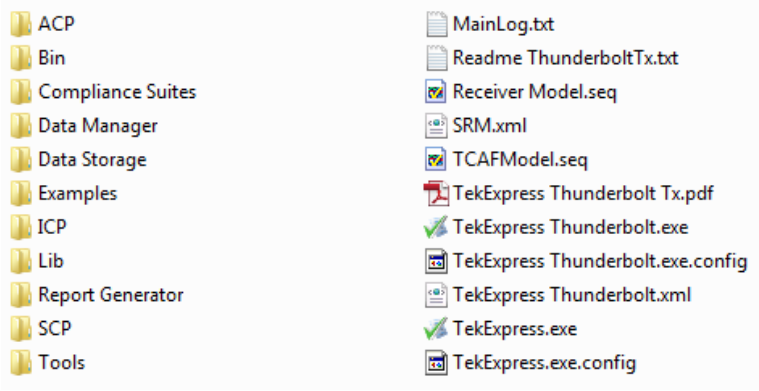
See also

Supported oscilloscopes (see page 6)

Equipment connection setup (see page 36)

Application directories and files

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

Directory names	Contains
InstallDir\TekExpress\TekExpress Thunderbolt	Application and associated files
TekExpress Thunderbolt\ACP	Instrument and Thunderbolt application-specific interface libraries
TekExpress Thunderbolt\Bin	Miscellaneous Thunderbolt application libraries
TekExpress Thunderbolt\Compliance Suites	Compliance-specific files
TekExpress Thunderbolt\Data Manager	Result management-specific libraries of the Thunderbolt application
TekExpress Thunderbolt\Data Storage	Libraries needed for storing data
TekExpress Thunderbolt\Examples	Various support files
TekExpress Thunderbolt\ICP	Instrument and Thunderbolt application-specific interface libraries
TekExpress Thunderbolt\Lib	Contains utility files specific to the Thunderbolt application
TekExpress Thunderbolt\Report Generator	Excel Active X interface Library for Report Generation
TekExpress Thunderbolt\SCP	Instrument and Thunderbolt application-specific interface libraries
TekExpress Thunderbolt\Tools	Instrument and Thunderbolt application-specific files

See also

View test-related files (see page 29)
File name extensions (see page 9)

File name extensions

The Thunderbolt application uses the following file name extensions:

File name extension	Description
.TekX	Saved session files. The extensions may not be displayed.
.seq	Test sequence files
.xml	Encrypted XML files that contain test-specific configuration information Note that the log file extension is also xml
.wfm	Test waveform files
.mht .xls .csv .html	Test result reports (default format).

See also

Select report options (see page 30)

View test-related files (see page 29)

Application directories and files (see page 8)

Run the application

To run the Thunderbolt application, do either of the following:

- Select **Analyze > TekExpress Thunderbolt** from the TekScope menu.
- Double-click any saved Thunderbolt session file.

When you open the application after installation, the application checks for a file called `Resources.xml` located in the `My TekExpress` folder. If this file is not found, instrument discovery is performed before launching Thunderbolt. The `Resources.xml` file contains information regarding instruments that are available on your network.

If the application license was not installed using the TekScope menu **Utilities > Option Installation** selection, you can open the application up to 10 times in evaluation mode. Each time you open the application without supplying a valid license key, one of the free trials is used.

See also

Activate the license (see page 3)

Exit the application

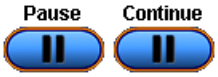
Use the following method to exit the application:

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

1. Click  on the application title bar.
2. Do one of the following:
   - If you have an unsaved session or test setup open, you are asked to save it before exiting. To save it, click **Yes**. Otherwise click **No**. The application closes.
   - A message box appears asking if you really want to exit TekExpress. To exit, click **Yes**.

Application controls

Table 4: Application control descriptions

Item	Description
Options menu (see page 13)	Opens the Options menu for access to global controls
Panels (see page 18)	Visual frames with sets of related options
Command buttons	Buttons that initiate an immediate action such as the Start, Stop, Pause, Continue, and Clear command buttons
Start button	 Starts continuous measurement acquisition and accumulation. If prior acquired measurements have not been cleared, the new measurements are added to the existing set.
Stop button	 Stops (aborts) the current measurement acquisition.
Pause \ Continue button	 Use the Pause button to temporarily interrupt the current acquisition. When a test is paused, the button name changes to Continue.
Clear button	 Clears all existing measurement results. Adding or deleting a measurement, or changing a configuration parameter of an existing measurement, also clears measurements. This prevents the accumulation of measurement statistics or sets of statistics that are not coherent. This button is available only on the Results panel (see page 27).
Clear Log	 This button is available only on the Status panel (see page 26).
Save	 This button is available only on the Status panel (see page 26).
Application window move icon	 Place the cursor over the three-dot pattern in the upper left corner of the application window. When the cursor changes to a hand, drag the window to the desired location.

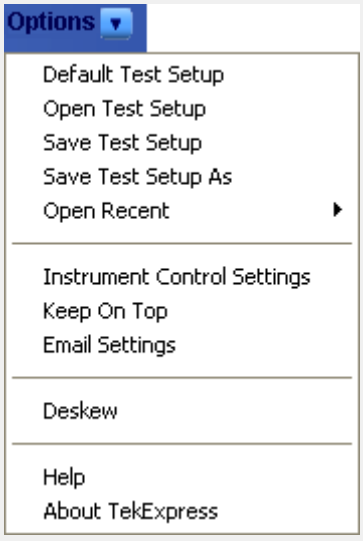
Options menu overview

The Options menu is located in the upper right corner of the application.

The Options menu (see page 13) has the following selections:

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 selections
Save Test Setup As	Creates a new test setup based on an existing one
Open Recent	Displays a menu of recently opened test setups from which to select
Instrument Control Settings (see page 14)	Shows the list of instruments connected to the test setup and allows you to locate and refresh connections to connected instruments
Keep On Top	Keeps the TekExpress Thunderbolt utility on top of other open windows on the desktop
Email settings dialog box (see page 14)	Use to configure email options for test run and results notifications
Deskew (see page 16)	Loads oscilloscope channel deskew settings into the application.
Help	Displays TekExpress Help
About TekExpress	■ Displays application details such as software name, version number, and copyright ■ Provides access to software version and license information (see page 4) for your Thunderbolt installation ■ Provides a link to the Tektronix Web site

Options menu

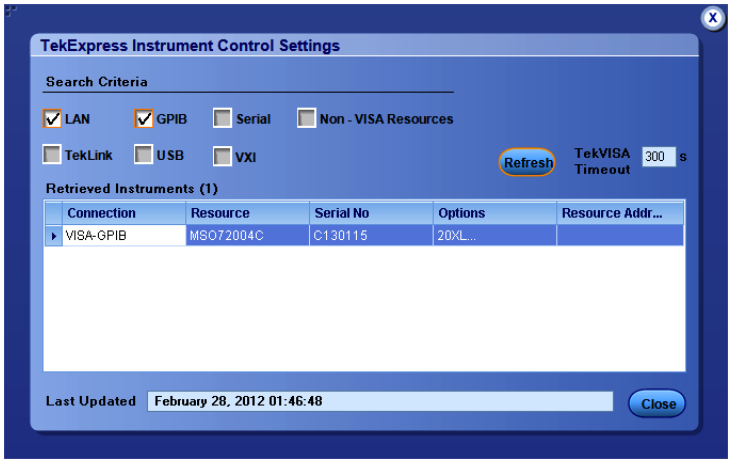


See also

Application controls (see page 12)

Instrument control settings

Use the TekExpress Instrument Control Settings dialog box to search for and list the connected resources (instruments) found on specified connections (LAN, GPIB, USB, and so on) and each instruments connection information. You access this dialog box from the Options menu.



Use the Instrument Control Settings feature to and view instrument connection details. Connected instruments displayed here can be selected for use in the Global Settings tab in the configuration section. See step 1 of Configure Tests (see page 24) for details.

See also

View connected instruments (see page 38)

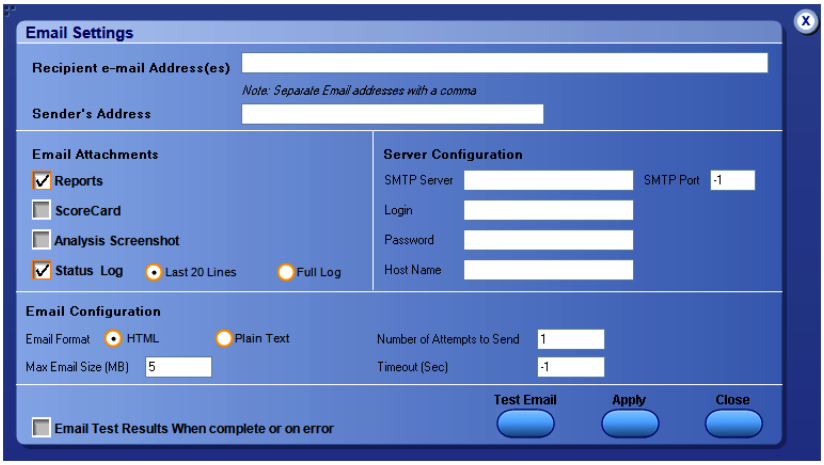
Options menu overview (see page 13)

Email Settings dialog box overview

Use the Email Settings dialog box to enable Thunderbolt to send an email message when a test completes, produces an error, or fails. Select the type of test run information to attach to the email (such as test reports and test logs), the email message format, and the email message size limit.

Open the Email Settings dialog box from the **Options** menu.

NOTE. *Recipient email address, sender's address, and SMTP Server are mandatory fields.*



See also

Configure email settings (see page 15)

Options menu (see page 13)

Select test notification preferences (see page 23)

Configure email settings

To be notified by email when a test completes, fails, or produces an error, configure the 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 email addresses to which to send the test notification. 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, the @ symbol and then the email server used. For example: DPO72004C_B130099@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**: Attach the test report to the notification email.
   - **Status Log**: Attach the test status log to 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:
   - Select the message file format to send: HTML (the default) or plain text.
   - Enter a maximum file size for the email message (message plus attachment files). 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.

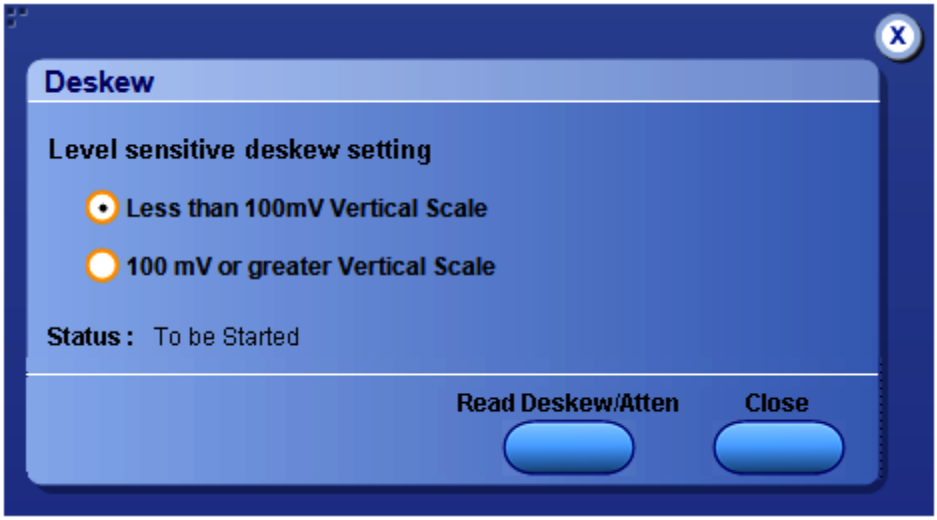
Deskew

The Deskew utility reads the instrument configuration and channel deskew settings from the connected oscilloscope and stores them in a file. When you run a test (click the Start button), Thunderbolt resets the oscilloscope to the factory default settings, loads the instrument configuration and channel deskew settings with the saved values, and then starts running the test session. This is to make sure that the instrument is set to a known state before each test run.

NOTE. *Make sure that you run the Thunderbolt Deskew utility after you have finalized the DUT setup, oscilloscope settings, and channel deskew values, and before you run compliance tests with that test configuration. See your oscilloscope user documentation or online help for information on channel deskew procedures.*

NOTE. *You will need to run the Deskew utility whenever you change the oscilloscope settings or channel deskew values for a test setup.*

NOTE. *Thunderbolt does not verify that the saved oscilloscope settings and deskew values are applicable to the current instrument attached to the application.*



1. In the TekExpress Thunderbolt application, select **Options > Deskew**.
2. Select the level sensitive deskew setting:
   - **Less than 100 mV Vertical Scale**: Select this if the oscilloscope vertical setting is less than 100 mV/division for the signal you are measuring.
   - **100 mV or greater Vertical Scale**: Select this if the oscilloscope vertical setting is greater than 100 mV/division for the signal you are measuring.
3. Click **Read Deskew/Attn**. The utility stores the instrument settings and deskew settings as follows:
   - **<100 mV**: C:\Program Files\Tektronix\TekExpress\TekExpress Thunderbolt\ICP\Deskew-Attenuation.txt
   - **≥100 mV**: C:\Program Files\Tektronix\TekExpress\TekExpress Thunderbolt\ICP\Deskew-Attenuation-GE100mV.txt
4. When the status in the dialog box indicates the deskew is finished, click **Close**.

Application panel overview

Panels group related configuration, test and results settings.

The TekExpress Thunderbolt panels are:

Table 5: Application panels

Panel name	Description
Setup (see page 18)	Set the DUT, test, acquisition, and report parameters.
Status (see page 26)	View the progress and analysis status of the selected tests, and view test logs.
Results (see page 27)	View a summary of test results and select results viewing preferences.
Reports (see page 29)	Browse for reports, save reports as specific file types, specify report naming conventions, select report content to include (such as summary information, detailed information, user comments, setup configuration, application configuration), and select report viewing options.

See also

Application controls (see page 12)

About setting up tests (see page 35).

Setup panel overview

The Setup panel contains sequentially ordered tabs that help guide you through a typical test setup process.

Set the DUT parameters (see page 18).

Select test(s) (see page 21).

Set lane acquisition source (see page 22).

Configure the selected tests (see page 24).

Select test notification preferences (see page 23).

Items selected in a preceding Setup tab may change options available in the following tabs. You can switch between the tabs in any order to modify your test parameters.

Set DUT parameters

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

1. Click **Setup > DUT**.
2. (Optional) Enter the ID for the device. The default value is DUT001. The DUT ID parameter is added to reports.
3. (Optional) To add comments to the test report, click the note pad icon () to the right of the DUT ID field. Enter comment text up to 256 characters. To enable or disable displaying comments in the test report, see Select report options (see page 30).
4. Select from the following parameters. Settings that do not apply to compliance testing cannot be changed and are grayed out.

Table 6: DUT tab settings

Setting	Description
Acquire live waveforms	Acquire active signals from the oscilloscope for testing.
Use pre-recorded waveform files	Run tests on a saved run session file. Select a run session file from the list.
View	Determines where to access the test configuration settings: ■ **Compliance**: View configuration settings by clicking **Setup > Test Selection > Configure** ■ **Advanced**: Enables the **Setup > Configuration** tab in which to view configuration settings.
Device	■ Device ■ Host
Version	Select the Thunderbolt testing specification version to use for testing. The latest version is the default.
Data Rates	Select the data rates to include in the tests. 10G is the only selection available at this time.
Patterns	All patterns are selected by default. To exclude a pattern from a test, clear the check boxes.
Number of Lanes to Test	Select the desired number of test lanes to use for this test session. The lanes shown here determines the number of test lanes you can select.

Table 6: DUT tab settings (cont.)

Setting	Description
Selected Test Lanes SETUP	Displays the test lanes selected for the test session. Click **Setup** to change lanes selected for testing. **a.** In the Test Lane Setup dialog box, select the desired number of lanes from the Link Width drop-down list. **b.** To select the lanes to use, click the corresponding lane buttons: - To select both lanes at once, click **Select All**. If you select this, select **2 Lanes** from the Number of Lanes to Test drop-down list. - To deselect all selected lanes, click **Deselect All**. - If you select 1 Lane, select the **Lane0** button. - If you select 2 Lanes, select **Lane0** and **Lane1**. **c.** Click **OK**. Your selections display in the Link Width section of the DUT tab.
DUT Automation	The DUT Automation field displays the type of DUT control currently in use. DUT controls are detected by through the Instrument Control Settings feature. - Manual: This option is displayed if your setup does not have the capability to control the DUT automatically. You will be prompted to configure the DUT manually during the test. You can change this to Custom if needed. - Custom: Select this option if you are using the TenLira custom utility to manage the DUT.

See also

About setting up tests (see page 35)

TenLira software installation (see page 5)

Select tests

Use the Test Selection tab to select the tests to run on the connected DUT.

1. Click **Setup > Test Selection**.
2. Select the test(s) to run:
   - Click one or more check boxes adjacent to each test.
   - Click **Deselect All** to deselect all tests. All tests are selected by default.
   - Click **Select Required** to select all tests that are required for compliance.
   - Click **Select All** to select all tests.
3. Configure the test(s):
   - Click **Configure** to open the configuration settings for a selected test.

   NOTE. *The Configure button is not displayed if the View in the DUT tab is set to Advanced.*

   - Click **Schematic** to display a schematic diagram that shows the DUT test setup. Use the diagram to verify the test setup before running the test.

See also

About setting up tests (see page 35)

Set acquisition parameters source

Use the Acquisitions tab to set the signal source (channel) used to acquire data. Acquisition options are available only when acquiring a live waveform.

1. Click **Setup > Acquisitions**.
2. Click the **Lane 0 (+)** and **Lane 0 (-)** fields to select the oscilloscope channels to use for those lanes.
3. Click on the **View Probes** button to view the probe configuration of each oscilloscope channel.
4. Select an **Acquisition and Save Options** to determine the order in which waveforms are acquired and analyzed.
   - Select **Save All Waveforms Before Analysis** to save all acquired waveforms before analysis begins.
   - Select **Analyze Immediately - No Waveform Saved** to perform an analysis without saving the waveform.
   - Select **Save and Analyze Acquisition in Sequence** to determine the order of acquisition and analysis during the test execution.
5. Select **Show Acquire Parameters** to show the acquisition parameters in the test list.
6. Select **Acquire Step By Step** to have the software prompt you to continue after each phase of the test completes.
7. Select a **Signal Validation** parameter:
   - Select **Prompt me if signal fails** to open a dialog box when the application fails to acquire a valid signal after a specified number of retries (as set in the Configuration tab or fields). Select one of the three options in the dialog box:
     - **Re-Acquire**: The application attempts to re-acquire the signal.
     - **Use Anyway**: Use the acquired signal for all applicable tests.
     - **Skip Test**: Skip (ignore) any test(s) that depend on this acquisition. Skipped tests are listed in the status panel and in the report.
   - Select **Use signal as is - Don't Check** to skip signal validation and use the signal as-is for testing. The test results may not be as expected.

Set test notification preferences

Use the Preferences tab to set the application to send an email when a test measurement completes:

1. Click **Setup > Preferences**.
2. Select **Highlight yellow warning if measured value is within [xx}% of limit** check box to display a yellow warning if the measured value is within the percent of limit value.

   If you select this option, enter the margin limit value.
3. Select the **On Test Failure, send me an email** check box to receive an email when a test fails.

   If you select this option, select the **Email Test Results when complete or on error** check box in the Email Settings dialog box. Click **Email Settings** to configure the email settings (see page 15).

See also

About setting up tests (see page 35)

Select report options (see page 30)

About configuring test parameters

Use the configuration settings to view the measurement parameters for selected tests. How the test configurations are accessed depend on the View selected in the DUT tab.

- If you selected Compliance View in the DUT tab, then in the Test Selection tab, select the desired test in the list and then click the Configure button.
- If you selected Advanced View in the DUT tab, click the Configuration tab in the Setup panel.

See also

Configure tests parameters

About running tests (see page 41)

Configure test parameters

The Configuration parameters let you set global and individual test parameters. To return to test selection from the Configuration tab or panel, click the **Test Selection** button.

NOTE. *You cannot change test parameters that are grayed out.*

1. Modify Global settings (see page 24) as desired:
   - To select the instruments for testing, click **Global Settings**. In the Instruments Detected section, click in the shaded areas to activate the drop-down lists and select an instrument. If you do not see the desired instrument in the list, refresh the list (see page 38).
2. To modify any individual test measurement settings, click **Measurements**, select the test in the tree view, and change the settings.

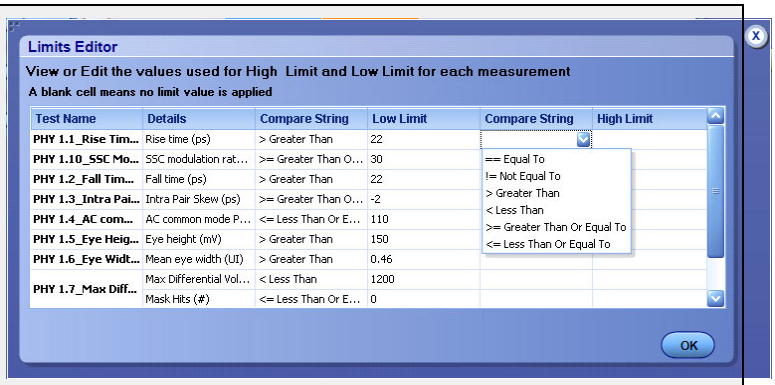
See also

About Setting Up Tests (see page 35)

Common test parameters

The following table lists the settings and parameters common to all tests.

Table 7: Common parameters and values

Parameter type	Parameter
Mode	Determines whether test parameters are in compliance mode or can be edited (User Defined Mode). ■ Compliance: Most test parameter values cannot be edited. ■ User Defined: Most test parameter values can be edited.
Global Settings	These settings apply to all tests selected for the current session. You can change only some of these settings. ■ Scope Settings: Specifies the instrument channel to use as the source for the trigger. This can be changed. ■ Channel Amplitude: The value of the amplitude to be set for the selected channel.
Measurements	These settings apply to the test selected in the tree view of the configuration section. For details, see Measurement Parameter Descriptions (see page 88).
Limits Editor	Shows the upper and lower limits for the applicable measurement using different types of comparisons. In Compliance Mode, you are able to view the measurement high and low limits used for the test selected in the tree view of the Measurements tab. When running tests in User Defined Mode, you can edit the limit settings (see page 25) in the Limits Editor. When running tests in User Defined Mode, the cells in the Limits Editor table are active for editing so you can change parameters.  When running tests in User Defined Mode, the cells in the Limits Editor table are active for editing so you can change parameters.

See also

Configure tests (see page 24)

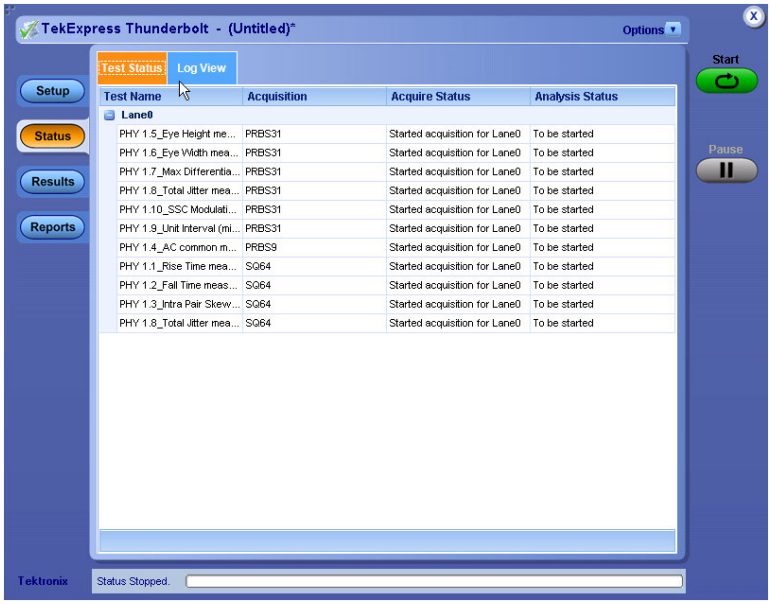
Status panel overview

The Status panel provides status on test acquisition and analysis (Test Status tab) 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.

The Test Status tab lists a high level status for each test.

Table 8: Test Status tab

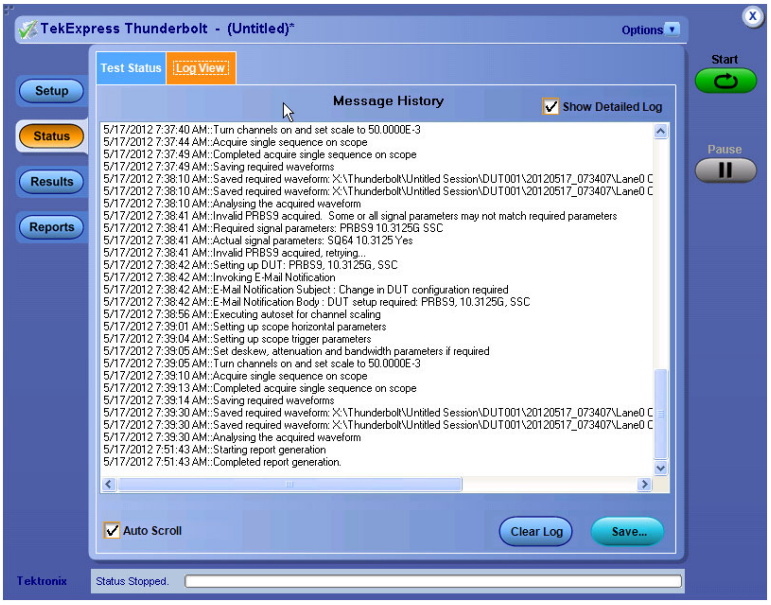
Column	Description
Test Name	Name of the test
Acquire Status, Analysis Status	Status of the signal acquisition or test analysis ■ To be started ■ In progress ■ Testing ■ Completed ■ Aborted



The Log View tab provides a list of the actions executed during the test. Use this information to review or troubleshoot tests.

Table 9: Log View fields

Item	Description
Message History	Displays all run messages with timestamp information.
Show Detailed Log	Records a detailed history of test execution.
Auto Scroll	Sets the program to automatically scroll down the Message History window as information is added to the log during the test.
Clear Log	Clears all messages in the Message History window.
Save	Saves the log file as a text file for examination. Displays a standard Save File window and saves the status messages in the file that you specify.

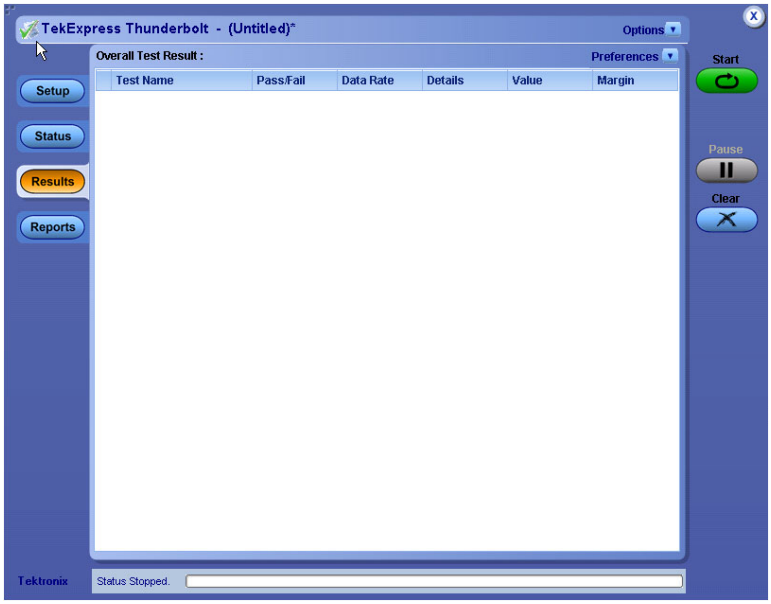


See also

View test results (see page 28)

Results panel overview

When a test finishes, the application switches to the Results panel to display a summary of test results. Set viewing preferences for this panel from the Preferences menu in the upper right corner. Viewing preferences include showing whether a test passed or failed, summary results or detailed results, and enabling wordwrap. For information on using this panel, see View test results (see page 28).



View test results

When a test finishes, the application switches to the Results panel (see page 27), which displays a summary of test results. The Overall Test Result is displayed at the top left of the Results table. If all of the tests for the session pass, the overall test result will be Pass. If one or more tests fail, the overall test result will show Fail.

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

See also

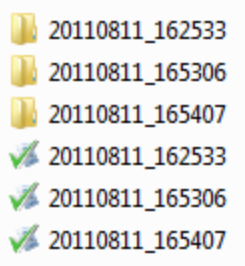
View a report (see page 32)

View test-related files

Files related to tests are stored in the `My TekExpress\Thunderbolt` folder. In the `Thunderbolt` folder, each test setup has a test setup file and a test setup folder, both with the test setup name. The test setup file is preceded by the Thunderbolt icon and usually has no visible file 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 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.



The first time you run a new, unsaved session, the session files are stored in the `Untitled Session` folder located at `..\My TekExpress\Thunderbolt`. When you name and save the session, the files are placed in a folder with the name that you specify.

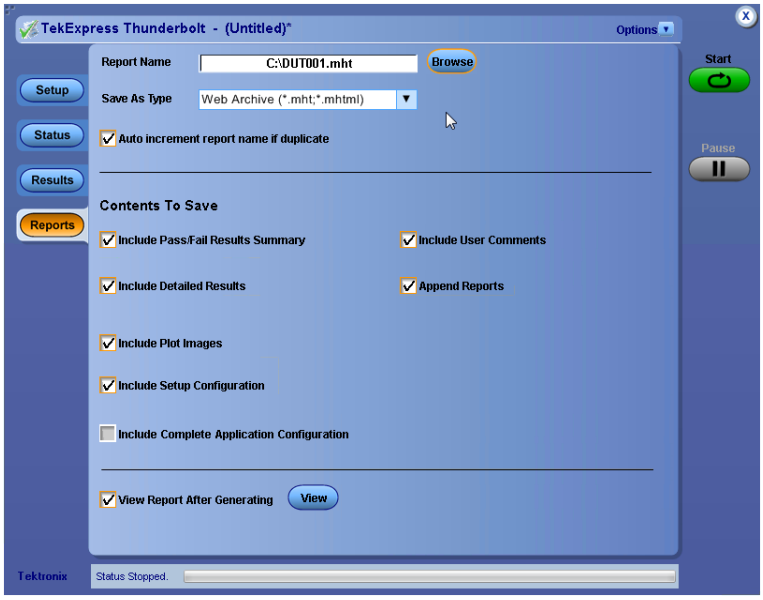
See also

File name extensions (see page 9)

Reports panel overview

Use the Reports panel to open reports, name and save reports, select report content to include, and select report viewing options.

For information on setting up reports, see Select report options (see page 30). For information on viewing reports, see View the report (see page 32).



Select report options

Use the Reports panel (see page 29) to select which test 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. Generally, you would 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 options:

Table 10: Report options

Setting	Description
Report Name	Displays the name and location from which to open a report. The default location is at \My TekExpress\Thunderbolt\Untitled Session. 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. Change the report name or location. Do one of the following: ■ In the Report Name field, type over the current folder path and name. ■ Double-click in the Report Name field. Make selections from the popup keyboard and then click the **Enter** button. Be sure to include the entire folder path, the file name, and the file type. For example: C:\Documents and Settings\your user name\My Documents\My TekExpress\Thunderbolt Tx\DUT001_Test_211.mht. ***NOTE.*** *You cannot change the file location using the Browse button.* Open an existing report. Click **Browse**, locate and select the report file and then click **View** at the bottom of the panel.
Save As Type	Saves a report in the specified file type. Lists supported file types to choose from. ***NOTE.*** *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.*
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.
Include Pass/Fail Results Summary	Sets the application to include the color block labeled Test Result (indicating whether the test passed or failed) in the report. For details, see Report Contents in View a Report (see page 32).
Include Detailed Results	Sets the application to include parameter limits, execution time, and test-specific comments generated during the test.
Include Setup Configuration	Sets the application to include information about hardware and software used in the test in the summary box at the top of the report. Information includes: the oscilloscope model and serial number, probe model and serial number, the oscilloscope firmware version, SPC and factory calibration status, and software versions for applications used in the measurements.
Include Complete Application Configuration	Sets the application to include a table listing general, common, and acquired parameters used in the test. This option is disabled by default.
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.
View Report After Generating	Automatically opens the report in your Web browser when the test completes. This option is selected by default.

See also

View a report (see page 32)

About setting up tests (see page 35)

View a report

The application automatically generates a report when test analysis is completed 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 if you want to view a different test report, do the following:

1. Click the **Reports > Browse** button and locate and select the report file to view.

NOTE. *If you did not save the test setup after running the test and you either closed the application or you ran another test, the report file was not saved.*

2. At the bottom of the Reports panel, click **View**.

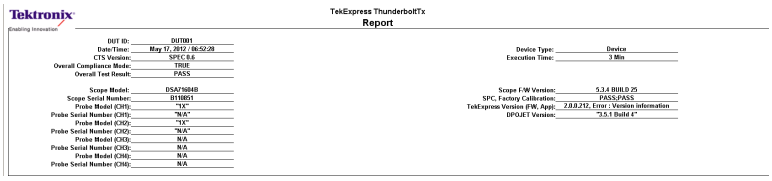
For information on changing the file type, file name, and other report options, see Select Report Options (see page 30).

Report contents

A report shows specified test details, as defined in the Reports panel.

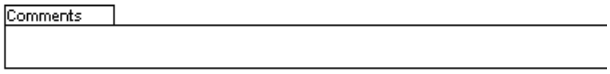
Setup Configuration Information

Setup configuration information is listed in the summary box at the top of the report. This information includes the oscilloscope model and serial number, probe model and serial number, and software versions. To exclude this information from the 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 of the Setup panel appear in the Comments section directly below the summary box.



Test results

This table lists the tests results. The contents of this table depend on the selections made in the Reports panel before running the test (Include Pass/Fail Results Summary, Include User Comments, Include Detailed Results, Include Complete Application Configuration, and so on).

If Include Plots Images is selected, then measurement waveform plots are shown below the test results table.

Test Name	Lane	Pattern	Speed	SSC	Measurement Details	Measured value	Units	Test Result	Margin
PHY 1.1_Rise Time measurement	Lane0	SQ64	10.3125G	SSC	RISETIME	54.827	ps	Pass	32.827
PHY 1.2_Fall Time measurement	Lane0	SQ64	10.3125G	SSC	FALLTIME	29.778	ps	Pass	7.778
PHY 1.3_Intra Pair Skew Test	Lane0	SQ64	10.3125G	SSC	SKEW(Fig-1)	13.421		Informative	13.421
PHY 1.4_AC common mode voltage Peak to	Lane0	PRBS9	10.3125G	SSC	PK2PK	47	mV	Pass	63
PHY 1.5_Eye Height measurement	Lane0	PRBS31	10.3125G	SSC	HEIGHT	393.159	mV	Pass	243.159
PHY 1.6_Eye Width measurement	Lane0	PRBS31	10.3125G	SSC	WIDTH	0.679	UI	Pass	0.219
PHY 1.7_Max Differential Voltage Measurement	Lane0	PRBS31	10.3125G	SSC	EYEHIGH(Fig-2)	213.005	mV	Informative	-N.A-
	Lane0	PRBS31	10.3125G	SSC	EYELOW	-209.147	mV	Informative	-N.A-
	Lane0	PRBS31	10.3125G	SSC	MASKHITS	0	#	Pass	0
	Lane0	PRBS31	10.3125G	SSC	MAXDIFFVGE	422.152	mV	Pass	777.848
PHY 1.8_Total Jitter measurement	Lane0	PRBS31	10.3125G	SSC	DJDIRAC	0.231		Informative	-N.A-
	Lane0	SQ64	10.3125G	SSC	RJDIRAC	0.007		Informative	-N.A-
	Lane0	SQ64	10.3125G	SSC	TJBER	0.341	UI	Pass	0.199
PHY 1.9_Unit Interval (min) Measurement	Lane0	PRBS31	10.3125G	SSC	PERIOD(Fig-3)	6222.707	ps	Fail	6126.251 , 6125.223

See also

View test results (see page 28)

View test-related files (see page 29)

About setting up tests

Set up tests using the tabs in the Setup panel. Settings in the DUT tab use a top-down, left-to-right logic flow, so that any parameter that affects or acts as a filter for other parameters appears either to the top of or to the left of the affected parameters.

Tests are saved when you save a test setup. To avoid overwriting test results, remember to assign a unique name to the test either before running it or immediately after.

See also

About test setups (see page 45)

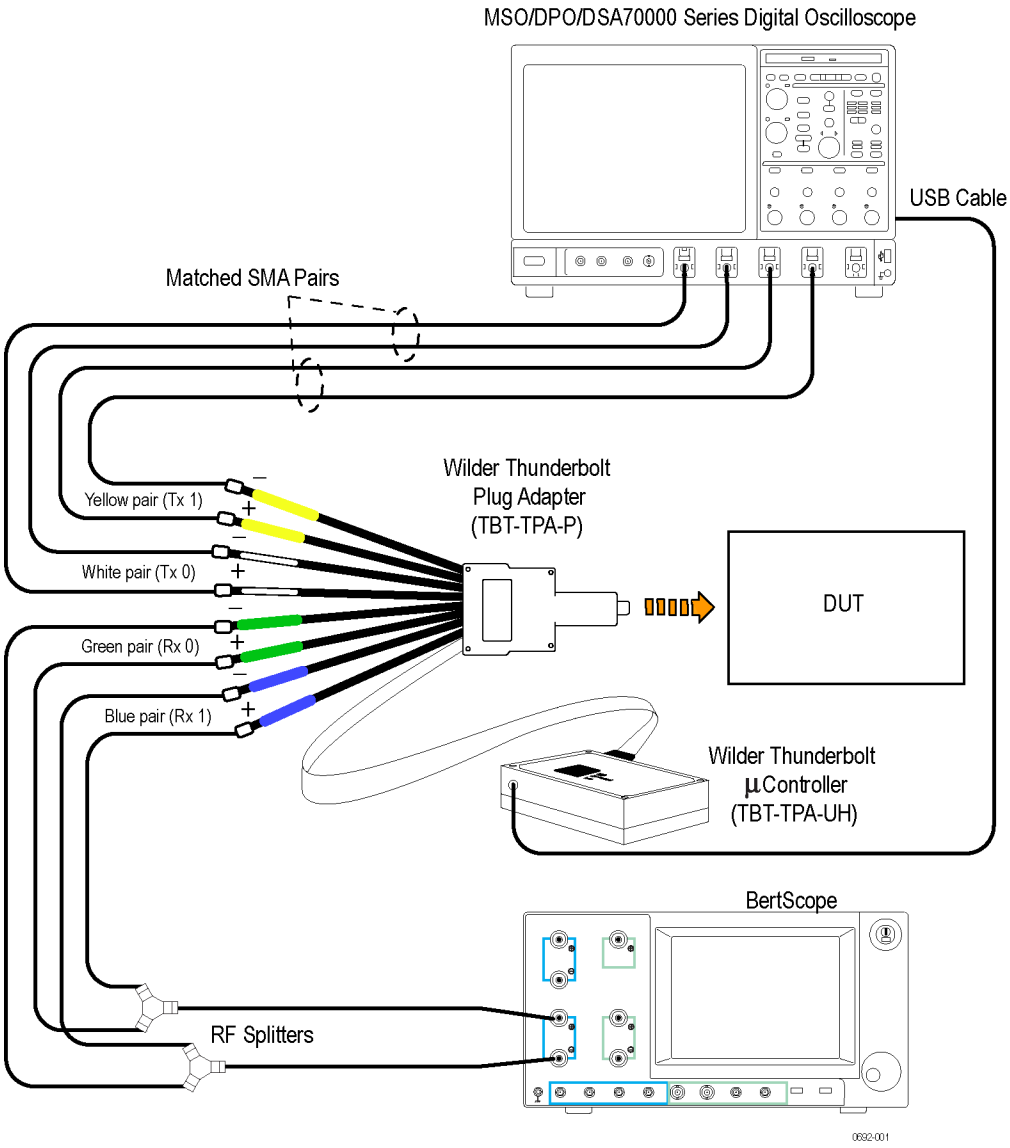
Before you click Start (see page 41)

About running tests (see page 41)

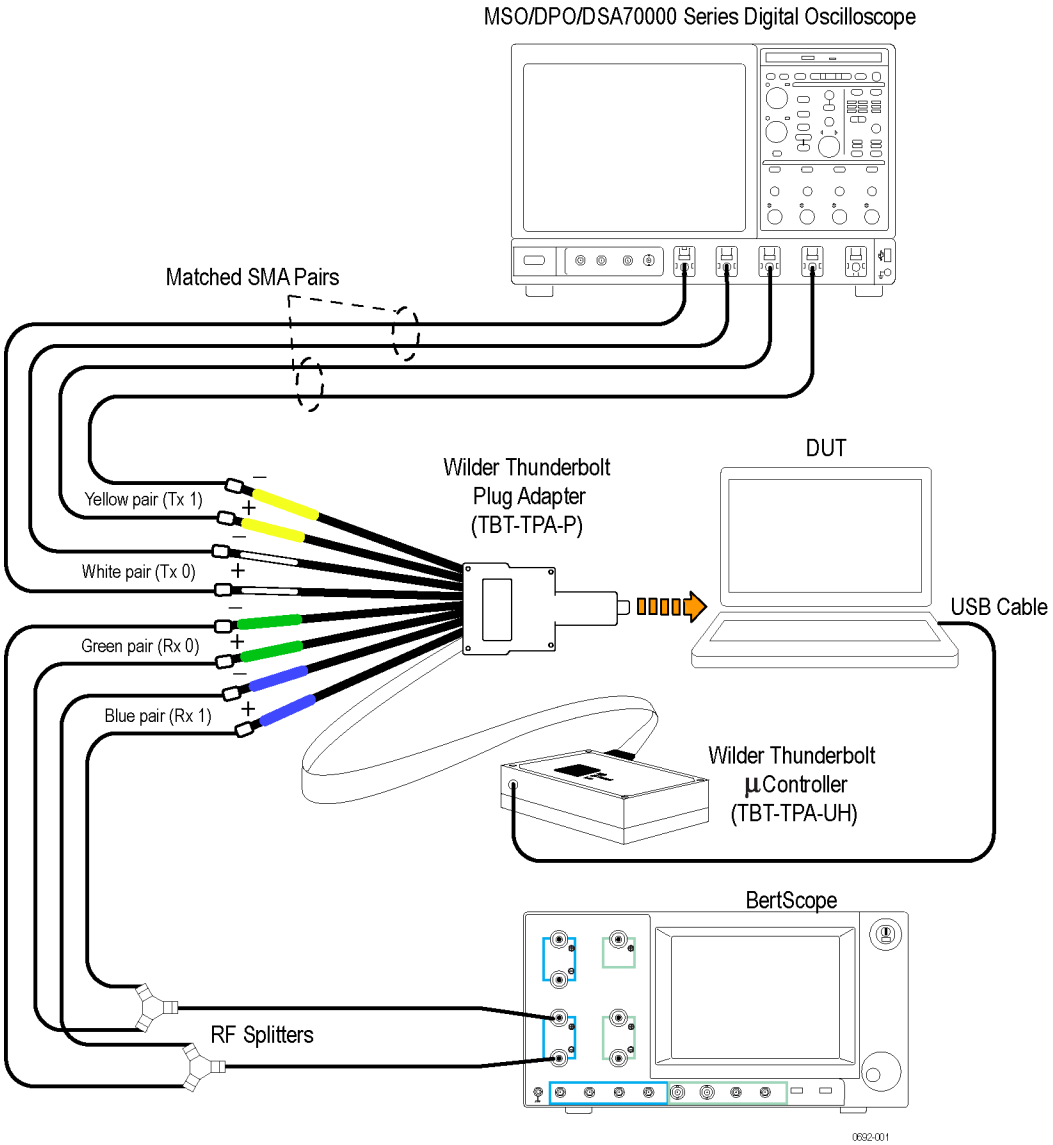
Equipment connection setup

The following diagrams shows how to connect the DUT to the oscilloscope for all the Thunderbolt measurements.

Device connection diagram



Host connection diagram



See also

[Minimum system requirements (see page 7)]
[View connected instruments (see page 38)]
[About setting up tests (see page 35)]

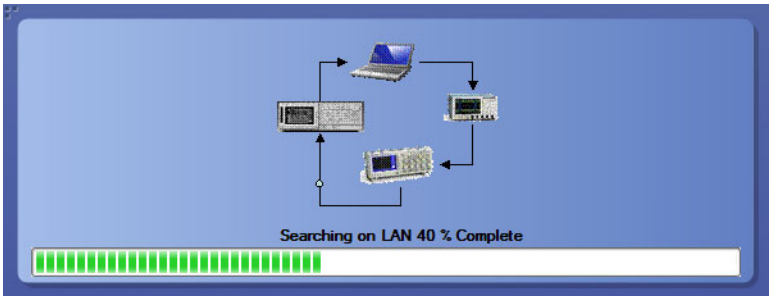
View connected instruments

Use the Instrument Control Settings dialog box to view or search for connected instruments required for the tests. The application uses TekVISA to discover the connected instruments.

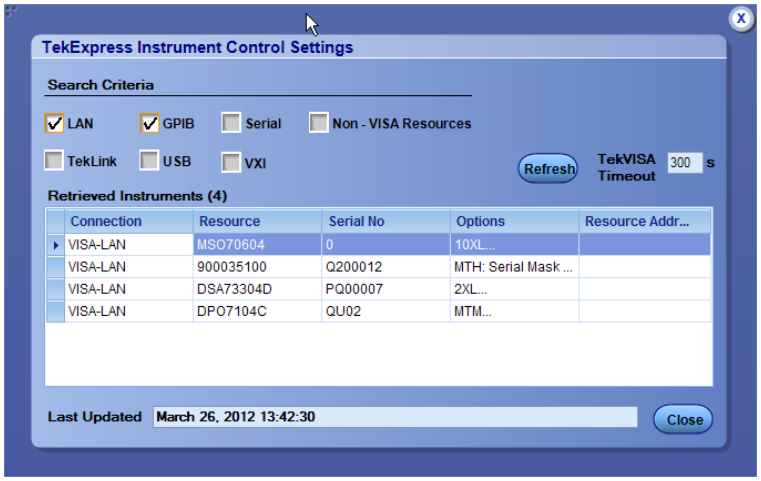
To refresh the list of connected instruments:

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 for.

   Instrument search is based on the VISA layer but different connected cables 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. If the search does not find any instruments that match a selected resource type, a message appears telling you that no such instruments were found.
3. Click **Refresh**. TekExpress searches for connected instruments.



4. After discovery, the dialog box lists the instrument-related details based on the search criteria you selected. For example, if you selected LAN and GPIB as the search criteria, the application checks for the availability of instruments over LAN, then GPIB.



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

Configure Tests (see page 24)

Equipment Setup (see page 36)

Test setup overview

A test setup includes configuration parameters and report options. Use the options in the Setup panel and Reports panel (see page 29) to select and configure tests.

1. Select the DUT parameters (see page 18).
2. Select one or more tests (see page 21).
3. Configure tests (see page 24).
4. Select test notification preferences (see page 23).
5. Select report options (see page 30).

See also

About test Setups (see page 45)

Pre-run checklist (see page 43)

Before you click Start (see page 41)

About running tests (see page 41)

About running tests

After selecting and configuring the test, review the pre-run checklist (see page 43) and then click **Start** to run the tests. While tests are running, you cannot access the Setup or Reports panels. To monitor the test progress, switch back and forth between the Status panel and the Results panel.

The application displays a report when the tests are complete. While the tests are running, other applications may display windows in the background. The TekScope application takes precedence over other applications, but you can switch to other applications by using the **Alt** + **Tab** key combination. To keep the TekExpress Thunderbolt application on top, select **Keep On Top** from the Thunderbolt Options menu.

See also

About Configuring Tests (see page 24)

About Setting Up Tests (see page 35)

Before You Click Start (see page 41)

Before you click Start

Before you run tests for the first time, do the following:

1. Understand where your test files are stored on the instrument.

   After you install and launch TekExpress Thunderbolt, it creates the following folders on the oscilloscope:

   - `\Program Files (x86)\Tektronix\TekExpress\TekExpress Thunderbolt`

   NOTE. *The Program Files folder for Windows 7 is `Program Files (x86)` for 64-bit. Windows XP uses `Program Files`.*

   - `\My Documents\My TekExpress\Thunderbolt`
   - `\My Documents\My TekExpress\Thunderbolt\Untitled Session`

     Every time you launch TekExpress Thunderbolt, an `Untitled Session` folder is created in the `Thunderbolt` folder. The `Untitled Session` folder is automatically deleted when you exit the Thunderbolt application.

CAUTION. *Do not directly edit or modify any of the session files or folders because this may result in loss of data or corrupted session files. Each session has multiple files associated with it. When you save a session, the application creates a .TekX file, and a folder named for the session that contains associated files, on the oscilloscope X: drive.*

2. Map the shared My TekExpress folder (see page 88) as X: (X drive) on all instruments used in test setup running Microsoft Windows Operating System.

   The `My TekExpress` folder has the shared name format `<domain><user ID>My TekExpress`. Or, if the instrument is not connected to a domain, then the shared name format is `<instrument name><user ID>My TekExpress`. This shared folder is used to save the test session files and is used during any other file transfer operations.

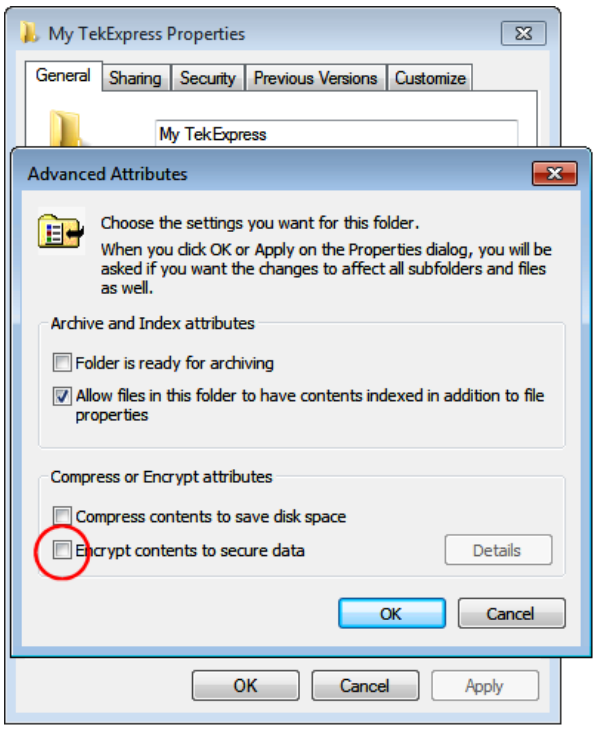
NOTE. *If the X: drive is mapped to any other shared folder, the application will display a warning message asking you to disconnect the X: drive manually.*

3. Make sure that the My TekExpress folder has read and write access and that the contents are not set to be encrypted:

   a. Right-click the folder and select **Properties**.

   b. Select the **General** tab and then click **Advanced**.

   c. In the Advanced Attributes dialog box, make sure that the option **Encrypt contents to secure data** is NOT selected. Example



4. Review the pre-run checklist (see page 43) before you run a test.

See also

View Test-Related Files (see page 29)

Application Directories and Usage (see page 8)

File Name Extensions (see page 9)

Pre-run checklist

Do the following before you click Start to run a test. If this is the first time you are running a test for a setup, refer to the information in Before You Click Start (see page 41).

On the oscilloscope:

- Make sure that all the required instruments are properly warmed up.
- Perform Signal Path Compensation (SPC).
    a. On the oscilloscope main menu, select the **Utilities** menu.
    b. Select **Instrument Calibration** and run the SPC utility..
- Perform deskew on any cables.

In the Thunderbolt application:

1. Verify that the application is able to find the DUT. If it cannot, perform a search for connected instruments (see page 38).
    a. Select **Setup > Test Selection**. Select any test and then click **Configure**.
    b. In the Configuration section, click **Global Settings**.
    c. In the Instruments Detected section, click the drop-down arrow to the right of the listed instruments and make sure that the instrument is listed.
2. Run the Deskew utility (**Options > Deskew**).

See also

Equipment Connection Setup (see page 36)

Test setups overview

A test setup file contains the oscilloscope connection and setting information, general parameters, measurement limits, test selections, and other settings for the current application configuration. This information is saved to a file under the setup name. Use a saved test setup to quickly load and run a test without having to do any other setup except ensuring that the correct oscilloscope is connected and accessible by Thunderbolt.

See also

About setting up tests (see page 35)

Save a test setup (see page 45)

Recall a saved test setup (see page 46)

Create a new test setup based on an existing one (see page 46)

Save a test setup

Save a test setup before or after running a test using the parameters you want saved. Create a new test setup from any open setup or from the default setup. When you select the default test setup, all application settings are returned to their defaults.

The following instructions start from the default test setup:

1. Select **Options > Default Test Setup**.
2. Select **Setup** and set required options and parameters in the tabs (DUT, Test Selection, and so on).
3. Select Reports and set your report options (see page 30).
4. Click Start to run the test and verify that it runs correctly and captures the information you want. If it does not, edit the parameters and repeat this step until the test runs to your satisfaction.
5. Select **Options > Save Test Setup**. Enter the file name for the setup file. The application saves the file to X:\Thunderbolt\<*session_name*>.

See also

About setting up tests (see page 35)

About configuring tests (see page 24)

Test setup overview (see page 39)

Recall a saved test setup

To recall a saved test setup:

1. Select **Options > Open Test Setup**.
2. Select the setup from the list and click **Open**.

See also

Test setups overview (see page 45)

Create a new test setup based on an existing one (see page 46)

Create a new test setup based on an existing one

Use this procedure to create a variation on a test setup without having to create the entire setup from the beginning.

1. Select **Options > Open Test Setup**.
2. Select a setup from the list and then click **Open**.
3. Use the Setup and Reports panels to modify the parameters to meet your testing requirements.
4. Select **Options > Save Test Setup As**.
5. Enter a test setup name and click **Save**.

See also

Test setups overview (see page 45)

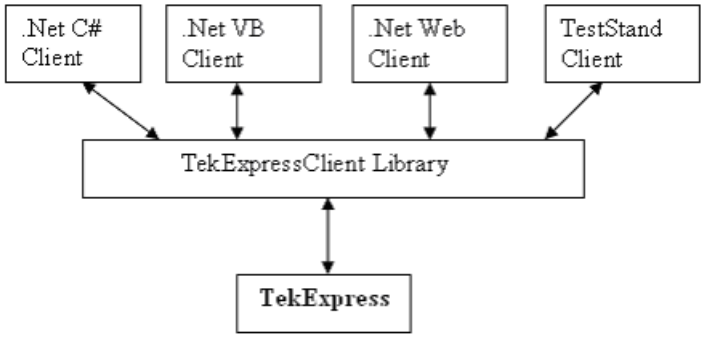
About the programmatic interface

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 the TekExpress application running on a local or a remote computer.

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

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

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



See also

Requirements for Developing TekExpress Client (see page 48)

Remote Proxy Object (see page 49)

Client Proxy Object (see page 49)

Requirements for developing TekExpress client

While developing the TekExpress Client, 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 the `Samples` folder.

References required

- *TekExpressClient.dll* has an internal reference to *IIdlglib.dll* and *IRemoteInterface.dll*.
- *IIdlglib.dll* has a reference to *TekDotNetLib.dll*.
- *IRemoteInterface.dll* provides the interfaces required to perform the remote automations. It is an interface that forms the communication line between the server and the client.
- *IIdlglib.dll* provides the methods to generate and direct the secondary dialog messages at the client-end.

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

Required steps for a client

The following steps are used by the client to programmatically control the server using `TekExpressClient.dll`:

Develop a client UI to access the interfaces exposed through the server. This client loads `TekExpressClient.dll` to access the interfaces. After `TekExpressClient.dll` is loaded, the client UI can call the specific functions to run the operations requested by the client. When the client is up and running, it does the following to run a remote operation:

1. To connect to the server, the client provides the IP address of the PC where the server is running.
2. The client locks 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 the server so that server state cannot be changed by manual operation.

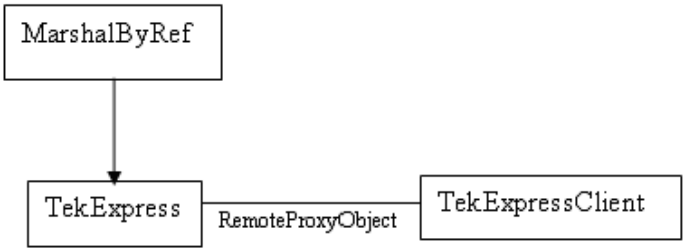
   If any other client tries to access a server that is locked, it will receive a notification that the server is locked by another client.
3. When the client has connected to and locked the server, the client can access any of the programmatic controls needed to run the remote automations.
4. After the client operations finish, the client unlocks the server.

See also

Thunderbolt application commands flow (see page 56)

Remote proxy object

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



The following is an example:

```
RemotingConfiguration.RegisterWellKnownServiceType (typeof (TekExpressRe-
moteInterface), “TekExpress Remote interface”, WellKnownObjectMode.Singleton);
```

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

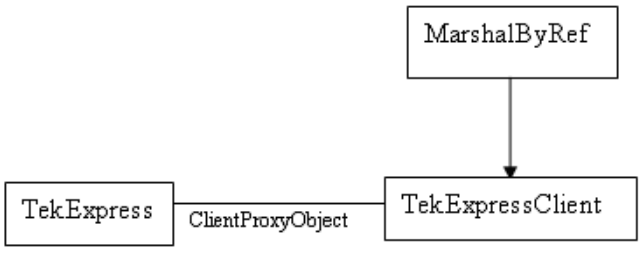
For example,

//Get a reference to the remote object

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

Client proxy object

Client exposes a proxy object to receive certain information.



For example,

//Register the client proxy object

```
WellKnownServiceTypeEntry[] e = RemotingConfiguration.GetRegisteredWell-
KnownServiceTypes();

clientInterface = new ClientInterface();

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

//Expose the client proxy object through marshalling

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

The client proxy object is used for the following:

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

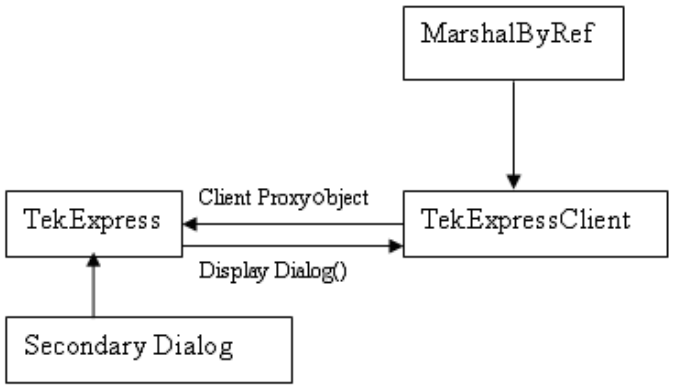
Examples

```
clientObject.clientIntf.DisplayDialog(caption, msg,iconType, btnType);

clientObject.clientIntf.TransferBytes(buffer, read, fileLength);
```

For more information, click the following links:

Secondary Dialog Message Handling



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

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

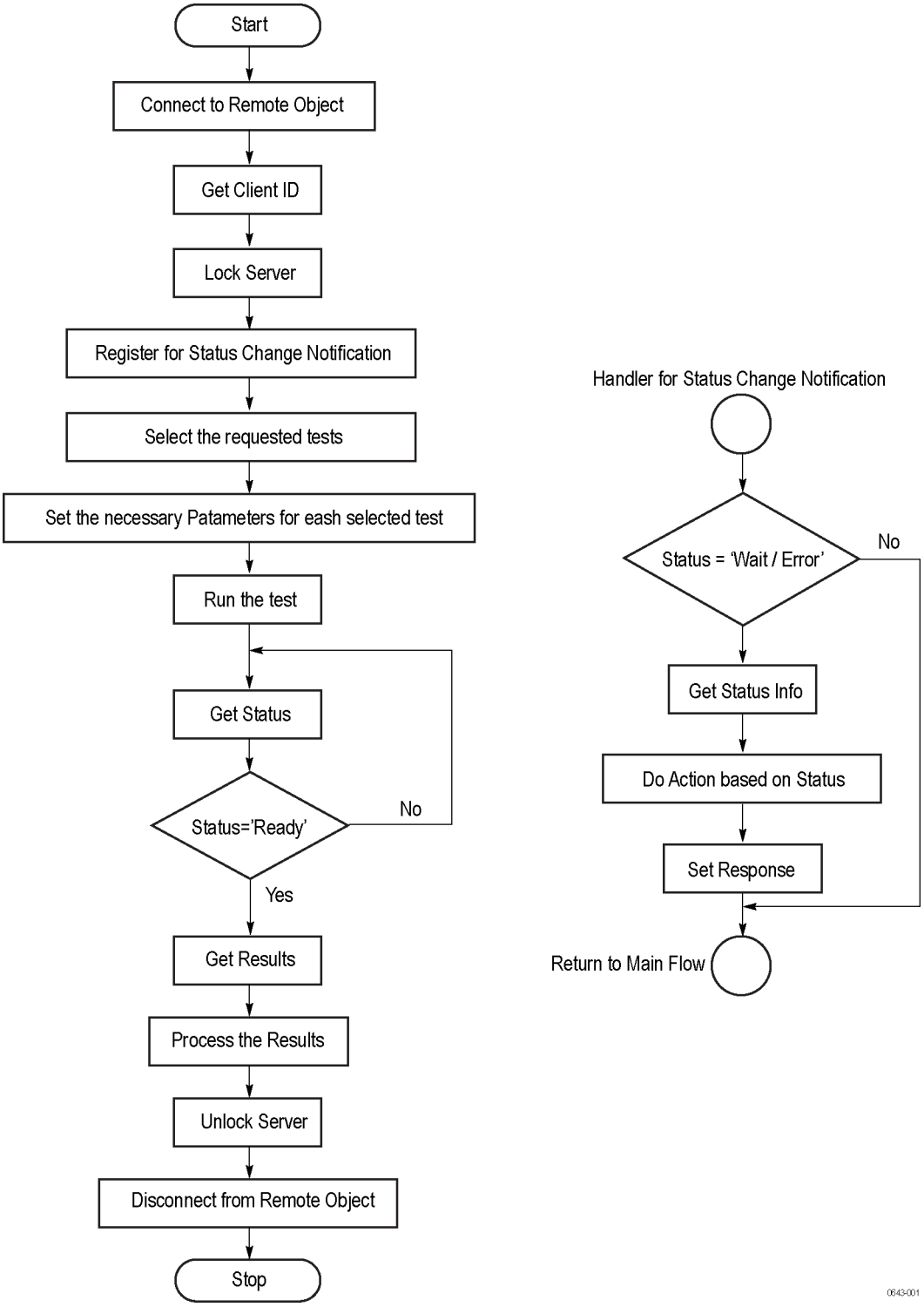
File Transfer Events

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

Client programmatic interface overview

The following is an overview of the client programmatic interface:

Process flowchart


Start
Connect to Remote Object
Get Client ID
Lock Server
Register for Status Change Notification
Select the requested tests
Set the necessary Patameters for eash selected test
Run the test
Get Status
Status='Ready'
No
Yes
Get Results
Process the Results
Unlock Server
Disconnect from Remote Object
Stop
Handler for Status Change Notification
Status = 'Wait / Error'
No
Get Status Info
Do Action based on Status
Set Response
Return to Main Flow
0643-001

1. Connect to a server or remote object using a programmatic interface.
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. *The 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 the 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. For details, see Handler of Status Change Notification (see page 53).

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 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. *Skip this step if you are registered for the status change notification and 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 after completing 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 that 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.

See also

Thunderbolt application commands flow (see page 56)

Program remote access code example (see page 54)

Program remote access code example

This code example shows how to communicate between a remote PC and TekExpress Thunderbolt.

Table 11: Remote access code example

Task	Code
Start the application	
Connect through an IP address.	`m_Client.Connect("localhost") 'True or False` `clientID = m_Client.getClientID`
Lock the server	`m_Client.LockServer(clientID)`
Disable the Popups	`m_Client.SetVerboseMode(clientID, false)`
Set the DUT ID	`m_Client.SetDutId(clientID, "DUT_Name")`

Select a test	For Device:
	`mClient.SelectsingleTest(clientID, "Device", "Source", "SPEC 0.7","PHY 1.12_Rise Time measurement", true)`
	For Host:
	`mClient.SelectsingleTest(clientID, "Host", "Source", “SPEC 0.7”,"PHY 1.12_Rise Time measurement", true)`

Run with set configurations	`m_Client.Run(clientID)`

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, WaitingMsb-BxMessage, WaitingMsbBxResponse)`
	`End Select`
	`Loop Until status = "Ready"`

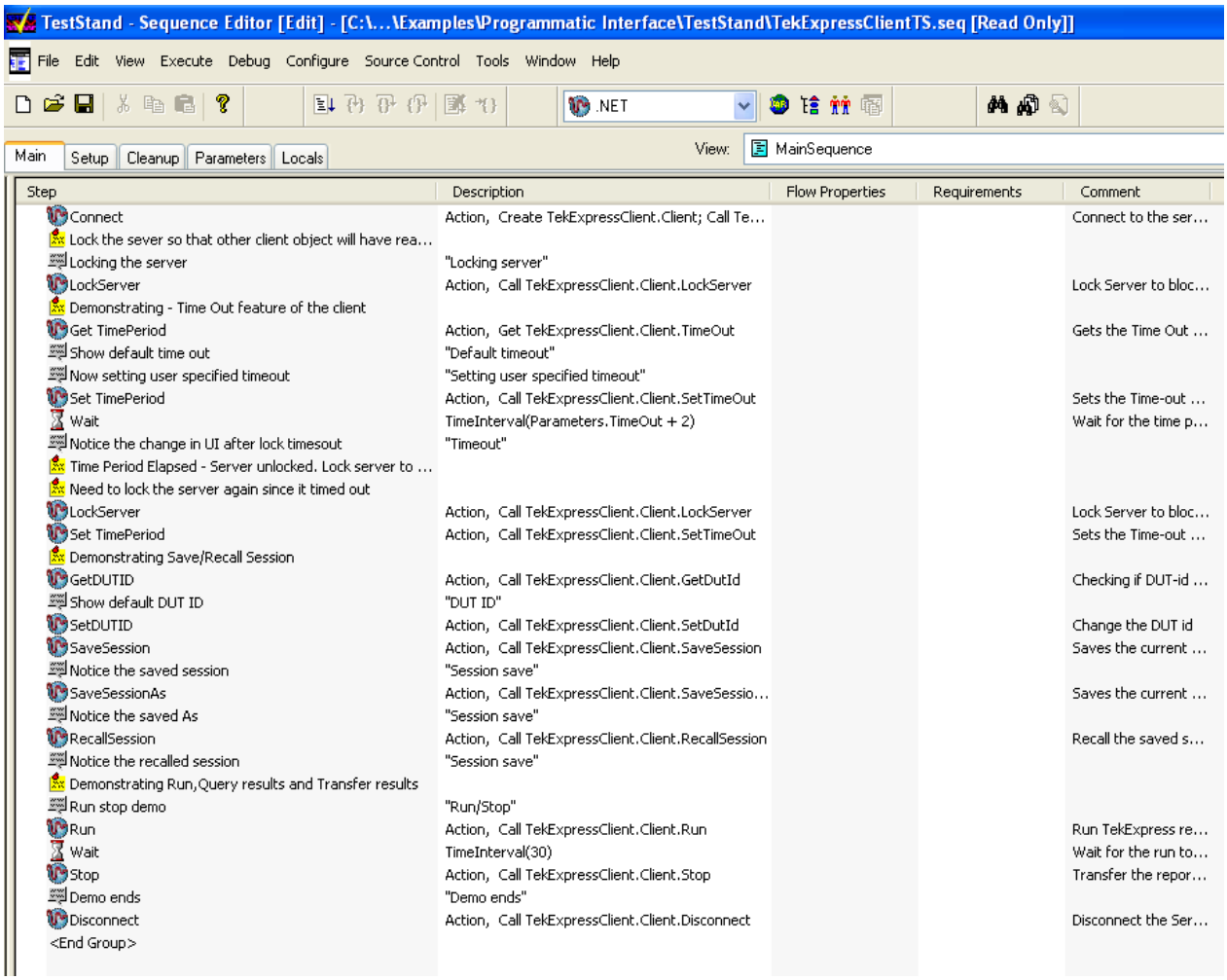
Save results	'Save all results values from folder for current run `m_Client.TransferResult(clientID, logDirname)`
Unlock the server	`m_Client.UnlockServer(clientID)`

Table 11: Remote access code example (cont.)

Task	Code
Disconnect from server	`m_Client.Disconnect()`
Exit the application	

NI TestStand client example

The following is an example for NI TestStand Client available in the path: `C:\Program Files\Tektronix\TekExpress\TekExpress Thunderbolt\Examples\Programmatic Interface\TestStand`



Thunderbolt application commands flow

Click a client action link to see the associated command name, description, parameters, return value, and an example.

string id

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

Ready: Test configured and ready to start

Running: Test running

Paused: Test paused

Wait: A popup that needs your inputs

Error: An error is occurred

string dutName

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

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.

string ipAddress

Name	Type	Direction	Description
ipAddress	string	IN	The ip address of the server to which the client is trying to connect. 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. *If the dutName parameter is null, the client is prompted to provide a valid DUT ID.*

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

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

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.

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	Selects or deselects a test

int rowNr

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

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

out string[] status

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

string name

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

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

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

string name

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

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

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

string name

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

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

bool isSelected

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

string time

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

The time parameter gives the timeout period, which 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; otherwise, the client is prompted to provide a valid timeout period.

bool_verbose

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

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

string filePath

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

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

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

out string WaitingMsbBxCaption

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

out string WaitingMsbBxMessage

Name	Type	Direction	Description
message	string	OUT	The wait state/error state message sent to you

out string[] WaitingMsbBxButtontexts

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

string WaitingMsbBxResponse

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)

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

Connect through an IP address

Command name	Parameters	Description	Return value	Example

Connect()	string ipAddress (see page 57)
	out string clientID (see page 57)
		This method connects the client to the server. Note (see page 57)
		The client provides the IP address to connect to the server.
		The server provides a unique client identification number when connected to it.
			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 boolean
				returnval = m_Client.Connect(ipaddress,m_clientID)

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

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

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

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

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

Lock the server

Command name	Parameters	Description	Return value	Example

LockSession()	string clientID (see page 61)	This method locks the server. Note (see page 58)
		The client must call this method before running any of the remote automations. The server can be locked by only one client.
			String value that gives the status of the operation after it has been performed
			The return value is "Session Locked..." on success.
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				returnval = m_Client.LockServer(clientID)

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

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

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

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

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

Disable the popups

Use these commands to disable popup messages that require user intervention.

Command name	Parameters	Description	Return value	Example

SetVerboseMode()	string clientID (see page 61)
	bool _verbose (see page 60)
		This method sets the verbose mode to either true or false.
		When the value is set to true, any message boxes that appear during the application will be routed to the client machine that is controlling TekExpress.
		When the value is set to false, all the message boxes are shown on the server machine.
			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 boxes will be shown to client”.
			When Verbose mode is set to false, the return value is “Verbose mode turned off. All dialog boxes 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
				Verbose mode is turned on
				return=m_Client.SetVerbose-Mode(clientID, true)
				Verbose mode is turned off
				returnval=m_Client.SetVer-boseMode(clientID, false)

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

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

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

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

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

Set or get the DUT ID

Command name	Parameters	Description	Return value	Example
SetDutId()	string clientID (see page 61) string dutName (see page 57)	This method changes the DUT ID of the setup. 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	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 (see page 57)
GetDutId()	string clientID (see page 61) string dutId (see page 58)	This method gets the DUT ID of the current setup.	String that gives the status of the operation after it has been performed	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)

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

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

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

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

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

Set the configuration parameters for a suite or measurement

Command name	Parameters	Description	Return value	Example
SetGeneralParameter	string clientID (see page 61) string device (see page 58) string suite (see page 58) string test (see page 58) string parameterString (see page 58)	This method sets the number of video lanes for the selected measurement. ***NOTE.*** *Use this command to 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 channel example (see page 66)
SetAnalyzeParameter()	string clientID (see page 61) string device (see page 58) string suite (see page 58) string test (see page 58) string parameterString (see page 58)	This method sets the configuration parameters in the Analyze panel of the Configuration Panel dialog box for a suite or measurement.	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

SetAcquireParameter()	string clientID (see page 61)
	string device (see page 58)
	string suite (see page 58)
	string test (see page 58)
	string parameterString (see page 58)
		This method sets the configuration parameters in the Acquire panel of the Configuration Panel dialog box for a suite or measurement.	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
				returnVal = remoteObject.SetAcquireParameter(id, device, suite, test, parameterString)
				if ((OP_STATUS)
				returnVal != OP_STATUS.SUCCESS)
				return CommandFailed(returnVal)

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

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

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

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

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

Select channel example

returnval=mClient.SetGeneralParameter(clientID, "Device", "Source", " PHY 1.12_Rise Time measurement ", "Lane0 Connected to:Lane0+: Single Ended$CH1")

Query the configuration parameters for a suite or measurement

Command name	Parameters	Description	Return value	Example
GetGeneralParameter()	string clientID (see page 61) string device (see page 58) string suite (see page 58) string test (see page 58) string parameterString (see page 58)	This method gets the general configuration parameters for a suite or measurement.	The return value is the general configuration parameter for a specified suite or measurement that is set.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string Query channel example (see page 68)
GetAnalyzeParameter()	string clientID (see page 61) string device (see page 58) string suite (see page 58) string test (see page 58) string parameterString (see page 58)	This method gets the configuration parameters set in the Analyze panel of the Configuration Panel dialog box for a specified suite or measurement.	The return value is the configuration parameter set in the Analyze panel of the Configuration Panel dialog box for a specified 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 (see page 61) string device (see page 58) string suite (see page 58) string test (see page 58) string parameterString (see page 58)	This method gets the configuration parameters set in the Acquire panel for a specified suite or measurement.	The return value is the configuration parameter set in the Acquire panel for a specified suite or measurement.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string

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

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

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

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

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

Query channel example

returnval = mClient.SetGeneralParameter(clientID, "Device", "Source", " PHY 1.12_Rise Time measurement ", "Lane0 Connected to:Lane0+: Single Ended$CH1")

Select a test

Command name	Parameters	Description	Return value	Example

SelectTest()	string clientID (see page 61)
	string device (see page 58)
	string suite (see page 58)
	string test (see page 58)
	bool isSelected (see page 59)
		This method selects or deselects a specified test.
		If this Setting parameter is set to true, you can select a measurement.
		If this Setting parameter is set 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
				Select test example (see page 68)

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

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

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

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

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

Select test example

To select measurement test PHY 1.12_Rise Time measurement:

returnval = mClient.SelectsingleTest(clientID, "Device", "Source", "SPEC 0.7"," PHY 1.12_Rise Time measurement", true)

Select a suite

Command name	Parameters	Description	Return value	Example

SelectSuite()	string clientID (see page 61)
	string device (see page 58)
	string suite (see page 58)
	bool isSelected (see page 59)
		This method selects or deselects a specified suite.
		When this parameter is set to true, you can select a suite.
		When this parameter is set 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
				Select Suite (Default):
				returnval=mClient.SelectSuitet(clientID, "Device","Source", true)

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

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

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

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

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

Select a channel

Command name	Parameters	Description	Return value	Example
SetGeneralParameter()	string clientID (see page 61) string device (see page 58) string suite (see page 58) string test (see page 58) string parameterString (see page 58)	This method sets the parameters that are not specific to any one test. ***NOTE.*** *Using this command we can select a lane, channel, or source type.*	String that gives the status of the operation after it has been performed The return value is "" (an empty String) on success.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string Set channel example (see page 71)
SetAnalyzeParameter()	string clientID (see page 61) string device (see page 58) string suite (see page 58) string test (see page 58) string parameterString (see page 58)	This method sets the configuration parameters in the Analyze panel of the Configuration Panel dialog box for a specified suite or measurement.	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 (see page 61)
	string device (see page 58)
	string suite (see page 58)
	string test (see page 58)
	string parameterString (see page 58)
		This method sets the configuration parameters in the Acquire panel of the Configuration Panel dialog box for a specified suite or measurement.	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
				returnVal = remoteObject.SetAcquireParameter(id, device, suite, test, parameterString)
				if ((OP_STATUS)
				returnVal != OP_STATUS.SUCCESS) return CommandFailed(returnVal)

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

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

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

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

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

Set channel example

returnval = mClient.SetGeneralParameter(clientID, "Device", "Source", " PHY 1.12_Rise Time measurement", "Lane0 Connected to:Lane0+: Single Ended$CH1")

Configure the selected measurement

Command name	Parameters	Description	Return value	Example
SetAnalyzeParameter()	string clientID (see page 61) string device (see page 58) string suite (see page 58) string test (see page 58) string parameterString (see page 58)	This method sets the Analyze parameters (Configuration parameters) for a specified 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

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

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

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

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

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

Run with set configurations or stop the run operation

Command name	Parameters	Description	Return value	Example
Run()	string clientID (see page 61)	Runs the selected tests Note (see page 59) After the server is set up and configured, run it remotely using this function.	String that gives the status of the operation after it has been performed. The return value is "Run started..." on success.	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)
Stop()	string clientID (see page 61)	Stops the currently running tests Note (see page 60)	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)

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

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

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

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

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

Handle error codes

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

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

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

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

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

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

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

Get or set the timeout value

Command name	Parameters	Description	Return value	Example
GetTimeOut()	string clientID (see page 61)	Returns the current timeout period set by the client	String that gives the status of the operation after it has been performed The default return value is 1800000.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.GetTime-Out()
SetTimeOut()	string clientID (see page 61) string time (see page 60)	Sets a timeout period specified by the client. After this timeout period expires, the server is unlocked automatically.	String that gives the status of the operation after it has been performed On success the return value is “TimeOut Period Changed”.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.SetTime-Out(clientID, desiredTimeOut)

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

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

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

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

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

Wait for the test to complete

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

Command name	Parameters	Description	Return value	Example
ApplicationStatus()	string clientID (see page 61)	This method gets the status of the server application. The states are Ready, Running, Paused, Wait, or Error. (see page 56)	String value that gives the status of the server application	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.ApplicationStatus(clientID)
QueryStatus()	string clientID (see page 61) out string[] status (see page 59)	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...".	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string Query status example (see page 77)

Command name	Parameters	Description	Return value	Example

GetCurrentState-Info()
NOTE. *This command is used when the application is running and is in the wait or error state.*
	string clientID (see page 61)
	out string Wait-ingMsbBx-Caption (see page 60)
	out string Wait-ingMsbBxMes-sage (see page 61)
	out string[] WaitingMsb-BxButtontexts (see page 61)
		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 populates the Out parameters that are passed when invoking this function.
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL
				mClient.GetCurrentState-Info(clientID, WaitingMsbBx-Caption, WaitingMsbBxMes-sage, WaitingMsbBxButton-texts)

SendResponse()
NOTE. *This command is used when the application is running and is in the wait or error state.*
	string clientID (see page 61)
	out string Wait-ingMsbBx-Caption (see page 60)
	out string Wait-ingMsbBxMes-sage (see page 61)
	string Wait-ingMsbBxRe-sponse (see page 61)
		After receiving the additional information using the method GetCurrentStateInfo(), the client can decide which response to send and then send the response to the application using this function. The response should be one of the strings that was received earlier 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
				mClient.SendResponse(cli-entID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxResponse)

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

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

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

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

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

Query status example

```
returnVal=m_Client.QueryStatus(clientID, out statusMessages)
if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)
    return “Status updated...”
else
    return CommandFailed(returnVal)
```

After the test is complete

Command name	Parameters	Description	Return value	Example

GetPassFailStatus()	string clientID (see page 61)
	string device (see page 58)
	string suite (see page 58)
	string test (see page 58)
		This method gets the pass or fail status of the measurement after test completion.
		NOTE. *Execute this command after completing the measurement.*
			String that gives the status of the operation after it has been performed
			Returns the pass or fail status in the form of a string
				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, "PHY 1.12_Rise Time measurement") //Pass or Fail

GetResultsValue()	string clientID (see page 61) string device (see page 58) string suite (see page 58) string test (see page 58) string parameterString (see page 58)	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 srting returnval=m_Client.GetResultsValue(clientID,"M-PHY","Receiver", "PHY 1.12_Rise Time measurement", "Margin")
GetResultsValueForSubMeasurements()	string clientID (see page 61) string device (see page 58) string suite (see page 58) string test (see page 58) string parameterString (see page 58) int rowNr (see page 58)	This method gets the result values for individual submeasurements 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 Get results for a submeasurement example (see page 79)

Command name	Parameters	Description	Return value	Example
GetReportParameter()	string clientID (see page 61) string device (see page 58) string suite (see page 58) string test (see page 58) string parameterString (see page 79)	This method gets the general report details such as oscilloscope model, TekExpress version, and Thunderbolt version.	The return value is the oscilloscope model, TekExpress version, and Thunderbolt version.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string **Oscilloscope Model** returnval=m_Client.GetReportParameter(clientID,"Scope Model") **TekExpress Version** returnval=m_Client.GetReportParameter(clientID,"TekExpress Version") **Thunderbolt Version** returnval=m_Client.GetReportParameter(clientID,"Application Version")
TransferReport()	string clientID (see page 61) string filePath (see page 60)	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")
TransferImages()	string clientID (see page 61) string filePath (see page 60)	This method transfers all the images (screen shots) from the specified client and folder for the current run (for a suite or measurement). ***NOTE.*** *Every time you click Start, a folder is created in the X: drive. Transfer the waveforms before clicking Start.*	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")

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

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

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

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

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

string parameterString

Name	Type	Direction	Description
parameterString	string	IN	Specifies the oscilloscope model, TekExpress version, and Thunderbolt version

Get results for a submeasurement example

This example returns the specified submeasurement results for test PHY 1.12_Rise Time measurement.

returnval=m_Client.GetResultsValueForSubMeasurements(clientID,"Device", "Source", "PHY 1.12_Rise Time measurement", "Margin",0)

returnval=m_Client.GetResultsValueForSubMeasurements(clientID,"Device", "Source", "PHY 1.12_Rise Time measurement", "Margin",1)

Save, recall, or query a saved session

Command name	Parameters	Description	Return value	Example

CheckSession-Saved()	string clientID (see page 61)
	out bool saved (see page 57)
		This method checks whether 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.Check-SessionSaved(m_clientID, out savedStatus)

RecallSession()	string clientID (see page 61)
	string name (see page 59)
		Recalls a saved session. The client provides the session name.	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.RecallSes-sion(clientID, savedSession-Name)

SaveSession()	string clientID (see page 61)
	string name (see page 59)
		Saves the current session. The client provides the session name.	String that gives the status of the operation after it has been performed
			The return value is "Session Sav-ed..."/"Failed..."
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				returnval=m_Client.SaveSes-sion(clientID, desiredSession-Name)

SaveSessionAs()	string clientID (see page 61)
	string name (see page 59)
		Saves the current session under a different name every time this method is called. The client provides the session name.	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.SaveSes-sionAs(clientID, desiredSes-sionName)

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

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

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

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

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

Unlock the server

Command name	Parameters	Description	Return value	Example

UnlockSession()	string clientID (see page 61)	This method unlocks the server from the client. The ID of the client to be unlocked must be provided. Note (see page 60)	String that gives the status of the operation after it has been performed
			The return value is "Session Un-Locked..."
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				returnval=m_Client.Unlock-Server(clientID)

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

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

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

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

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

Disconnect from the server

Command name	Parameters	Description	Return value	Example

Disconnect()	string clientID (see page 61)	This method disconnects the client from the server. Note (see page 57)	Integer value that gives the status of the operation after it has been performed
			1 for Success
			–1 for Failure
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				returnval=m_Client.Discon-nect(m_clientID)

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

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

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

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

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

PHY 1.1 Unit Interval (min) measurement

Verifies that the unit interval is within the conformation limits.

This measurement stores the waveform files and calls the DPOJET period measurement to provide results.

PHY 1.2 SSC Spread deviation

Verifies that the max SSC Spread is less that 0.53% or less that 5300 ppm.

This measurement stores the waveform files and calls the DPOJET “SSC Freq Dev” measurement to provide results.

PHY 1.3 SSC Rate measurement

Measures and displays the profile of the spread spectrum modulation frequency.

This measurement stores the waveform files and calls the DPOJET “SSC Mod Rate” measurement to provide results.

PHY 1.4 SSC Phase deviation

Measures the maximum phase deviation from the nominal frequency.

This measurement stores the waveform files and calls the DPOJET “TIE” measurement with 5Mhz filter and reports the Max – Min (peak to peak TIE) as the result.

Phy 1.5 SSC Slew rate

Measures the Slew rate of the device under test.

This measurement calls a Slew Rate algorithm to calculate the results.

PHY 1.6 Total Jitter

Verifies that the total jitter is within the conformation limits.

Measurements are made on the differential signals on Lane0 and Lane1 of the Thunderbolt link. This measurement can only be performed if the clock is recovered properly. For clock recovery, PLL Type II clock is used with 5 MHz JTF BW and 580m Damping. The Jitter decomposition method used is BUJ+Spectral.

PHY 1.7 Total Jitter-2

Verifies that the total jitter is within the conformation limits.

Measurements are made on the differential signals on Lane0 and Lane1 of the Thunderbolt link. This measurement can only be performed if the clock is recovered properly. For clock recovery, PLL Type II clock is used with 5 MHz JTF Bandwidth and 1.25 Damping. The Jitter decomposition method used is BUJ+Spectral.

PHY 1.8 Total Jitter-LF

Verifies that the total jitter is within the conformation limits.

Measurements are made on the differential signals on Lane0 and Lane1 of the Thunderbolt link. This measurement can only be performed if the clock is recovered properly. For clock recovery, PLL Type II clock is used with 5 MHz JTF and 580m Damping. The TJ is filtered through a 1st order 11MHz filter to calculate only low frequency jitter. The Jitter decomposition method used is BUJ+Spectral.

PHY 1.9 Eye Width measurement

Verifies that the eye width is within the conformation limits.

Measurements are made on the differential signals on Lane0 and Lane1 of the Thunderbolt link. This measurement can only be performed if the clock is recovered properly.

PHY 1.10 Eye Height measurement

Verifies that the Eye Height is within the conformation limits.

Measurements are made on the differential signals on Lane0 and Lane1 of the Thunderbolt link. This measurement can only be performed if the clock is recovered properly.

PHY 1.11 Mask Hits

This measurement ensures that the mask is not hit by the recovered eye.

This test also ensures that the maximum single side voltage is not more that 600 mV.

PHY 1.12 Rise Time measurement

Verifies that the transition (rise) time is within the conformation limits. Rise time is the time taken by the DUT output to rise from 10% to 90% of the high voltage during transition. Transition happens whenever a bit change happens (i.e. from 0 to 1).

Measurements are made on differential signals on Lane0 and Lane1 of the Thunderbolt link.

PHY 1.13 Fall Time measurement

Verifies that the transition (fall) time is within the conformation limits. Fall time is the time taken by the DUT output to fall from 90% to 10% of the high voltage during transition. Transition happens whenever a bit change happens (i.e. from 1 to 0).

Measurements are made on differential signals on Lane0 and Lane1 of the Thunderbolt link.

PHY 1.14 AC common mode voltage Peak to Peak

Verifies that the AC common mode noise (Pk-Pk) of the differential data line of a Thunderbolt interface is within the conformation limits.

Signals are captured as a single ended waveform. Five waveforms are captured with 20M of record length, totaling 100M RL. The maximum AC Common Mode voltage will be reported from the five waveforms.

PHY 1.15 Data Dependent Jitter

Verifies that the Data Dependent Jitter is within the conformation limits.

Measurements are made on the differential signals on Lane0 and Lane1 of the Thunderbolt link. This measurement can only be performed if the clock is recovered properly. For clock recovery, PLL Type II clock is used with 5 MHz JTF and 580m Damping.. The Jitter decomposition method used is BUJ+Spectral.

PHY 1.16 UJ_RJ measurement

Verifies that the Uncorrelated Jitter is within the confirmation limit. It calculates the jitter as TJ-DDJ after removing ISI effects.

This measurement can only be performed if the clock is recovered properly. For clock recovery, PLL Type II clock is used with 5 MHz JTF and 580m Damping. The Jitter decomposition method used is BUJ+Spectral.

PHY 1.17 Intra Pair Skew test

Measures the skew (or time delay) between single ended lines of the data lanes in the Thunderbolt interface. This measurement is to provide information to show if the Thunderbolt lane is providing data in synchronization with each other.

Measurements are made on differential signals on Lane0 and Lane1 of the Thunderbolt link.

Map the My TekExpress folder

In the case where you operate the TekExpress application on one oscilloscope, but acquire data from another (remotely-accessed) oscilloscope, you need to share and map the My TekExpress folder on the remote instrument with the Thunderbolt application.

To map the My TekExpress folder on a remote instrument:

1. Open Windows Explorer.
2. From the Windows Explorer menu, click **Computer**.
3. In the menu bar, click **Map network drive**.
4. Select the Drive letter as **X:** (if there is any previous connection on X:, disconnect it first through **Tools > Disconnect Network drive** menu of Windows Explorer. Windows 7 users: if you do not see the Tools menu, press the **Alt** key).
5. In the Folder field, enter the remote My TekExpress folder path (for example, \\192.158.97.65\ My TekExpress).
6. Click **Finish**.

To determine the IP address of the instrument where the `My TekExpress` folder exists, do the following:

1. On the instrument where the `My TekExpress` folder exists, click **Start** and select **Run**.
2. Type **cmd** and press **Enter**.
3. At the command prompt, type **ipconfig** and press **Enter**.

Measurement parameter descriptions

View or change measurement parameters in the Configuration tab of the Setup panel. Measurement parameters are displayed for the test selected in the tree view section. Not all of the parameters listed apply to all tests, and some are only available when running tests in User Defined Mode. You cannot change parameters if you selected Compliance Mode.

Table 12: Measurement parameters

Parameter type	Parameter	Description	Applies to test
Acquire	Pattern	The type of pattern used to acquire data	All tests
	Include this Pattern	When selected, the pattern displayed in the Pattern field is acquired for the test selected. If this option is not selected, the pattern will not be acquired for the test. In the Acquisitions table, the pattern will be marked "Excluded" in the Acquisition column of the Acquisitions table (if the Show Acquire Parameters check box is selected).	
	Data Rates	The data rate is for the DUT (Gbps) and is set to 10.3125 Gbps. The data rate is automatically selected and cannot be deselected or changed.	
	Device Control Script	TenLira scripts are used to control the device and host DUTs. The TenLira software must be installed so that the Thunderbolt application can access these scripts when running the tests in the User Defined Mode. For details, see TenLira software installation (see page 5).	
Scope Settings	Horizontal Scale	Horizontal scale used to capture the signals	
	Resolution	Resolution used on the measurement	
	Sample Rate	How often the digital oscilloscope takes a snapshot or sample of the signal	
	Edge Trigger Slope	Select the slope used for a trigger event.	
	Edge Trigger Level	Signal level that the trigger source must pass for a trigger event to occur	
BIT Config	BIT Type	■ All Bits: Eye analysis includes both transition and non transition bits ■ Transition: Eye analysis is only on transition bits ■ Non transition: Eye analysis is only on non transition bits	PHY 1.10 PHY 1.11
Clock Recovery	Method	Clock recovery method: ■ Constant Clock-Mean ■ PLL-Custom BW (Phase Locked Loop Custom Bandwidth)	PHY 1.4 PHY 1.5 PHY 1.9 PHY 1.10 PHY 1.11 PHY 1.12 PHY 1.13
	PLL Model Type	Selects between Type 1 and Type 2 phase-locked loop	
	Damping	Specifies the damping ratio of the PLL	
	Loop BW	Sets the bandwidth of the clock recovery PLL	
RjDj	RjDj Pattern Type	■ Set the window length ■ Set the pattern length	PHY 1.6 PHY 1.7 PHY 1.8
	Target BER	■ 1E-12: ■ 1E-13:	PHY 1.15

Parameter type	Parameter	Description	Applies to test
Ref Levels	Ref Levels	■ Absolute: Use to manually set the reference levels. ■ Percentage: Use to set the reference levels as a percentage.	PHY 1.5 PHY 1.6 PHY 1.7 PHY 1.8 PHY 1.9 PHY 1.10 PHY 1.11 PHY 1.12 PHY 1.13 PHY 1.15 PHY 1.16
	Autoset Basetop Method	Method used for calculating the Base and Top of the waveform. (Eye Histogram, Auto) ■ Min-Max: Uses the minimum and maximum values in the waveform to determine the base and top amplitude. Useful for waveforms with low noise and that are free from excessive overshoot ■ Full Histogram: Uses histogram to determine the base and top amplitude. Creates a histogram of the amplitudes of the entire waveform ■ Eye Histogram: Uses histogram to determine the base and top amplitude. Creates a histogram of the amplitudes in the center of each bit (unit interval) while ignoring the waveform during bit transitions ■ Auto: Automatically determines the best Base Top method to use	
	High Level	The high voltage level having maximum population	
	Mid Level	A reference voltage level that defines when the waveform state transition occurs at a given threshold	
	Low Level	The low voltage level having minimum population	
	Hysteresis	Used to prevent small amounts of noise in a waveform from producing multiple threshold crossings. Use when the rising and falling thresholds for a given reference voltage level are set to the same value	
Edges	From Edge	■ Rise: ■ Fall: ■ Both:	PHY 1.1 PHY 1.2 PHY 1.3 PHY 1.17
	To Edge	■ Same As: ■ Opposite As:	

Parameter type	Parameter	Description	Applies to test
Filters	High pass filter (F1) spec	■ None: ■ First: ■ Second: ■ Third:	PHY 1.1 PHY 1.2 PHY 1.3 PHY 1.5 PHY 1.8 PHY 1.12 PHY 1.13 PHY 1.17
	High pass filter	Filter to remove low frequency components	
	Low pass filter (F1) spec	■ None: ■ First: ■ Second: ■ Third:	
	Low pass filter	Filter to remove high frequency noise from the output	
	Filter ramp time	Time taken by the filter to smooth and reach a steady state	
	Filter blanking time	Time for which the output is neglected for measurement	
Masks	Mask file path	Specifies the mask file to use for the different data rates	PHY 1.11
Clock and Signal	Clock recovery method	■ Constant Clock-Mean ■ PLL-Custom BW (Phase Locked Loop Custom Bandwidth)	PHY 1.6 PHY 1.7 PHY 1.8
	PLL Model Type	Selects between Type 1 and Type 2 phase-locked loop	
	Damping	Specifies the damping ratio of the PLL	
	Loop BW	Sets the bandwidth of the clock recovery PLL	
	Jitter Breakdown	■ Spectral + BUJ ■ Spectral	
	Signal Type	■ Clock ■ Data ■ Auto	
	Clock Edge	Constant Clock Mean	

Index

O

P

Q

R

S

T

U