

TekExpress®
M-PHY Transmitter Automated Solution
Printable Online Help



TekExpress®
M-PHY Transmitter Automated Solution
Printable Online Help

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

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

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

TekExpress is a registered trademark of Tektronix, Inc.

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

Getting help and support

Related documentation.....	1
Conventions Used in Help.....	1
Technical support.....	2

Getting started

Installing the software	
Install the software.....	3
Activate the license.....	3
View software version and license information.....	4
Application overview	
M-PHY Tx key features.....	5
Supported oscilloscopes and probes.....	5
Minimum system requirements.....	7
Application directories and files.....	8
File name extensions.....	9

Operating basics

Run the application.....	11
Exit the application.....	11
Application controls and menus	
Application controls.....	12
Options menu	
Options menu overview.....	13
Instrument control settings.....	14
Email settings	
Email Settings dialog box overview.....	14
Configure email settings.....	15
Deskew.....	16
Application panels	
Application panel overview.....	18
Setup panel	
Setup panel overview.....	18
DUT tab controls.....	18
Test Selection tab	
Test Selection tab controls.....	20
Configure test parameters.....	21
Limits Editor controls.....	23

Acquisitions tab controls	23
Status panel overview	25
Results panel	
Results panel overview	26
View test results	27
View test-related files	28
Reports panel	
Reports panel overview	28
Select report options	29
View a report	31
Report contents	32
Setting up and configuring tests	
About setting up tests	35
Instrument connection setup	36
View connected instruments	37
Test setup overview	38
Running tests	
About running tests	39
Before you click Start	39
Pre-test checklist	41
Saving and recalling test setups	
Test setups overview	43
Save a test setup	43
Recall a saved test setup	44
Create a new test setup based on an existing one	44
TekExpress Programmatic Interface	
Programmatic interface overview	45
Requirements for developing TekExpress client	46
Remote proxy object	47
Client proxy object	47
Client programmatic interface overview	49
Program remote access code example	52
M-PHY Tx application commands	
M-PHY Tx application command flow	54
Connect through an IP address	59
Lock the server	60
Disable popup messages	61
Set or get the DUT ID	62

Set the configuration parameters for a suite or measurement	63
Query the configuration parameters for a suite or measurement	65
Select a test	66
Select a suite	67
Select a channel	68
Configure the selected measurement	69
Run with set configurations or stop the run operation	70
Handle error codes	71
Get or set the timeout value	72
Wait for the test to complete	73
After the test is complete	75
Save, recall, or query a saved session	78
Unlock the server	79
Disconnect from the server	79

Reference

HS test parameters

HS test user defined mode parameters	81
1.1.1 HS-TX Unit Interval and Frequency Offset test	84
1.1.2 HS-TX Common-Mode AC Power Spectral Magnitude Limit test	84
1.1.3 HS-TX PREPARE Length test	84
1.1.4 HS-TX Common-Mode DC Output Voltage Amplitude test	85
1.1.5 HS-TX Differential DC Output Voltage Amplitude test	85
1.1.6 HS-TX Minimum Differential AC Eye Opening test	85
1.1.7 HS-TX Maximum Differential AC Eye Opening test	85
1.1.8 HS-TX 20/80% Rise and Fall Times test	85
1.1.10 HS-TX Slew Rate test	86
1.1.11 HS-TX Slew Rate Monotonicity test	86
1.1.12 HS-TX Slew Rate Resolution test	86
1.1.13 HS-TX Intra-Lane Output Skew test	86
1.1.14 HS-TX Transmitter Pulse Width test	86
1.1.15 HS-TX Total Jitter test	87
1.1.16 HS-TX Short-Term Total Jitter test	87
1.1.17 HS-TX Deterministic Jitter test	87
1.1.18 HS-TX Short-Term Deterministic Jitter test	87
Global HS test parameters list	88

PWM test parameters

PWM tests user defined mode parameters	89
1.2.1 PWM-TX Transmit Bit Duration test	90
1.2.2 PWM-TX Transmit Ratio test	90
1.2.3 PWM-TX PREPARE Length test	90
1.2.4 PWM-TX Common-Mode DC Output Voltage Amplitude test	90

Table of Contents

1.2.5 PWM-TX Differential DC Output Voltage Amplitude test.....	90
1.2.8 PWM-TX 20/80% Rise and Fall Times test.....	91
1.2.10 PWM-TX G1 Transmit Bit Duration Tolerance test.....	91
1.2.11 PWM-TX G0 Minor Duration test.....	91
Global PWM test parameters list.....	92
Map the My TekExpress folder	93
Waveform naming conventions (pre-recorded mode)	93

Index

Related documentation

The following information is available as part of the TekExpress M-PHY Tx Automated Solution application.

Table 1: Product documentation

Item	Purpose	Location
Online Help	In-depth operation and UI help	Application Help menu
PDF of the Online Help	Operation and user interface help	Application Help menu Downloadable file from www.tektronix.com (Search keywords 'tekexpress m-phy tx', type = manual; sort by date for latest release)

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 software

The TekExpress M-PHY Tx web installer page provides a link to the software package TekExpressM-PHYTXWebInstaller.exe used to install the application on a supported oscilloscope. For a list of compatible instruments, see [Supported oscilloscopes and probes \(see page 5\)](#).

To download and install the files:

1. Close the TekScope application.
2. Go to the www.tek.com Web site and search for M-PHYTX to locate the installation file. Download the file TekExpressM-PHYTXwebInstaller.exe.
3. If you downloaded the file to a PC, copy the file to a supported Tektronix oscilloscope model.
4. Double-click the executable file to extract the installation files and automatically start the installer program. The software is installed at the following location:
 - Windows XP instruments: C:\Program Files\Tektronix\TekExpress\TekExpress M-PHY Tx
 - Windows 7 instruments: C:\Program Files (x86)\Tektronix\TekExpress\TekExpress M-PHY Tx
- 5.
6. To run the application, open the TekScope application and select **Analyze > TekExpress M-PHY Tx**.

See also

[Minimum system requirements \(see page 7\)](#)

[Supported oscilloscopes and probes \(see page 5\)](#)

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 M-PHY Tx 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

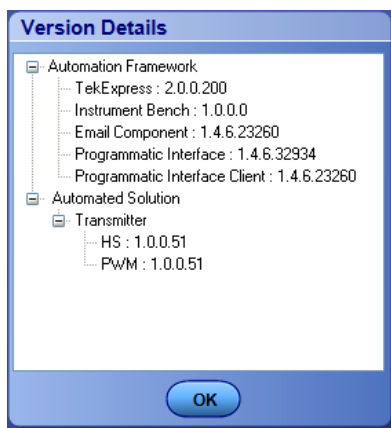
[View software version and license information \(see page 4\)](#)

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. Select **Options > About TekExpress**.
2. Click the View software information 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 M-PHY Tx: M-PHY Transmitter solution.

2. Look in the **Option Installation Key** section for option key information.

See also

[Activate the license \(see page 3\)](#)

[Options menu overview \(see page 13\)](#)

M-PHY Tx key features

Welcome to the TekExpress® M-PHY Transmitter Automated Solution application (Option M-PHYTX). M-PHY Tx provides an automated, simple, and efficient way to test M-PHY Transmitter interfaces and devices consistent to the requirements of the M-PHY Base Specification v1.0 and Conformance Test Suite (CTS) v0.8.

M-PHY Tx 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 M-PHY transmitter testing features include:

- Automated testing reduces the complexity of executing transmitter tests and enables you to test devices faster
- Highly optimized setup performs Power Spectral Density (PSD) tests using oscilloscope-integrated algorithms that do not require an external spectrum analyzer or extra hardware to perform PSD measurements
- Provides 95% automated test coverage of high speed (HS) measurements for gears 0-3
- Provides 75% automated test coverage of PWM measurements for all PWM gears (0-7)
- Seamless debug allows pause on each test in automation, and switch to DPOJET analysis tool for detailed debug
- User-defined mode allows modifying HS and PWM test parameters for comprehensive debug analysis and characterization
- Differential acquisition mode enables Multi-lane M-PHY setup, to connect up to 4 lanes of an M-PHY transmitter to 4 channels on an oscilloscope.
- Configuration for regression allows selecting different gears and sub-gears of HS and PWM signals, large/small amplitudes, and impedance termination/non-termination
- Single printable report for approximately 1000 tests across different combinations, provides pass/fail summary table, along with margin details, optional waveform captures, and eye diagrams
- Supports single-ended and differential acquisition mode

Supported oscilloscopes and probes

The TekExpress M-PHY Tx application runs HS and PWM tests on the following Tektronix oscilloscope and probe configurations:

Table 2: DPO/DSA/MSO oscilloscopes that support M-PHY Tx HS tests

Gear	70604, B, C	70804, B, C	71254, B, C	71604, B, C	72004, B, C
G1A	•	•	•	•	•
G1B	•	•	•	•	•
G2A	–	•	•	•	•
G2B	–	•	•	•	•
G3A	–	–	–	–	•
G3B	–	–	–	–	•

Table 3: DPO/DSA oscilloscopes that support M-PHY Tx HS tests

Gear	72504D	73304D
All (G1A, G1B, G2A, G2B, G3A, G3B)	•	•

Table 4: Probes that support M-PHY Tx HS tests

Gear	P7313 ¹	P7360A	P7380A	P7506	P7508	P7513A	P7516	P7520
G1A	•	•	•	•	•	•	•	•
G1B	•	•	•	•	•	•	•	•
G2A	•	–	–	–	–	•	•	•
G2B	•	–	–	–	–	•	•	•
G3A	–	–	–	–	–	–	–	•
G3B	–	–	–	–	–	–	–	•

¹ Probe P7313 can be a SMA or nonSMA probe based on DUT connection.

Table 5: Oscilloscopes and probes that support M-PHY Tx PWM tests

Gear	Supported DPO/DSA/MSO oscilloscopes	Supported DPO/DSA oscilloscopes	Supported probes
All (G0–G7)	70604, B, C 70804, B, C 71254, B, C 71604, B, C 72004, B and C	72504D 73304D	P7360A P7380A P7380SMA P7313 ¹

¹ Probe P7313 can be a SMA or nonSMA probe based on DUT connection.

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 6: System requirements

Oscilloscope	<ul style="list-style-type: none"> ■ Supported oscilloscopes and probes (see page 5)
Processor	Same as the oscilloscope
Operating System	Same as the oscilloscope: <ul style="list-style-type: none"> ■ 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	DPO/DSA/MSO TekScope v6.1.1.32 or later
Software	<ul style="list-style-type: none"> ■ National Instruments LabVIEW Runtime 9.0.1 or later ■ National Instruments TestStand engine 4.2.1 ■ 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
Probes	<ul style="list-style-type: none"> ■ Supported oscilloscopes and probes (see page 5)
Other Devices	<ul style="list-style-type: none"> ■ Matched pair of SMA cables, two-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.

NOTE. *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 and probes \(see page 5\)](#)

[Instrument connection setup \(see page 36\)](#)

Application directories and files

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

Table 7: Application directories

Directory names	Contains
InstallDir\TekExpress\TekExpress MIPI M-PHY Tx	Path to the root application directory and associated sub-folders. The folders described in this table (ACP, Bin, and so on) are all in this root directory
ACP	Instrument and M-PHY Tx application-specific interface libraries
Bin	Miscellaneous M-PHY Tx application libraries
Compliance Suites	Compliance-specific files
Data Manager	Result management-specific libraries of the M-PHY Tx application
Data Storage	Libraries needed for storing data
Documents	Technical documentation for the M-PHY Tx application
Examples	Various support files
ICP	Instrument and M-PHY Tx application-specific interface libraries
Lib	Utility files specific to the M-PHY Tx application
Report Generator	Excel Active X interface Library for Report Generation
SCP	Instrument and M-PHY Tx application-specific interface libraries
Tools	Instrument and M-PHY Tx application-specific files

See also

[View test-related files \(see page 28\)](#)

[File name extensions \(see page 9\)](#)

File name extensions

The M-PHY Tx 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	Test result report file formats (.mht is the default format).
.xls	
.csv	
.html	

See also

[Select report options \(see page 29\)](#)

[View test-related files \(see page 28\)](#)

[Application directories and files \(see page 8\)](#)

Run the application

To run the M-PHY Tx application, do either of the following:

- Select **Analyze > TekExpress M-PHY Tx** from the TekScope menu.
- Double-click any saved M-PHY 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 M-PHY Tx. 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 and demo the application 10 times. 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





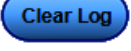


Use the following method to exit the application:

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 8: Application control descriptions

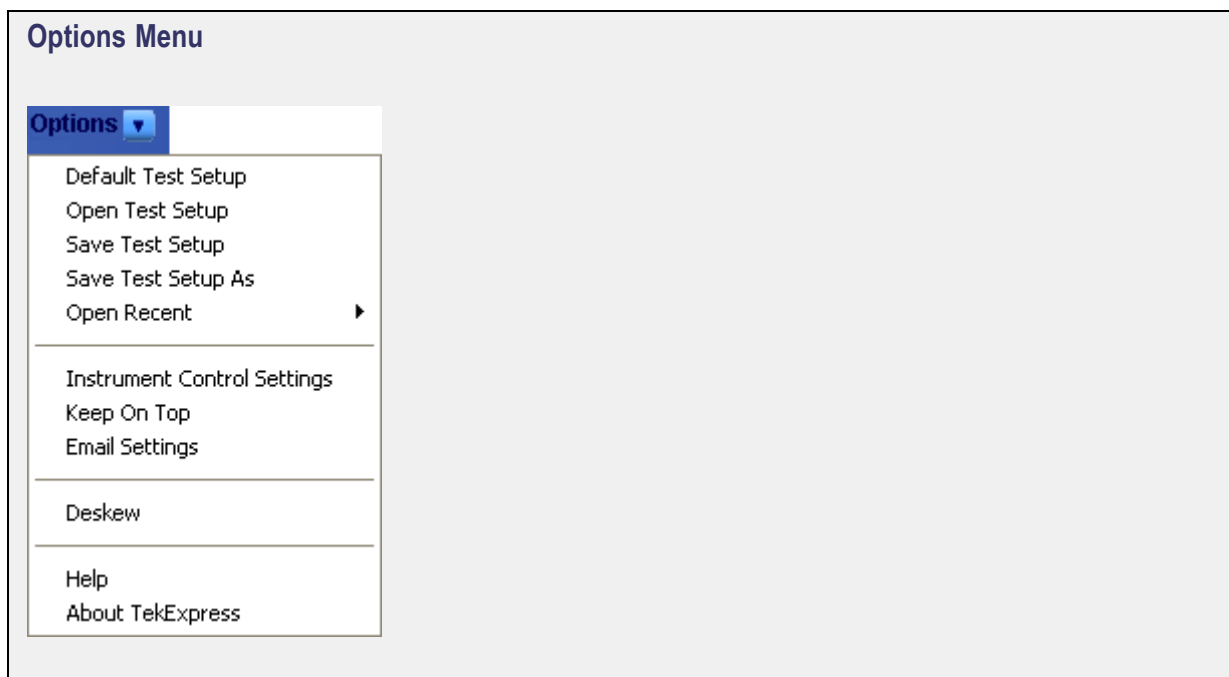
Item	Description
Options menu (see page 13)	Opens the Options menu for access to global controls
Panels (see page 18)	Visual frames with sets of related options
Command buttons	Buttons that initiate an immediate action such as the Start, Stop, Pause, Continue, and Clear command buttons
Start button	<p style="text-align: center;">Start</p>  <p>Starts measurement acquisition and analysis.</p>
Stop button	<p style="text-align: center;">Stop</p>  <p>Stops (aborts) the current test run.</p>
Pause \ Continue button	<p style="text-align: center;">Pause Continue</p>  <p>Use the Pause button to temporarily interrupt the current test run. When a test is paused, the button name changes to Continue.</p>
Clear button	<p style="text-align: center;">Clear</p>  <p>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 26).</p>
Clear Log	 <p>This button is available only on the Status panel (see page 25).</p>
Save	 <p>This button is available only on the Status panel (see page 25).</p>
Application window move icon	 <p>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.</p>

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 M-PHY Tx 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	<ul style="list-style-type: none"> ■ Displays application details such as software name, version number, and copyright ■ Provides access to software version and license information (see page 4) for your M-PHY Tx installation ■ Provides a link to the Tektronix Web site

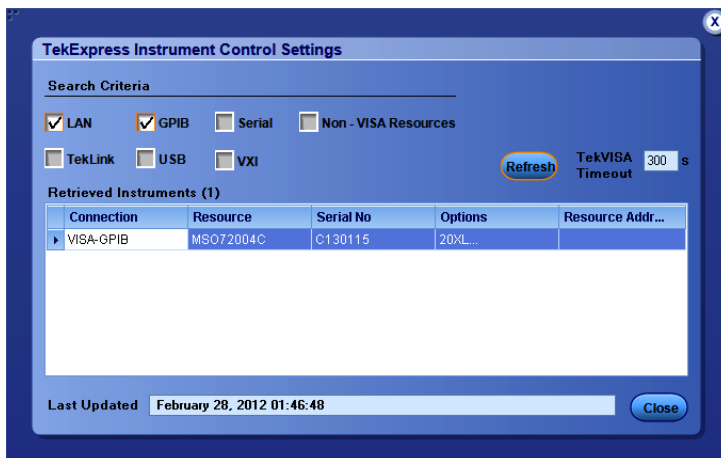


See also

[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 \(see page 21\)](#) for details.

See also

[View connected instruments \(see page 37\)](#)

[Options menu overview \(see page 13\)](#)

Email Settings dialog box overview

Use the Email Settings dialog box to enable M-PHY Tx 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.

You cannot clear these fields after you enter a value. You can change the email address or enable or disable the mailing option.

NOTE. Once user enters the value in these fields, user will not be able to clear the fields. However user can enable or disable the mailing option.

See also

[Configure email settings \(see page 15\)](#)

[Options menu \(see page 13\)](#)

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.

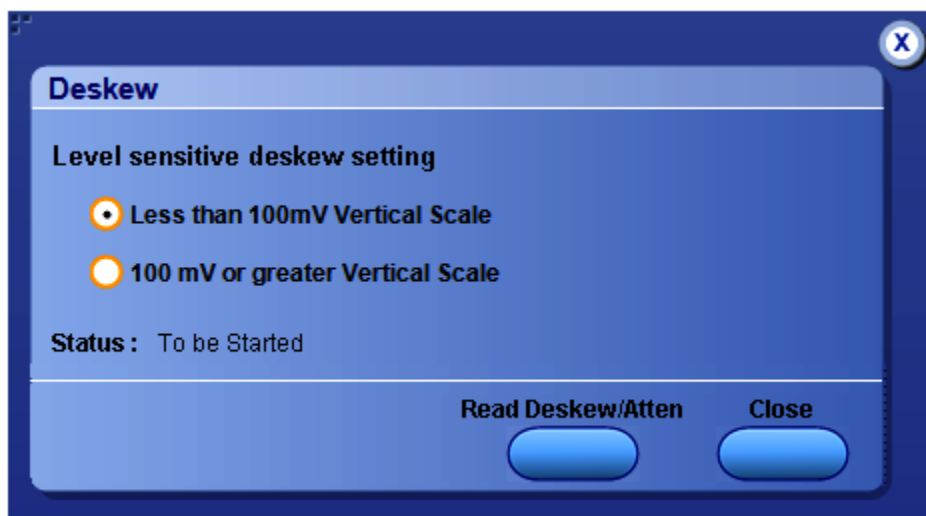
Deskew

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), M-PHYTX 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 M-PHYTX 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. *M-PHYTX 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 M-PHY TX 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 M-PHY Tx\ICP\Deskew-Attenuation.txt
 - **≥100 mV:** C:\Program Files\Tektronix\TekExpress\TekExpress M-PHY Tx\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 M-PHY Tx panels are:

Table 9: Application panels

Panel name	Description
Setup (see page 18)	Set the DUT, test, acquisition, and report parameters.
Status (see page 25)	View the progress and analysis status of the selected tests, and view test logs.
Results (see page 26)	View a summary of test results and select results viewing preferences.
Reports (see page 28)	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\(s\) \(see page 20\).](#)

[Set lane acquisition source \(see page 23\)](#)

[Configure tests \(see page 21\).](#)

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

DUT tab controls

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 notepad 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 \(see page 29\)](#).
4. Select from the following parameters. Settings that do not apply to compliance testing cannot be changed and are grayed out.

Table 10: DUT tab settings

Control or field	Description
Acquire live waveforms	Acquire active signals from the oscilloscope for testing.
Use pre-recorded waveform files	Run an analysis on a saved run session file. Select a file from the Run Session field.
Run Session	Lists saved run sessions (waveform acquisitions) on which you can perform analysis. Use in conjunction with the 'Use pre-recorded waveform files' control. NOTE. Pre-recorded (saved) waveform file names follow a specific naming syntax. To use your own waveform files (generated outside of this application), name the waveform files appropriately. Waveform naming conventions (pre-recorded mode) (see page 93)
View	Determines where to access the test configuration settings: <ul style="list-style-type: none"> ■ Compliance: View configuration settings by clicking Setup > Test Selection > Configure ■ Advanced: Enables the Setup > Configuration tab in which to view configuration settings.
Version	Select the M-PHY Tx testing specification version to use for testing. The latest version is the default.
Signal Type	<ul style="list-style-type: none"> ■ HS: High Speed. ■ PWM: Pulse Width Modulation.
DUT Operation Mode	<ul style="list-style-type: none"> ■ Burst ■ Continuous <p>NOTE. HS tests that are available in Continuous mode are tests 1.1.1, 1.1.2, 1.1.6, 1.1.7, and tests 1.1.13-1.1.18</p> <p>PWM tests that are available in Continuous mode are tests 1.2.1, 1.2.2, 1.2.10 and 1.2.11</p>
Amplitude	<ul style="list-style-type: none"> ■ Small ■ Large <p>You can select both of these settings.</p>
Termination	<ul style="list-style-type: none"> ■ Terminated ■ Non Terminated <p>You can select both of these settings.</p>

Table 10: DUT tab settings (cont.)

Control or field	Description
Probing	<ul style="list-style-type: none"> ■ Single-Ended ■ Differential <p>NOTE. All HS tests are available in Differential mode except for tests 1.1.2, 1.1.4, and 1.1.13.</p> <p>All PWM tests are available in Differential mode except for test 1.2.4.</p> <p>Instrument connection setup (see page 36)</p>
GEAR Selection for HS	<p>Select the GEAR and GEAR variation supported by the DUT. Available parameters are:</p> <p>GEAR 1</p> <ul style="list-style-type: none"> ■ A : 1.248Gbps ■ B : 1.457Gbps ■ A and B <p>GEAR 2</p> <ul style="list-style-type: none"> ■ A : 2.496Gbps ■ B : 2.915Gbps ■ A and B <p>GEAR 3</p> <ul style="list-style-type: none"> ■ A : 4.992Gbps ■ B : 5.830Gbps ■ A and B
GEAR Selection for PWM	Select one or more GEARS (0-7) relevant to your DUT PWM signal testing requirements.

See also

[About setting up tests \(see page 35\)](#)

Test Selection tab controls

Use the Test Selection tab to select the tests to run on the connected DUT and configure individual test parameters. Click **Setup > Test Selection** to open this tab.

Table 11: Test Selection tab settings

Control or field	Description
Deselect All	Deselects (clears) all tests.
Select Required	Selects all test required to pass compliance.

Table 11: Test Selection tab settings (cont.)

Control or field	Description
Select All	Selects all listed tests.
Test selection field	Lists available test and if they are selected to run. Click in the box adjacent to a test to select or unselect a test. Tests listed are affected by DUT tab selections (GEAR, Continuous Mode, Burst Mode, Single-Ended, or Differential).
Test Description field	Shows a description of the selected test.
Schematic	Opens a connection diagram that shows the DUT test setup. Use the diagram to verify the test setup before running the test.
Configure	Opens the configuration settings view. Use this display to view or change global and measurement parameters.

See also

[About setting up tests \(see page 35\)](#)

Configure test parameters

Use the **Configuration** button in the Test Selection tab to view or edit the global and measurement parameters for selected tests. Global parameters set the instrument source and related parameters that are common for all tests. Measurement parameters listed are specific to the tests in the Configuration view.

NOTE. *You can also display a separate measurements Configuration tab in the Setup view by selecting **Setup > DUT > View = Advanced**.*

Table 12: Test configuration settings

Control or field	Description
Compliance Mode	Sets the global and measurement parameters of all measurements to the values required to pass compliance testing. If you are in User Define Mode, selecting compliance mode returns all global settings and measurement parameters to their compliance default values. NOTE. <i>Save test setups made in the User Defined Mode before changing to Compliance Mode, as the application does not automatically save user defined changes or prompt you to save settings before changing modes.</i>
User Defined Mode	Enables editing of the global and measurement parameters for tests.
Test Selection	Returns the application window to the default Test Selection tab listing.
Limits Editor	Opens the Limits Editor window where you can set high and low limits for each test when in User Defined Mode. If you are in Compliance Mode you can only view the default limit values. Limits Editor (see page 23)

Table 12: Test configuration settings (cont.)

Control or field	Description
Global Settings	<p>Click this tab to list the detected instrument(s) and any available global parameters that you can set. The global settings apply to all tests.</p> <p>To select a different instrument, click on the instruments list in the Instruments Detected area and select an instrument from the menu. If you do not see the desired instrument in the list, refresh the list (see page 37). If the instrument is still not listed, verify that the instrument is powered on, has correct network settings, and is connected to the network or instrument.</p>
Measurements	<p>Lists all tests for the selected measurement type (HS or PWM). Click on a measurement to view the available parameters in the tabbed field below the list. The parameters and parameter type tabs shown depend on the selected test.</p> <p>Select User Defined Mode to edit test parameters.</p> <p>Use the scroll bar in the parameters area to scroll through all available parameters.</p> <p>NOTE. <i>Save test setups made in the User Defined Mode before changing to Compliance Mode, as the application does not automatically save user defined changes or prompt you to save settings before changing modes.</i></p>

See also[Configure tests](#)[About running tests \(see page 39\)](#)[Global HS test parameters list \(see page 88\)](#)[Global PWM test parameters list \(see page 92\)](#)

Limits Editor controls

The Limits Editor window lets you set high and low limits for each test when in User Defined Mode. You can view (but not change) the limit settings when in Compliance Mode.

Table 13: Limits Editor controls

Control or field	Description
Test Name	The name of the test.
Details	The limit test(s) that are part of the overall test suite.
Compare String	Sets the comparison logic for the high or low limit test parameter. Click on this field and select the appropriate comparison logic from the menu. There is a separate Compare String field for each low limit and the high limit parameter.
Low Limit/High Limit	The numeric value of the low or high limit parameter. Click in the field and enter a value. The limit units are shown in the Description field for that limit test.
OK	Click OK to close the Limits Editor window and return to the test configuration view. NOTE. Save test setups made in the User Defined Mode before changing to Compliance Mode, as the application does not automatically save user defined changes or prompt you to save settings before changing modes.

Limits Editor caveats

- Press the keyboard Esc button or click the X button (upper right corner of Limits Editor dialog box) to cancel or not save the current edits.
- If you enter the wrong compare string or wrong value in a field, you cannot select or move to another cell until you enter a correct value into the field in question.
- You cannot delete or clear a compare string or value once entered.

Acquisitions tab controls

Use the Acquisitions tab to set the signal source (channel) used to acquire data for the tested lanes.

1. Click **Setup > Acquisitions**.
2. Click the **Lane 0 (+)** and **Lane 0 (-)** fields to select the oscilloscope channels to use for those lanes. Lane 0(+) and Lane 0(-) fields apply to Single-ended mode (DUT Panel > Probing). A single Lane 0 field is available for Differential mode probing.

3. Select **Show Acquire Parameters** to show the acquisition parameters in the test list.
4. Select a [Signal validation \(see page 24\)](#) parameter (Signal validation is valid only for Live acquisitions):
 - 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 **Skip test if signal fails** to skip all tests that depend on an acquired signal that fails signal validation. The comments section in the report file will show the details as "User skipped acquisitions for this pattern".
 - 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.

NOTE. *In Pre-recorded mode, the acquisition table includes a column called "Waveform File Name." Use the fields in this column to browse and select waveform files to load for analysis.*

Signal validation overview

Signal validation checks the following signal parameters on each acquisition for the indicated measurement type:

- **Data Rate** (HS, PWM): If the difference between the measured and expected data rate is less than 2000 ppm, the signal is considered valid for this parameter. The Data Rate check is mandatory for signal validation.
- **MARKER0** (HS): If MARKER0 is present, the signal is considered valid for this parameter. The MARKER0 check is mandatory for signal validation.
- **CRPAT** (HS): If one complete CRPAT (LLI specific) is present, then the signal is considered valid for this parameter. The CRPAT check is not mandatory for signal validation.
- **PWM** (PWM): Pulse Width modulation verified. The PWM check is mandatory for PWM signal validation.

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 14: Test Status tab

Column	Description
Test Name	Name of the test
Acquire Status, Analysis Status	Status of the signal acquisition or test analysis <ul style="list-style-type: none"> ■ To be started ■ In progress ■ Testing ■ Completed ■ Aborted

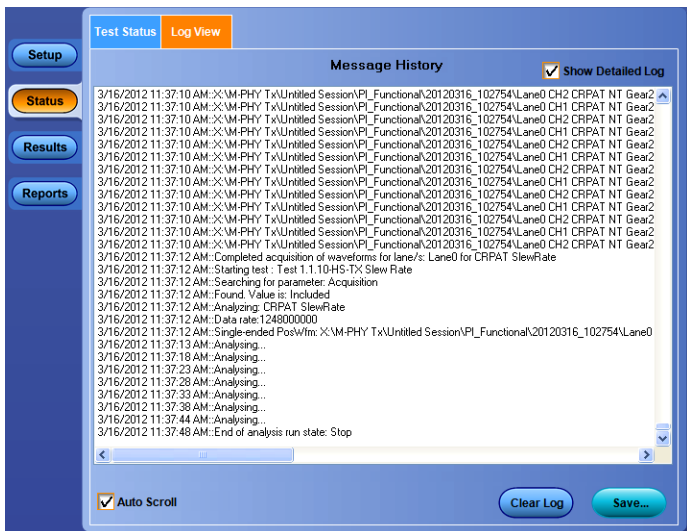
The Log View tab provides a list of the actions executed during the test. Use this information to review or troubleshoot tests.

Table 15: 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 \(see page 27\)](#)



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 \(see page 27\)](#).

The screenshot shows the 'Results' panel with the overall test result 'Fail'. The table below summarizes the test results:

Test Name	Pass/Fail	Lane	Data Rate	Details	Value	Margin
Test 1.1.1-HS-TX Unit Interval and Frequency Offset	Informative	Lane0	Gear 1A	CRPAT RT LA SR2 10usdv 10pspt-SSC PROFILE (dB)	801.282	-N.A.
Test 1.1.2-HS-TX Common-Mode AC Power Spectral Magnitude Limit	Fail	Lane0	Gear 1A	CRPAT RT LA SR1 10usdv 10pspt-PSD Margin (#)	0	0
High Limit	N/A				N/A	
Low Limit	Pass				0	
Test 1.1.2-HS-TX Common-Mode AC Power Spectral Magnitude Limit	Fail	Lane0	Gear 1A	CRPAT RT LA SR2 10usdv 10pspt-PSD Margin (#)	0	0

View test results

When a test finishes, the application switches to the [Results panel \(see page 26\)](#), 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

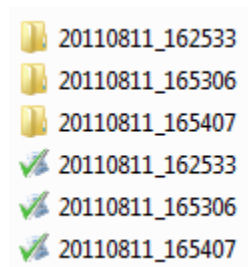
[View a report \(see page 31\)](#)

View test-related files

Files related to tests are stored in the `My TekExpress\M-PHY Tx` folder. In the `M-PHY Tx` 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 M-PHY Tx 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\M-PHY Tx`. When you name and save the session, the files are placed in a folder with the name that you specify.

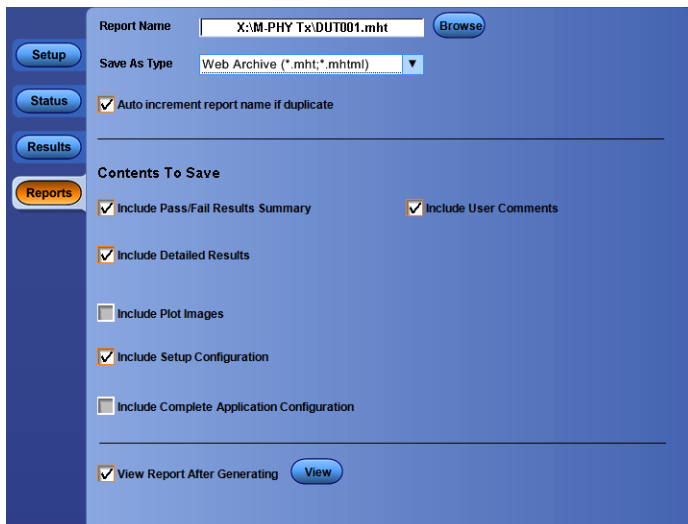
See also

[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 \(see page 29\)](#). For information on viewing reports, see [View a report \(see page 31\)](#).



Select report options

Use the [Reports panel \(see page 28\)](#) 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 16: Report options

Setting	Description
Report Name	<p>Displays the name and location from which to open a report. The default location is at \My TekExpress\ M-PHY Tx \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.</p> <p>Do one of the following:</p> <ul style="list-style-type: none"> ■ 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. <p>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\M-PHY Tx\DUT001_Test_211.mht.</p> <p>NOTE. <i>You cannot change the file location using the Browse button.</i></p> <p>Open an existing report.</p> <p>Click Browse, locate and select the report file and then click View at the bottom of the panel.</p>
Save As Type	<p>Saves a report in the specified file type. Supported file types are: .mht (Web archive), .html (HTML page), .csv (comma-separated values), and .xls (Microsoft Excel spreadsheet).</p> <p>NOTE. <i>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.</i></p>
Auto increment report name if duplicate	<p>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.</p>
Include Pass/Fail Results Summary	<p>Includes 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 31).</p>
Include Detailed Results	<p>Includes parameter limits, execution time, and test-specific comments generated during the test.</p>
Include Plot Images	<p>Includes waveform plots in the report.</p>
Include Setup Configuration	<p>Includes 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.</p>
Include Complete Application Configuration	<p>Includes a table listing general, common, and acquired parameters used in the test. This option is disabled by default.</p>
Include User Comments	<p>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.</p>
View Report After Generating	<p>Automatically opens the report in your Web browser when the test completes. This option is selected by default.</p>

See also

[View a report \(see page 31\)](#)

[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 \(see page 29\)](#).

Report contents

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

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.



TekExpress HS-TX
Report

DUT ID: <u>DUT001</u>	Device Type: <u>Transmitter</u>
Date/Time: <u>March 06, 2012 7:22:29:53</u>	Execution Time: <u>2 Min</u>
CTS Version: <u>CTS v0.9</u>	
Overall Compliance Mode: <u>TRUE</u>	
Overall Test Result: <u>PASS</u>	
Scope Model: <u>MS072004C</u>	Scope FW Version: <u>5.3.4 BUILD 25</u>
Scope Serial Number: <u>C130115</u>	SPC, Factory Calibration: <u>PASS:PASS</u>
Probe Model (CH1): <u>"1X"</u>	TekExpress Version (FW, App): <u>2.0.0.200, 1.0.0.50</u>
Probe Serial Number (CH1): <u>"NA"</u>	DPOJET Version: <u>"3.5.95 Build 113"</u>
Probe Model (CH2): <u>"1X"</u>	
Probe Serial Number (CH2): <u>"NA"</u>	
Probe Model (CH3): <u>NA</u>	
Probe Serial Number (CH3): <u>NA</u>	
Probe Model (CH4): <u>NA</u>	
Probe Serial Number (CH4): <u>NA</u>	

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.

Comments	
----------	--

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	Termination	Gear	Amplitude	Measurement Details	Measured value	Units	Test Result	Margin
Test 1.1.10-HS-TX Slew Rate	Lane0	RT	Gear1A	LA	Maximum Slew Rate	1.904	V/nS	Pass	0.239
					Minimum Slew Rate	1.803	V/nS	Fail	1.255
Test 1.1.1-HS-TX Unit Interval and Frequency Offset	Lane0	RT	Gear1A	LA	SQI (Units/Unit)	11.477	ppm	Pass	1989.52
					SQI (Units/Unit)	801.292	pS	Informative	NA
Test 1.1.3-HS-TX PREPARE Length	Lane0	RT	Gear1A	LA	Test HS PREPARE Length	0.011	UI	Pass	0.089

See also

[View test results \(see page 27\)](#)

[View test-related files \(see page 28\)](#)

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.

All listed tests are required for compliance testing.

See also

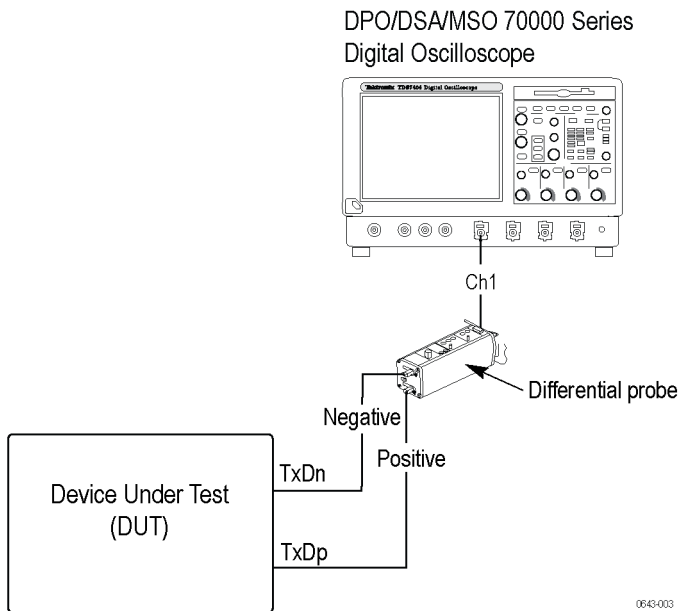
[About test setups \(see page 43\)](#)

[Before you click Start \(see page 39\)](#)

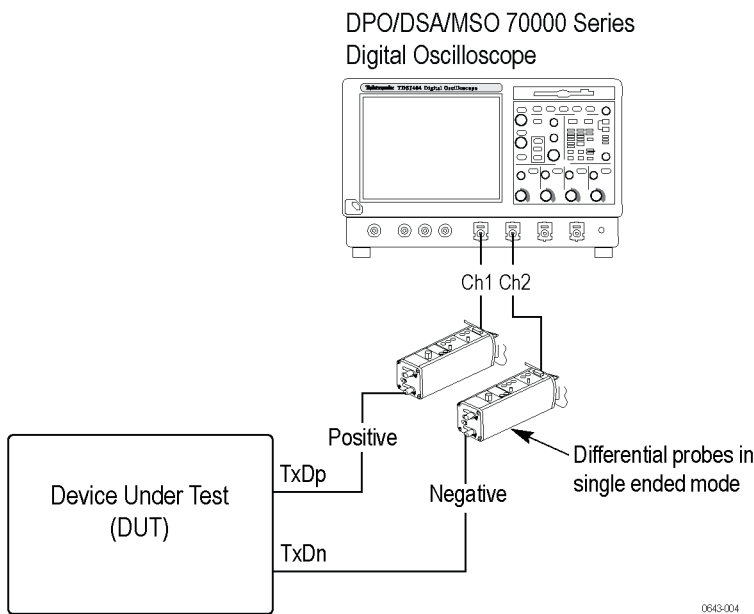
[About running tests \(see page 39\)](#)

Instrument connection setup

The following diagram shows how to connect the DUT to the oscilloscope for all the M-PHY Tx measurements with differential acquisition.



The following diagram shows how to connect the DUT to the oscilloscope for all the M-PHY measurements (HS and PWM) with single ended acquisition.



See also

[Minimum system requirements \(see page 7\)](#)

[View connected instruments \(see page 37\)](#)

[About setting up tests \(see page 35\)](#)

View connected instruments

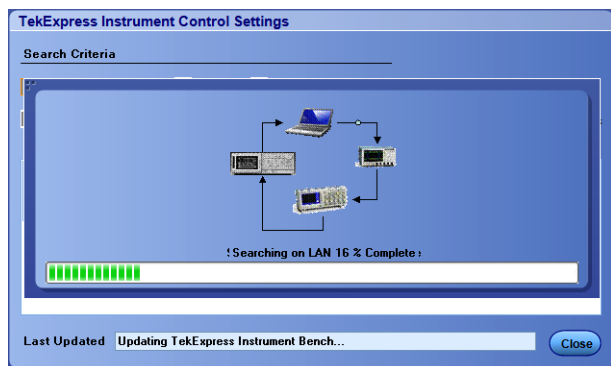
Use the Instrument Control Settings dialog box to view or search for connected instruments required for the tests. The application uses TekVISA to discover the connected instruments.

To refresh the list of connected instruments:

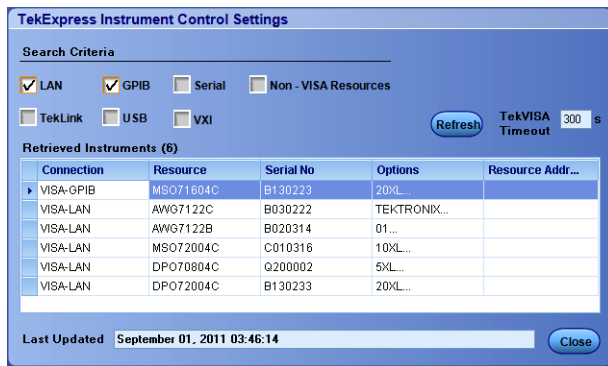
1. From the Options menu, select **Instrument Control Settings**.
2. In the Search Criteria section of the Instrument Control Settings dialog box, select the connection types of the instruments to search for.

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 test parameters \(see page 21\)](#)

[Instrument connection 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 \(see page 28\)](#) to select and configure tests.

1. [Select the DUT parameters \(see page 18\).](#)
2. [Select one or more tests \(see page 20\).](#)
3. [Configure test parameters \(see page 21\).](#)
4. [Select test notification preferences .](#)
5. [Select report options \(see page 29\).](#)

See also

[Test setups overview \(see page 43\)](#)

[Pre-run checklist \(see page 41\)](#)

[Before you click Start \(see page 39\)](#)

[About running tests \(see page 39\)](#)

About running tests

After selecting and configuring the test, [review the pre-run checklist \(see page 41\)](#) 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 M-PHY Tx application on top, select **Keep On Top** from the M-PHY Tx Options menu.

See also

[Configure test parameters \(see page 21\)](#)

[About setting up tests \(see page 35\)](#)

[Before you click Start \(see page 39\)](#)

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 M-PHY Tx, it creates the following folders on the oscilloscope:

- \Program Files (x86)\Tektronix\TekExpress\TekExpress M-PHY Tx

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

- \My Documents\My TekExpress\M-PHY Tx
- \My Documents\My TekExpress\M-PHY Tx\Untitled Session

Every time you launch TekExpress M-PHY Tx, an **Untitled Session** folder is created in the M-PHY Tx folder. The **Untitled Session** folder is automatically deleted when you exit the M-PHY Tx application (you are prompted to save the session before exiting if it has not already been saved or if there are unsaved changes).



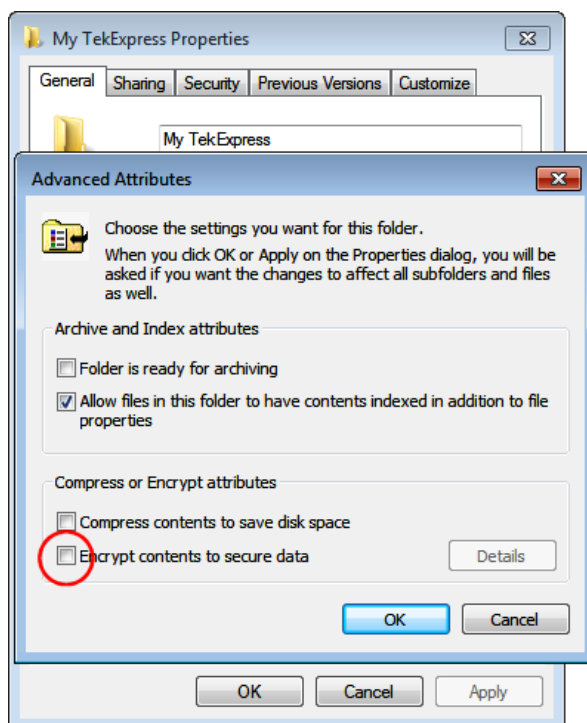
CAUTION. Do not directly edit or modify any of the session files or folders because this may result in loss of data or corrupted session files. Each session has multiple files associated with it. When you save a session, the application creates a .TekX file, and a folder named for the session that contains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

2. [Map the shared My TekExpress folder \(see page 93\)](#) 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 My TekExpress 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 \(see page 41\)](#) before you run a test.

See also

[View test-related files \(see page 28\)](#)

[Application directories and file types \(see page 8\)](#)

[File name extensions \(see page 9\)](#)

Pre-test 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 \(see page 39\)](#).

On the oscilloscope:

- Make sure that all the required instruments are properly warmed up (approximately 20 minutes).
- 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 M-PHYTX application:

1. Verify that the application is able to find the instrument. If it cannot, [perform a search for connected instruments \(see page 37\)](#).
 - 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

[Instrument 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 M-PHY Tx.

See also

[About setting up tests \(see page 35\)](#)

[Save a test setup \(see page 43\)](#)

[Recall a saved test setup \(see page 44\)](#)

[Create a new test setup based on an existing one \(see page 44\)](#)

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 \(see page 29\)](#).
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:\M-PHYTx*<session_name>*.

See also

[About setting up tests \(see page 35\)](#)

[Configure test parameters \(see page 21\)](#)

[Test setup overview \(see page 38\)](#)

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 \(see page 43\)](#)

[Create a new test setup based on an existing one \(see page 44\)](#)

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 \(see page 43\)](#)

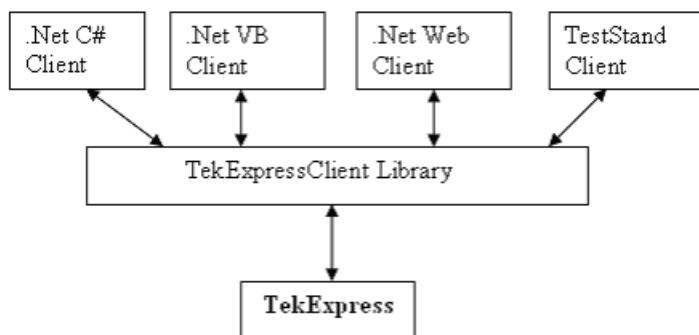
Programmatic interface overview

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 a TekExpress client \(see page 46\)](#)

[Remote proxy object \(see page 47\)](#)

[Client proxy object \(see page 47\)](#)

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.

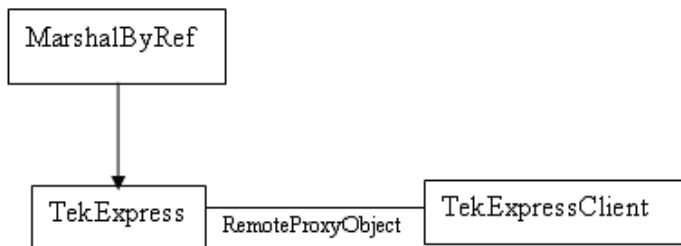
3. When the client has connected to and locked the server, the client can access any of the programmatic controls needed to run the remote automations.
4. After the client operations finish, the client unlocks the server.

See also

[M-PHY Tx application command flow \(see page 54\)](#)

Remote proxy object

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



The following is an example:

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

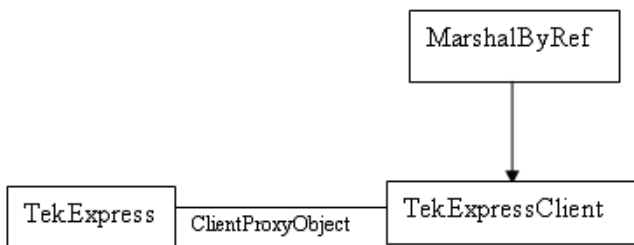
This object lets the remote client access the interfaces exposed at the server side. The client gets the reference to this object when the client gets connected to the server.

For example,

```
//Get a reference to the remote object  
remoteObject = (IRemoteInterface)Activator.GetObject(typeof(IRemoteInterface), URL.ToString());
```

Client proxy object

Client exposes a proxy object to receive certain information.



For example,

```
//Register the client proxy object
wellKnownServiceTypeEntry[] e = RemotingConfiguration.GetRegisteredWellKnownServiceTypes();

clientInterface = new ClientInterface();

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

//Expose the client proxy object through marshalling
RemotingServices.Marshal(clientInterface, "Remote Client Inteface");
```

The client proxy object is used for the following:

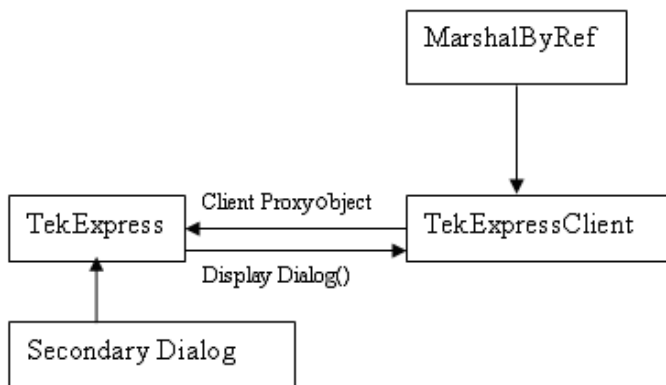
- To get the secondary dialog messages from the server.
- To get the file transfer commands from the server while transferring the report.

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.

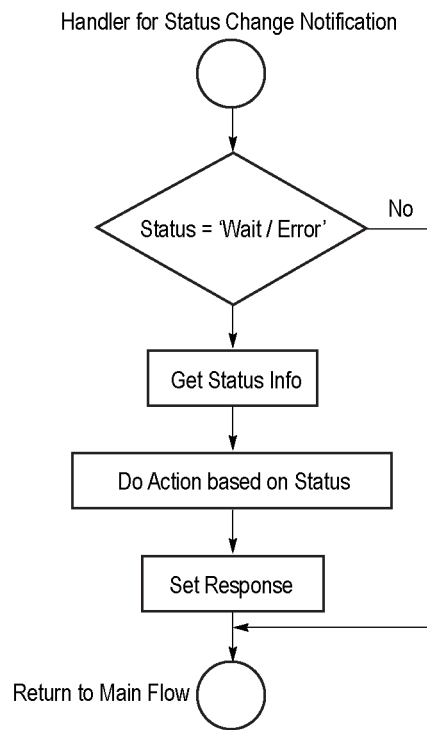
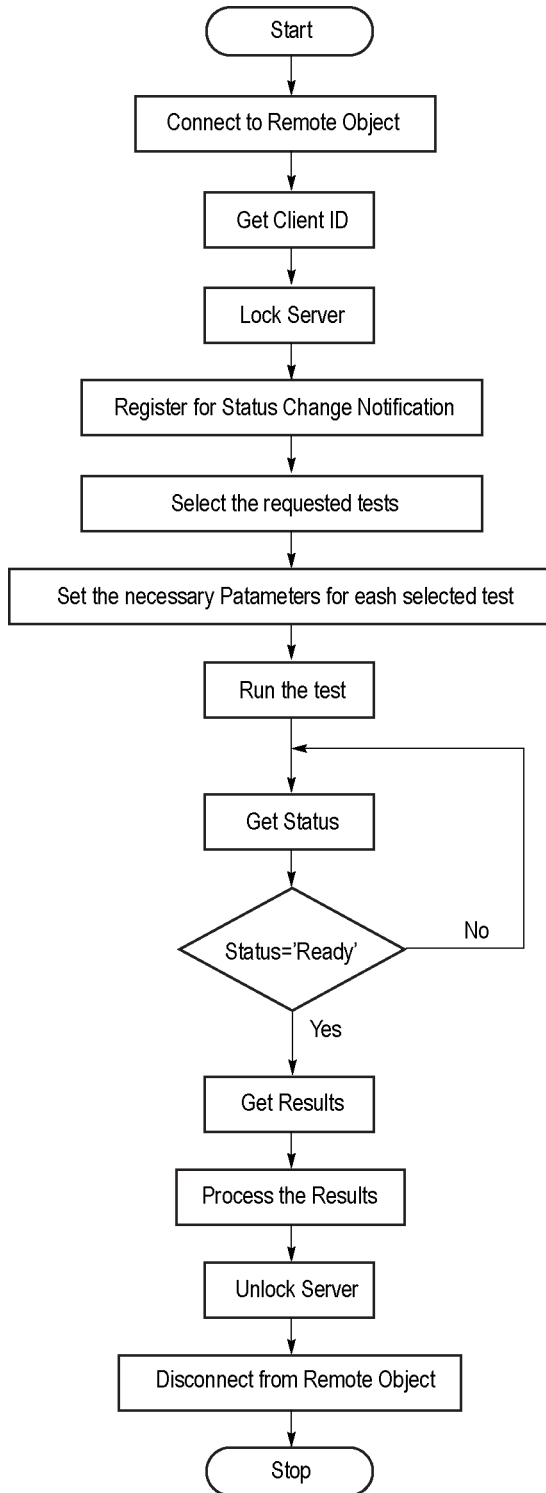
File Transfer Events

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



0643-001

Process overview:

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 \(see page 51\)](#).

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

[M-PHY Tx application command flow \(see page 54\)](#)

[Program remote access code example \(see page 52\)](#)

Program remote access code example

This code example shows how to communicate between a remote PC and TekExpress M-PHY Tx.

Table 17: Remote access code example

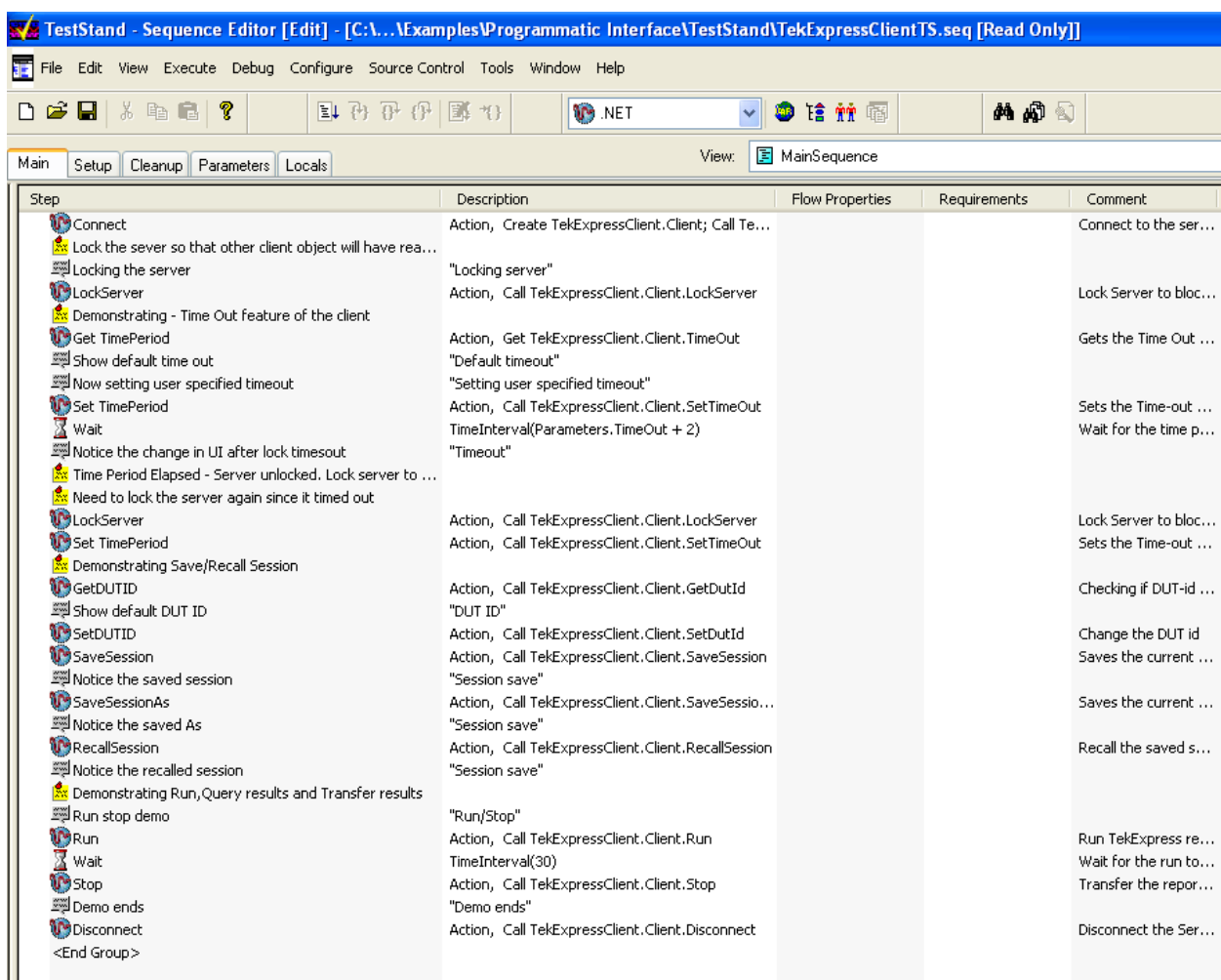
Task	Code
Start the application	
Connect through an IP address.	<code>m_Client.Connect("localhost") 'True or False</code> <code>clientID = m_Client.getClientID</code>
Lock the server	<code>m_Client.LockServer(clientID)</code>
Disable the Popups	<code>m_Client.SetVerboseMode(clientID, false)</code>
Set the DUT ID	<code>m_Client.SetDutId(clientID, "DUT_Name")</code>
Select a test	<code>mClient.SelectsSingleTest(clientID, "Transmitter", "HS", "CTS v0.8", "Test 1.1.6-HS-TX Minimum Differential AC Eye Opening", true)</code>
Select an amplitude	<code>mClient.SetGeneralParameter(clientID, "Transmitter", "HS", " Test 1.1.6-HS-TX Minimum Differential AC Eye Opening ", "Amplitude\$Large Amplitude")</code>
Select a termination	<code>mClient.SetGeneralParameter(clientID, "Transmitter", "HS", " Test 1.1.6-HS-TX Minimum Differential AC Eye Opening ", "Termination\$Unterminated")</code>
Run with set configurations	<code>m_Client.Run(clientID)</code>
Wait for the test to complete.	<code>Do</code> <code> Thread.Sleep(500)</code> <code> m_Client.Application_Status(clientID)</code> <code> Select Case status</code> <code> Case "wait"</code>
Get the current state information	<code>mClient.GetCurrentStateInfo(clientID, waitingMsbBxCaption, waitingMsbBxMessage, waitingMsbBxButtontexts)</code>
Send the response	<code>mClient.SendResponse(clientID, waitingMsbBxCaption, waitingMsbBxMessage, waitingMsbBxResponse)</code> <code>End Select</code> <code>Loop Until status = "Ready"</code>
Save results	'Save all results values from folder for current run <code>m_Client.TransferResult(clientID, logDirname)</code>

Table 17: Remote access code example (cont.)

Task	Code
Unlock the server	<code>m_Client.UnlockServer(clientID)</code>
Disconnect from server	<code>m_Client.Disconnect()</code>
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 M-PHY Tx\Examples\Programmatic Interface\TestStand`



M-PHY Tx application command flow

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

[Connect through an IP address \(see page 59\)](#)

[Lock the server \(see page 60\)](#)

[Disable popup messages \(see page 61\)](#)

[Set or get the DUT ID \(see page 62\)](#)

[Set the configuration parameters for a suite or measurement \(see page 63\)](#)

[Query the configuration parameters for a suite or measurement \(see page 65\)](#)

[Select a test \(see page 66\)](#)

[Select a suite \(see page 67\)](#)

[Select a channel \(see page 68\)](#)

[Configure the selected measurement \(see page 69\)](#)

[Run with set configurations or stop the run operation \(see page 70\)](#)

[Handle error codes \(see page 71\)](#)

[Get or set the timeout value \(see page 72\)](#)

[Wait for the test to complete \(see page 73\)](#)

[After the test is complete \(see page 75\)](#)

[Save, recall, or query a saved session \(see page 78\)](#)

[Unlock the server \(see page 79\)](#)

[Disconnect from the server \(see page 79\)](#)

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 55) out string clientID (see page 55)	This method connects the client to the server. Note (see page 55) 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	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as boolean returnval = m_Client.Connect(ipaddress,m_clientID)</pre>

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

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

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

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

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

Lock the server

Command name	Parameters	Description	Return value	Example
LockSession()	string clientID (see page 59)	This method locks the server. Note (see page 56) 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 popup messages

Use these commands to disable popup messages that require user intervention. Popup messages will interfere with remote access testing procedures.

Command name	Parameters	Description	Return value	Example
SetVerboseMode()	string clientID (see page 59) bool _verbose (see page 58)	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".	<pre>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)</pre>

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

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

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

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

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

Set or get the DUT ID

Command name	Parameters	Description	Return value	Example
SetDutId()	string clientID (see page 59) string dutName (see page 55)	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	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string return=m_Client.SetDutId(clientID,desiredDutId) Note (see page 55)</pre>
GetDutId()	string clientID (see page 59) string dutId (see page 56)	This method gets the DUT ID of the current setup.	String that gives the status of the operation after it has been performed	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string return=m_Client.GetDutId(clientID, out DutId)</pre>

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

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

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

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

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

Set the configuration parameters for a suite or measurement

Command name	Parameters	Description	Return value	Example
SetGeneralParameter	string clientID (see page 59) string device (see page 56) string suite (see page 56) string test (see page 56) string parameterString (see page 56)	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.	<pre>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 64) Select termination example (see page 64)</pre>
SetAnalyzeParameter()	string clientID (see page 59) string device (see page 56) string suite (see page 56) string test (see page 56) string parameterString (see page 56)	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.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string</pre>
SetAcquireParameter()	string clientID (see page 59) string device (see page 56) string suite (see page 56) string test (see page 56) string parameterString (see page 56)	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.	<pre>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)</pre>

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

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

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

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

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

Select channel example

```
returnval=mClient.SetGeneralParameter(clientID, "Transmitter", "PWM", "Test 1.2.1-PWM-TX  
Transmit Bit Duration", "Scope Channel for Trigger$CH1")
```

Select a termination example

```
returnval=mClient.SetGeneralParameter(clientID, "Transmitter", "HS", " Test 1.1.6-HS-TX Minimum  
Differential AC Eye Opening ", "Termination$Unterminated")
```


Query the configuration parameters for a suite or measurement

Command name	Parameters	Description	Return value	Example
GetGeneralParameter()	string clientID (see page 59) string device (see page 56) string suite (see page 56) string test (see page 56) string parameterString (see page 56)	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.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string Query Channel (see page 66)</pre>
GetAnalyzeParameter()	string clientID (see page 59) string device (see page 56) string suite (see page 56) string test (see page 56) string parameterString (see page 56)	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.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string</pre>
GetAcquireParameter()	string clientID (see page 59) string device (see page 56) string suite (see page 56) string test (see page 56) string parameterString (see page 56)	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.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string</pre>

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

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

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

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

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

```
Query channel for trigger example
returnval = mClient.SetGeneralParameter(clientID, "Transmitter", "HS", "Test 1.1.1-HS-TX Unit Interval and Frequency Offset", "Scope Channel for Trigger$CH1")
```

Select a test

Command name	Parameters	Description	Return value	Example
SelectTest()	string clientID (see page 59) string device (see page 56) string suite (see page 56) string test (see page 56) bool isSelected (see page 57)	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.	<pre>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 66)</pre>

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

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

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

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

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

```
Select test example
To select measurement test 1.1.6-HS-TX Minimum Differential AC Eye Opening:
returnval = mClient.SelectTest(clientID, "Transmitter", "HS", "1.1.6-HS-TX Minimum Differential AC Eye Opening", true)
```

Select a suite

Test suite must be set for HS or PWM tests.

Command name	Parameters	Description	Return value	Example
SelectSuite()	string clientID (see page 59) string device (see page 56) string suite (see page 56) bool isSelected (see page 57)	<p>This method selects or deselects a specified suite.</p> <p>When this parameter is set to true, you can select a suite.</p> <p>When this parameter is set to false, you can deselect a suite.</p>	<p>String that gives the status of the operation after it has been performed</p> <p>The return value is "" (an empty String) on success.</p>	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string Select Suite (Default): returnval=mClient.Select- Suite(clientID, "Transmitter", "HS", true)</pre>

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

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

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

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

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

Select a channel

Command name	Parameters	Description	Return value	Example
SetGeneralParameter()	string clientID (see page 59) string device (see page 56) string suite (see page 56) string test (see page 56) string parameterString (see page 56)	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.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string</pre> Set channel for trigger example (see page 69) Set amplitude example (see page 69)
SetAnalyzeParameter()	string clientID (see page 59) string device (see page 56) string suite (see page 56) string test (see page 56) string parameterString (see page 56)	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.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string</pre>
SetAcquireParameter()	string clientID (see page 59) string device (see page 56) string suite (see page 56) string test (see page 56) string parameterString (see page 56)	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.	<pre>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)</pre>

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

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

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

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

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

Set channel for trigger example

```
returnval = mClient.SetGeneralParameter(clientID, "Transmitter", "HS", "Test 1.1.1-HS-TX Unit Interval and Frequency Offset", "Scope Channel for Trigger$CH1")
```

Set amplitude example

```
returnval=mClient.SetGeneralParameter(clientID, "Transmitter", "HS", "Test 1.1.6-HS-TX Minimum Differential AC Eye Opening", "Amplitude$Large Amplitude")
```

Configure the selected measurement

Command name	Parameters	Description	Return value	Example
SetAnalyzeParameter()	string clientID (see page 59) string device (see page 56) string suite (see page 56) string test (see page 56) string parameterString (see page 56)	This method sets the Analyze parameters (Configuration parameters) for a specified test.	The return value is "" (an empty String) on success.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string</pre>

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 59)	Runs the selected tests Note (see page 57) 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 59)	Stops the currently running tests Note (see page 58)	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:

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 59)	Returns the current timeout period set by the client	String that gives the status of the operation after it has been performed The default return value is 1800000.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.GetTimeout()
SetTimeout()	string clientID (see page 59) string time (see page 58)	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.SetTimeout(clientID, desiredTimeOut)

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

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

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

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

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

Wait for the test to complete

The commands in this group 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 59)	This method gets the status of the server application. The states are Ready , Running , Paused , Wait , or Error . (see page 54)	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 59) out string[] status (see page 57)	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 75)

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 59) out string WaitingMsbBxCaption (see page 58) out string WaitingMsbBxMessage (see page 59) out string[] WaitingMsbBxButtontexts (see page 59)	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.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL m_Client.GetCurrentState-Info(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxButtontexts)</pre>
SendResponse() NOTE. This command is used when the application is running and is in the wait or error state.	string clientID (see page 59) out string WaitingMsbBxCaption (see page 58) out string WaitingMsbBxMessage (see page 59) string WaitingMsbBxResponse (see page 59)	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.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL m_Client.SendResponse(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxResponse)</pre>

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

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

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

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

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

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 59) string device (see page 56) string suite (see page 56) string test (see page 56)	This method gets the pass or fail status of the measurement after test completion. NOTE. <i>Execute this command after completing the measurement.</i>	String that gives the status of the operation after it has been performed Returns the pass or fail status in the form of a string	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.GetPassFailStatus(clientID, device, suite, "Test 1.2.1-PWM-TX Transmit Bit Duration")
GetResultsValue()	string clientID (see page 59) string device (see page 56) string suite (see page 56) string test (see page 56) string parameterString (see page 56)	This method gets the result values of the measurement after the run.	String that gives the status of the operation after it has been performed Returns the result value in the form of a string	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.GetResultsValue(clientID,"Transmitter", "PWM", "Test 1.2.1-PWM-TX Transmit Bit Duration", "Measured Value")
GetResultsValueForSubMeasurements()	string clientID (see page 59) string device (see page 56) string suite (see page 56) string test (see page 56) string parameterString (see page 56) int rowNr (see page 56)	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 77)

Command name	Parameters	Description	Return value	Example
GetReportParameter()	string clientID (see page 59) string device (see page 56) string suite (see page 56) string test (see page 56) string parameterString (see page 77)	This method gets the general report details such as oscilloscope model, TekExpress version, and M-PHY Tx version.	The return value is the oscilloscope model, TekExpress version, and M-PHY Tx version.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string</pre> <p>Oscilloscope Model</p> <pre>returnval=m_Client.GetReportParameter(clientID,"Scope Model")</pre> <p>TekExpress Version</p> <pre>returnval=m_Client.GetReportParameter(clientID,"TekExpress Version")</pre> <p>M-PHY Tx Version</p> <pre>returnval=m_Client.GetReportParameter(clientID,"Application Version")</pre>
TransferReport()	string clientID (see page 59) string filePath (see page 58)	<p>This method transfers the report generated after the run.</p> <p>The report contains the summary of the run.</p> <p>The client must provide the location where the report is to be saved at the client-end.</p>	<p>String that gives the status of the operation after it has been performed</p> <p>Transfers all the result values in the form of a string</p>	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string</pre> <pre>returnval=m_Client.TransferReport(clientID,"C:\Report")</pre>
TransferImages()	string clientID (see page 59) string filePath (see page 58)	<p>This method transfers all the images (screen shots) from the specified client and folder for the current run (for a suite or measurement).</p> <p>NOTE. Every time you click Start, a folder is created in the X: drive. Transfer the waveforms before clicking Start.</p>	<p>String that gives the status of the operation after it has been performed</p> <p>Transfers all the images in the form of a string</p>	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string</pre> <pre>returnval=m_Client.TransferImages(clientID, "C:\Waveforms")</pre>

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

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

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

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

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

string parameterString

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

Get results for a submeasurement example

This example returns the specified submeasurement results for test 1.2.1-PWM-TX (transmit bit duration).

```
returnval=m_Client.GetResultsValue( clientID,"Transmitter", "PWM", "Test 1.2.1-PWM-TX Transmit Bit Duration", "Measured Value",0)
```

```
returnval=m_Client.GetResultsValue( clientID,"Transmitter", "PWM", "Test 1.2.1-PWM-TX Transmit Bit Duration", "Measured Value",1)
```

Save, recall, or query a saved session

Command name	Parameters	Description	Return value	Example
CheckSession-Saved()	string clientID (see page 59) out bool saved (see page 55)	This method checks whether the current session is saved.	Return value is either True or False	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.Check- SessionSaved(m_clientID, out savedStatus)</pre>
RecallSession()	string clientID (see page 59) string name (see page 57)	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..."	<pre>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)</pre>
SaveSession()	string clientID (see page 59) string name (see page 57)	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 Saved..."/"Failed..."	<pre>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)</pre>
SaveSessionAs()	string clientID (see page 59) string name (see page 57)	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..."	<pre>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)</pre>

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

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

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

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

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

Unlock the server

Command name	Parameters	Description	Return value	Example
UnlockSession()	string clientID (see page 59)	This method unlocks the server from the client. The ID of the client to be unlocked must be provided. Note (see page 58)	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 59)	This method disconnects the client from the server. Note (see page 55)	Integer value that gives the status of the operation after it has been performed 1 for Success -1 for Failure	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.Disconnect(m_clientID)

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...".

HS test user defined mode parameters

This table lists which HS test parameters can be changed when in the User Defined Mode. See each HS test for which group(s) apply to that command.

Table 18: HS test parameters (User Defined mode)

Parameter group	Parameter name	Range	Default	Units
HS: Acquire	Horizontal scale	1–100	1	μs/div
	Resolution	1–100	10	ps/pt
	Trigger level ^{1, 2}	0–100	0	V
	Slope ¹	RISE, FALL, EITHER	RISE	N/A
	Coupling ¹	DC, AC, HF REJECT, LF REJECT, NOISE REJ	DC	N/A
	Pulse width when ¹	WITHIN, OUTSIDE	WITHIN	N/A
	Lower limit ¹	1–100	10	UI
	Gear 1A/B Upper limit ¹	100–200	150	UI
	Gear 1A/B Polarity ¹	Positive, Negative	Positive	N/A
	Gear 2A/B Upper limit ¹	300–400	300	UI
	Gear 2A/B Polarity ¹	Positive, Negative	Positive	N/A
	Gear 3A/B Upper limit ¹	600–3000	2550	UI
	Gear 3A/B Polarity ¹	Positive, Negative	Negative	N/A
	Sample rate G1A(GS/s) ¹	12.5, 25, 50, 100	25	GS/s
	Sample rate G1B(GS/s) ¹	12.5, 25, 50, 100	25	GS/s
	Sample rate G2A(GS/s) ¹	12.5, 25, 50, 100	25	GS/s
	Sample rate G2B(GS/s) ¹	12.5, 25, 50, 100	25	GS/s
	Sample rate G3A(GS/s) ¹	12.5, 25, 50, 100	50	GS/s
	Sample rate G3B(GS/s) ¹	12.5, 25, 50, 100	50	GS/s

Table 18: HS test parameters (User Defined mode) (cont.)

Parameter group	Parameter name	Range	Default	Units
HS: Ref Levels	Absolute/Percentage	Absolute, Percentage	Absolute	N/A
	Autoset basetop method	AUTO, MINMAX, FULLHISTOGRAM, EYEHISTOGRAM	AUTO	N/A
	High level ³	51 to 90 (in %) -10 to 10 (in Absolute)	90 (in %) 0 (in Absolute)	% or V
	Mid level ³	20 to 80 (in %) -10 to 10 (in Absolute)	50 (in %) 0 (in Absolute)	% or V
	Low level ³	10 to 49 (in %) -10 to 10 (in Absolute)	10 (in %) 0 (in Absolute)	% or V
	Hysteresis ³	2 to 20 (in %) -2 to 2(in Absolute)	5 (in %) 0.05 (in Absolute)	% or V
HS: Filters	Low pass filter (F1) spec	NONE, FIRST, SECOND, THIRD	NONE	N/A
	High pass filter	1–100 (1–10 for test 1.1.1 only)	1	MHz
	Low pass filter (F2) spec	NONE, FIRST, SECOND, THIRD	SECOND	N/A
	Low pass filter	620–630 (1–10 for test 1.1.1 only)	2	MHz
	Filter ramp time	0–10	0.25	/F
	Filter blanking time	0–10	0.25	/F
HS: Prepare Length	Gear1A-HS Prepare Length	0–15	10	N/A
	Gear1B-HS Prepare Length	0–15	10	N/A
	Gear2A-HS Prepare Length	0–15	10	N/A
	Gear2B-HS Prepare Length	0–15	10	N/A
	Gear3A-HS Prepare Length	0–15	10	N/A
	Gear3B-HS Prepare Length	0–15	10	N/A

Table 18: HS test parameters (User Defined mode) (cont.)

Parameter group	Parameter name	Range	Default	Units
HS: Clock Recovery	Clock Recovery Method	Constant Clock-Mean, PLL Custom BW	Constant Clock-Mean	N/A
	PLL Model Type	1/2		N/A
	Damping	500–2000		m
	Loop Bandwidth	0.1–10		MHz
	Nominal data rate	Enabled, Disabled	Enabled	N/A
	Known data pattern	Enabled, Disabled	Disabled	N/A
	Pattern file path	N/A	INSTALL FOLDER\Compliance Suites\MIPI\MPHY\Transmitter\HS\Pattern Files\SomePattern.txt	N/A
HS: Mask file path	Gear Type	Gear1A, Gear1B, Gear2A, Gear2B, Gear3A, Gear3B	Gear1A	N/A
	NT LA Mask File path	Path to mask folder	INSTALL FOLDER\Compliance Suites\MIPI\MPHY\Transmitter\HS\Mask Files\Gear1A\1.1.6_VD-IFF_AC_LA_RT_Gr1A.msk	N/A
	NT SA Mask File path	Path to mask folder	INSTALL FOLDER\Compliance Suites\MIPI\MPHY\Transmitter\HS\Mask Files\Gear1A\1.1.6_VD-IFF_AC_SA_RT_Gr1A.msk	N/A
	RT LA Mask File path	Path to mask folder	INSTALL FOLDER\Compliance Suites\MIPI\MPHY\Transmitter\HS\Mask Files\Gear1A\1.1.6_VD-IFF_AC_SA_RT_Gr1A.msk	N/A
	RT SA Mask File path	Path to mask folder	INSTALL FOLDER\Compliance Suites\MIPI\MPHY\Transmitter\HS\Mask Files\Gear1A\1.1.6_VD-IFF_AC_SA_RT_Gr1A.msk	N/A
HS: Skew	From Edge	RISE, FALL, EITHER	RISE	N/A
	To Edge	SAMEAS, OPPOSITEAS	OPPOSITEAS	N/A

Table 18: HS test parameters (User Defined mode) (cont.)

Parameter group	Parameter name	Range	Default	Units
HS: RjDj	RjDj Pattern type	Arbitrary, Repeating	Arbitrary	N/A
	Arbitrary window length (UI)	2–24	5	UI
	Target BER (1e-)	10 or 12	10	N/A
	Repeating pattern length (UI)	1–1000000	1270	UI
	Population	5–5000	100	N/A

¹ This parameter is global for all tests. If the value is changed in one test, it is applied to all other tests of that group.

² This parameter is applicable only in continuous mode. In Burst mode, AutoSetLevel is executed.

³ The unit of this field (value) is based on the 'Ref Level' parameter. It will be either percentage (%) or Volts (V).

1.1.1 HS-TX Unit Interval and Frequency Offset test

Verifies that the unit interval and frequency offset of the DUTs HS-TX are within the conformance limits for all supported HS GEARS, RATES and LANES.

Configurable parameters. [HS Acquire](#), [HS Ref Levels](#), [HS Filters](#)

1.1.2 HS-TX Common-Mode AC Power Spectral Magnitude Limit test

Verifies that the common-mode AC power spectral magnitude ($P_{SDCM-TX}$) of the DUTs HS-TX is below the conformance limit, for large and small amplitudes, in terminated mode, for all LANES, for HS-G1.

Configurable parameters. [HS Acquire](#)

1.1.3 HS-TX PREPARE Length test

Verifies that the length of the DUT's transmitted HS-PREPARE period ($T_{HS-PREPARE}$) is consistent with the value indicated by its TX_HS_PREPARE_LENGTH configuration attribute.

Configurable parameters. [HS Acquire](#), [HS Prepare Length](#)

1.1.4 HS-TX Common-Mode DC Output Voltage Amplitude test

Verifies that common mode DC output voltage amplitude (V_{CM-TX}) of the DUT's HS TX is within the conformance limits for all combinations of supported amplitudes, terminations, LANEs and HS GEARS.

Configurable parameters. [HS Acquire](#)

1.1.5 HS-TX Differential DC Output Voltage Amplitude test

Verifies that the differential DC output voltage amplitude ($V_{DIF-DC-TX}$) of the DUT's HS-TX is within the conformance limits, for all combinations of supported amplitudes, terminations, LANEs and HS GEARS.

Configurable parameters. [HS Acquire](#)

1.1.6 HS-TX Minimum Differential AC Eye Opening test

Verifies that the DUT's HS-TX meets the requirements for transmitter eye opening (T_{EYE-TX}), at the minimum differential AC output voltage amplitude levels, for all combinations of supported amplitudes, terminations, LANEs and HS GEARS.

Configurable parameters. [HS Acquire](#), [HS Ref Levels](#), [HS Clock Recovery](#), [HS Mask Filter Path](#)

1.1.7 HS-TX Maximum Differential AC Eye Opening test

Verifies that the DUTs HS-TX meets the requirements for the maximum differential AC output voltage amplitude ($V_{DIF-AC-TX}$) for all combinations of supported amplitudes, terminations, LANEs and HS GEARS.

Configurable parameters. [HS Acquire](#), [HS Ref Levels](#), [HS Clock Recovery](#), [HS Mask Filter Path](#)

1.1.8 HS-TX 20/80% Rise and Fall Times test

Verifies that the 20%/80% rise and fall times ($T_{R-HS-TX}$ and $T_{F-HS-TX}$) of the DUT's HS-TX are within the conformance limits, for all combinations of supported amplitudes, terminations, LANEs and HS GEARS.

Configurable parameters. [HS Acquire](#)

1.1.10 HS-TX Slew Rate test

Verifies that the slew rate (SR_{DIF-TX}) of the DUT's HS-TX is within the conformance limits, for all supported amplitudes, in terminated mode, and for all LANEs at HS-G1.

Configurable parameters. [HS Acquire](#)

1.1.11 HS-TX Slew Rate Monotonicity test

Verifies that the slew rate control states of the DUT's HS-TX support monotonically decreasing slew rate settings, for large and small amplitudes, in terminated mode, for all LANEs, for HS-G1.

Configurable parameters. [HS Acquire](#)

1.1.12 HS-TX Slew Rate Resolution test

Verifies that the slew rate state resolution (ΔSR_{DIF-TX}) of the DUT's HS-TX slew rate control satisfies the conformance requirements.

Configurable parameters. [HS Acquire](#)

1.1.13 HS-TX Intra-Lane Output Skew test

Verifies that the intra lane output skew ($T_{INTRA-SKEW-TX}$) of the DUT's HS-TX is within the conformance limits, for all combinations of supported amplitudes, terminations, LANEs and HS GEARS.

Configurable parameters. [HS Acquire](#), [HS Ref Levels](#), [HS Filters](#), [HS Skew](#)

1.1.14 HS-TX Transmitter Pulse Width test

Verifies that the pulse width ($T_{PULSE-TX}$) of the DUT's HS-TX is within the conformance limits, for all combinations of supported amplitudes, terminations, LANEs and HS-GEARS.

Configurable parameters. [HS Acquire](#), [HS Ref Levels](#), [HS Filters](#)

1.1.15 HS-TX Total Jitter test

Verifies that the total jitter (TJ_{TX}) of the DUT's HS-TX is within the conformance limits, for all combinations of supported amplitudes, terminations, LANEs and HS GEARS.

Configurable parameters. [HS Acquire](#), [HS Ref Levels](#), [HS Clock Recovery](#), [HS Filters](#), [HS RjDj](#)

1.1.16 HS-TX Short-Term Total Jitter test

Verifies that the short-term total jitter ($STTJ_{TX}$) of the DUT's HS-TX is within the conformance limits, for all combinations of supported amplitudes, terminations, LANEs and HS GEARS.

Configurable parameters. [HS Acquire](#), [HS Ref Levels](#), [HS Clock Recovery](#), [HS Filters](#), [HS RjDj](#)

1.1.17 HS-TX Deterministic Jitter test

Verifies that the deterministic jitter (DJ_{TX}) of the DUT's HS-TX is within the conformance limits, for all combinations of supported amplitudes, terminations, LANEs and HS GEARS.

Configurable parameters. [HS Acquire](#), [HS Ref Levels](#), [HS Clock Recovery](#), [HS Filters](#), [HS RjDj](#)

1.1.18 HS-TX Short-Term Deterministic Jitter test

Verifies that the short term deterministic jitter ($STDJ_{TX}$) of the DUT's HS-TX is within the conformance limits, for all combinations of supported amplitudes, terminations, LANEs and HS GEARS.

Configurable parameters. [HS Acquire](#), [HS Ref Levels](#), [HS Clock Recovery](#), [HS Filters](#), [HS RjDj](#)

Global HS test parameters list

Table 19: Common HS test parameters

Parameter	Enum values	Default	Minimum	Maximum
DUT Operation Mode	Burst Continuous	Burst	NA	NA
Amplitude	Both Supported Large Amplitude Small Amplitude	Large Amplitude	NA	NA
Termination	Both Supported Terminated Unterminated	Terminated	NA	NA
Gear1	Included Excluded	Included	NA	NA
Gear Selection for Gear1	A : 1.248Gbps B : 1.457Gbps A and B	A : 1.248Gbps	NA	NA
Gear2	Included Excluded	Excluded	NA	NA
Gear Selection for Gear2	A : 2.496Gbps B : 2.915Gbps A and B	A : 2.496Gbps	NA	NA
Gear3	Included Excluded	Excluded	NA	NA
Gear Selection for Gear3	A : 4.992Gbps B : 5.830Gbps A and B	A : 4.992Gbps	NA	NA
Vterm Source (V _t)	NA	0	-4	4
Number of slew rates	NA	4	2	8
Signal Validation	Prompt me if signal fails Skip test if signal fails Use signal as is - Don't Check	Use signal as is - Don't Check	NA	NA

PWM tests user defined mode parameters

This table lists which PWM parameters can be changed when in the User Defined Mode. See each PWM test for which group(s) apply to that command.

Table 20: PWM test parameters (User Defined mode)

Parameter group	Parameter name	Range	Default	Units
PWM: Acquire	Horizontal scale	1000–10000	1	µs/div
	Resolution	2000–10000	10	ps/pt
	Sampling Rate ¹	0.10, 0.25, 0.50	0.25	GS/s
	Trigger Level ^{1, 2}	0–100	0	V
	Slope ¹	RISE, FALL, EITHER	RISE	N/A
	Coupling ¹	DC, AC, HF REJECT, LF REJECT, NOISE REJ	DC	N/A
	Pulse width when ¹	WITHIN, OUTSIDE	WITHIN	N/A
	Polarity	Positive, Negative	Positive	N/A
	Lower limit ¹	1–10	1	UI
	Gear 0 Upper limit ¹	1–20	15	UI
	Gear 1 Upper limit ¹	1–20	15	UI
	Gear 2 Upper limit ¹	1–20	15	UI
	Gear 3 Upper limit ¹	1–20	15	UI
	Gear 4 Upper limit ¹	1–20	15	UI
	Gear 5 Upper limit ¹	1–20	15	UI
	Gear 6 Upper limit ¹	1–20	15	UI
	Gear 7 Upper limit ¹	1–20	15	UI
	PWM: Prepare Length	G0 - LS Prepare Length	0–15	10
G1 - LS Prepare Length		0–15	9	N/A
G2 - LS Prepare Length		0–15	8	N/A
G3 - LS Prepare Length		0–15	7	N/A
G4 - LS Prepare Length		0–15	6	N/A
G5 - LS Prepare Length		0–15	5	N/A
G6 - LS Prepare Length		0–15	4	N/A
G7 - LS Prepare Length		0–15	3	N/A

¹ This parameter will be global for all tests. If the value is changed in one test, it will be applied to all other tests of that group.

² This parameter is applicable only in continuous mode. In Burst mode, AutoSetLevel is executed.

1.2.1 PWM-TX Transmit Bit Duration test

Verifies that the transmit bit duration ($T_{\text{PWM-TX}}$) of the DUT's PWM-TX is within the conformance limits for all combinations of supported amplitudes, terminations, LANEs, and PWM GEARS.

Configurable parameters. [PWM Acquire](#)

1.2.2 PWM-TX Transmit Ratio test

Verifies that the PWM transmit ratio ($k_{\text{PWM-TX}}$) of the DUT's PWM-TX is within the conformance limits for all LANEs.

Configurable parameters. [PWM Acquire](#)

1.2.3 PWM-TX PREPARE Length test

Verifies that the length of the DUT's transmitted period ($P_{\text{WM-PREPARE}}$) is consistent with the value indicated by its TX_LS_PREPARE_LENGTH configuration attribute.

Configurable parameters. [PWM Acquire](#), [PWM Prepare Length](#)

1.2.4 PWM-TX Common-Mode DC Output Voltage Amplitude test

Configurable parameters. [PWM Acquire](#)

1.2.5 PWM-TX Differential DC Output Voltage Amplitude test

Verifies that the Differential DC Output Voltage Amplitude ($V_{\text{DIF-DC-TX}}$) of the DUT's PWM-TX is within the conformance limits for all combinations of supported amplitudes, terminations, LANEs, and PWM GEARS.

Configurable parameters. [PWM Acquire](#)

1.2.8 PWM-TX 20/80% Rise and Fall Times test

Verifies that the rise and fall times ($T_{R-PWM-TX}$ and $T_{F-PWM-TX}$) of the DUT's PWM-TX are less than the maximum conformance limit for all combinations of supported amplitudes, terminations, LANEs, and PWM GEARS.

Configurable parameters. [PWM Acquire](#)

1.2.10 PWM-TX G1 Transmit Bit Duration Tolerance test

Verifies that the transmit bit duration tolerance ($TOL_{PWM-G1-TX}$) of the DUT's PWM-TX is within the conformance limits for PWM-GEAR1

Configurable parameters. [PWM Acquire](#)

1.2.11 PWM-TX G0 Minor Duration test

Verifies that the PWM-G0 Minor Duration ($T_{PWM-MINOR-G0-TX}$) of the DUT's PWM-TX is within the conformance limits, for all LANEs.

Configurable parameters. [PWM Acquire](#)

Global PWM test parameters list

Table 21: Common PWM test parameters

Parameter	Enum values	Default	Minimum	Maximum
DUT Operation Mode	Burst Continuous	Burst	NA	NA
Amplitude	Both Supported Large Amplitude Small Amplitude	Large Amplitude	NA	NA
Termination	Both Supported Terminated Unterminated	Terminated	NA	NA
Gear0	Included Excluded	Included	NA	NA
Gear1 through Gear7	Included Excluded	Excluded	NA	NA
Vterm source (V _t)	NA	0	-4	4
Signal Validation	Prompt me if signal fails Skip test if signal fails Use signal as is - Don't Check	Use signal as is - Don't Check	NA	NA

See also

[Configure test parameters \(see page 21\)](#)

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 M-PHYTX 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**.

Waveform naming conventions (pre-recorded mode)

HS test output file name conventions

The following file name syntax is used for HS test output waveform files:

[LANEx] [CHy/MATHy] [CRPAT] [Termination] [Gear] [Amplitude] [Slew Rate] [Horizontal Scale]
[Resolution] [POS/NEG].wfm

Table 22:

HS tests (1.1.xx)	Acquire type	Example file names
1, 3–8	CRPAT	Lane0 CH1 CRPAT RT Gear1A LA SR1 10usdiv 10pspt POS.wfm Lane0 CH2 CRPAT RT Gear1A LA SR1 10usdiv 10pspt NEG.wfm
2	CRPAT PSD-Single-ended ¹	Lane0 CH1 CRPAT RT Gear1A LA SR1 10usdiv 10pspt POS.wfm Lane0 CH2 CRPAT RT Gear1A LA SR1 10usdiv 10pspt NEG.wfm Lane0 MATH1 CRPAT RT Gear1A LA SR1 10usdiv 10pspt CM.wfm Lane0 MATH2 CRPAT RT Gear1A LA SR1 10usdiv 10pspt SPECMAG.wfm Lane0 MATH2 CRPAT RT Gear1A LA SR1 10usdiv 10pspt OUTdB.wfm ² Lane0 MATH2 CRPAT RT Gear1A LA SR1 10usdiv 10pspt OUTLimit.wfm ²
10–12	CRPAT Slew Rate	Lane0 CH1 CRPAT RT Gear1A LA SR1 10usdiv 10pspt POS.wfm Lane0 CH2 CRPAT RT Gear1A LA SR1 10usdiv 10pspt NEG.wfm
13	CRPAT Single-ended	Lane0 CH1 CRPAT RT Gear1A LA SR1 10usdiv 10pspt POS.wfm Lane0 CH2 CRPAT RT Gear1A LA SR1 10usdiv 10pspt NEG.wfm

¹ Test 1.1.2 (HS-TX Common-Mode AC Power Spectral Magnitude Limit) cannot be executed in pre-recorded mode.

² These waveforms are output of PSD test.

PWM test output file name conventions

The following file name syntax is used for PWM test output waveform files:

[LANEx] [CHy] [CRPAT] [Termination] [Gear] [Amplitude] [Horizontal Scale] [Resolution] [POS/NEG].wfm

The following are a few PWM file name examples:

Lane0 CH3 CRPAT RT Gear0 LA 2000usdiv 4000pspt POS.wfm

Lane0 CH3 CRPAT RT Gear0 LA 2000usdiv 4000pspt NEG.wfm

Index

A

- Acquire status, 25
- Acquisitions tab, 23
- Activating the license (M-PHY Tx), 3
- Analysis Status, 25
- Application
 - controls, 12
 - directories and files, 8
 - exit, 11
 - features, 5
 - folders, 8
 - license, 11
 - version information, 4
 - window, 12
- ApplicationStatus(), 73
- Auto Scroll, 25

C

- Channel selection commands, 68
- CheckSessionSaved(), 78
- Clear test log, 25
- Client (TekExpress), 45
- Client programmatic interface
 - flowchart, 49
- Client proxy object, 47
- Client requirements,
 - TekExpress, 46
- Code example, remote access, 52
- Command buttons, 12
- Commands
 - ApplicationStatus(), 73
 - CheckSessionSaved(), 78
 - Connect(), 59
 - Disconnect(), 79
 - error codes, 71
 - GetAcquireParameter(), 65
 - GetAnalyzeParameter(), 65
 - GetCurrentStateInfo(), 73
 - GetDutId(), 62
 - GetGeneralParameter(), 65
 - GetPassFailStatus(), 75
 - GetReportParameter(), 75

- GetResultsValue(), 75
- GetResultsValueForSubMeasurements(), 75
- GetTimeout(), 72
- LockSession(), 60
- QueryStatus(), 73
- RecallSession(), 78
- Run(), 70
- SaveSession(), 78
- SaveSessionAs(), 78
- SelectSuite(), 67
- SelectTest(), 66
- SendResponse(), 73
- SetAcquireParameter(), 63
- SetAnalyzeParameter(), 69
- SetDutId(), 62
- SetGeneralParameter(), 68
- SetTimeout(), 72
- SetVerboseMode(), 61
- Stop(), 70
- TransferImages(), 75
- TransferReport(), 75
- UnlockSession(), 79
- Commands list, 54
- Common parameters
 - HS tests, 88
 - PWM tests, 92
- Compare String, 23
- Compliance Mode, 21
- Configuration Parameter
 - query, 65
- Configure button, 20
- Configure email settings, 15
- Configure Selected Measurement
 - commands, 69
- Configuring tests, 21
- Connect(), 59
- Connected instrument(s), 22
- Connected instruments, 37
- Contact Tektronix, 2
- Conventions, 1

D

- Default application directory, 8
- Demo mode, 11
- Description of main application
 - panels, 18
- Deskew utility, 16
- Differential probe setup, 36
- Directories and files, 8
- Disable Popups command, 61
- Disconnect from Server
 - command, 79
- Disconnect(), 79
- Download the software, 3
- DUT
 - connection setup, 36
 - definition, 1
 - ID commands, 62
 - settings, 18
 - tab, 18
- DUT connection diagrams, 36

E

- Email notifications, 15
- Email settings, 14
- Equipment setup, 36
- Error codes, 71
- Evaluation mode, 11
- Exiting the application, 11
- Extensions, file name, 9

F

- File name extensions, 9
- Files and directories, 8

G

- Gear selection, 18
- GetAcquireParameter(), 65
- GetAnalyzeParameter(), 65
- GetCurrentStateInfo(), 74
- GetDutId(), 62
- GetGeneralParameter(), 65
- GetPassFailStatus(), 75

GetReportParameter(), 76
 GetResultsValue(), 75
 GetResultsValueForSubMeasurements(), 75
 GetTimeOut(), 72
 Global settings, 21
 Global settings (HS tests), 88
 Global settings (PWM tests), 92
 Global Settings tab, 22

H

High Limit, 23
 HS common parameters, 88
 HS tests, 5

- 1.1.1 HS-TX Unit Interval and Frequency Offset, 84
- 1.1.10 HS-TX Slew Rate, 86
- 1.1.11 HS-TX Slew Rate Monotonicity, 86
- 1.1.12 HS-TX Slew Rate Resolution, 86
- 1.1.13 HS-TX Intra-Lane Output Skew, 86
- 1.1.14 HS-TX Transmitter Pulse Width, 86
- 1.1.15 HS-TX Total Jitter, 87
- 1.1.16 HS-TX Short-Term Total Jitter, 87
- 1.1.17 HS-TX Deterministic Jitter, 87
- 1.1.18 HS-TX Short-Term Deterministic Jitter, 87
- 1.1.2 HS-TX Common-Mode AC Power Spectral Magnitude Limit, 84
- 1.1.3 HS-TX PREPARE Length, 84
- 1.1.4 HS-TX Common-Mode DC Output Voltage Amplitude, 85
- 1.1.5 HS-TX Differential DC Output Voltage Amplitude, 85
- 1.1.6 HS-TX Minimum Differential AC Eye Opening, 85

- 1.1.7 HS-TX Maximum Differential AC Eye Opening, 85
- 1.1.8 HS-TX 20-80% Rise and Fall Times, 85

I

Installing the software, 3
 Instrument reset before each test run, 16
 Instruments

- connected, 22
- selecting, 22
- viewing connected, 37

Instruments detected, 22
 Interface error codes, 71
 IP address, 93
 IP address connection commands, 59

K

Keep On Top, 11
 Key features, 5

L

Lane source, 23
 License, 11
 License activation (M-PHY Tx), 3
 License information, 4
 Limit test Details field, 23
 Limits Editor, 23
 LockSession(), 60
 Log files, 25
 Log View tab (Status panel), 25
 Low Limit, 23

M

M-PHY Tx Application

- Commands, 54

 M-PHY Tx features, 5
 Main application panels, 18
 Map the My TekExpress folder, 93
 Measurement selection command, 66

Measurements tab, 22
 Menu, 12

- Options, 13
- Preferences, 27

 Message History, 25
 Minimum system requirements, 7
 My TekExpress folder, 28

- locating, 39
- mapping, 93

O

Options menu, 13

- Instrument Control Settings, 14
- Keep On Top, 11

 Oscilloscopes supported, 5

P

Panels

- Reports, 28
- Results, 26
- Setup, 18
- Status, 25

 Panels overview, 18
 Pass/Fail summary, 31
 Pre-run checklist, 39
 probes supported, 6
 Program example, 52
 Programmatic interface

- client, 45
- client programmatic interface flowchart, 49
- client proxy object, 47
- client requirements, 46
- example, 52
- overview, 45
- remote proxy object, 47
- server, 45

 PWM common parameters, 92
 PWM tests, 6

- 1.2.10 PWM-TX G1
 - Transmit Bit Duration Tolerance, 91
- 1.2.11 PWM-TX G0 Minor Duration, 91
- 1.2.2 PWM-TX Transmit Ratio, 90

- 1.2.3 PWM-TX PREPARE Length, 90
- 1.2.4 PWM-TX Common-Mode DC Output Voltage Amplitude, 90
- 1.2.5 PWM-TX Differential DC Output Voltage Amplitude, 90
- 1.2.8 PWM-TX 20/80% Rise and Fall Times, 91
- xxx, 90

Q

QueryStatus(), 73

R

- Recall a test setup, 44
- RecallSession(), 78
- Related documentation, 1
- Remote proxy object, 47
- Reports
 - examples, 32
 - file type (.mht, .html, .csv, .xls), 29
 - main panel, 28
 - names, 29
 - options, 29
 - selecting contents, 29
 - viewing, 31
- Reset before test run, 16
- Restore oscilloscope settings, 16
- Results panel, 26
- Run commands, 70
- Run(), 70
- Running tests, 39

S

- Save oscilloscope settings before running tests, 16
- SaveSession(), 78
- SaveSessionAs(), 78
- Saving test setups, 43
- Schematic button, 20
- Select lane source, 23
- Select Required button, 20
- Select tests command, 66

- Selecting a connected instrument, 22
- SelectSuite(), 67
- SelectTest(), 66
- SendResponse(), 74
- Server (TekExpress), 45
- Server control, 46
- Server locking commands, 60
- Session files, 28
- Session folders, 28
- Session Status commands, 78
- Set Configuration Parameters commands, 63
- Set DUT parameters, 18
- Set email parameters, 14
- SetAcquireParameter(), 63
- SetAnalyzeParameter(), 63
- SetDutId(), 62
- SetGeneralParameter(), 63
- SetTimeout(), 72
- Setting up tests, 38
- Setup panel, 18
- Setup tabs
 - Acquisitions, 23
 - DUT, 18
 - Test Selection, 20
- SetVerboseMode(), 61
- Show application version, 4
- Show Detailed Log, 25
- Signal Path Compensation (SPC), 39
- Signal source validation
 - overview, 23
 - setting, 23
- single ended probe setup, 36
- Software download and installation, 3
- Status panel, 25
- Stop commands, 70
- Stop(), 70
- Suite (session) selection command, 67
- Supported instruments, 6
- Supported oscilloscopes, 5
- Supported probes, 5
- System requirements, 7

T

- Tab
 - Acquisitions, 23
 - Configuration, 21
 - DUT, 18
 - Test Selection, 20
- Tabs
 - Log View, 25
 - Test Status, 25
- Technical support, 2
- Tektronix support, 2
- Test configuration, 21
- Test groups, 35
- Test parameters (HS), 88
- Test parameters (PWM), 92
- Test Results, 27
- Test Results commands, 75
- Test run instrument reset, 16
- Test Selection button, 21
- Test selection controls, 20
- Test Selection tab, 20
- Test setup files, 28
- Test setup overview, 35
- Test setup steps, 38
- Test setups
 - creating based on existing, 44
 - recalling, 44
 - saving, 43
- Test setups overview, 43
- Test Status commands, 73
- Test Status tab (Status panel), 25
- Tests, 21
 - running, 39
 - selecting, 20
 - setting up, 35
- TestStand Client example, 53
- Timeout value commands, 72
- TransferImages(), 76
- TransferReport(), 76

U

- Unlock Server command, 79
- UnlockSession(), 79
- Untitled Session folder, 39
- User Comments, 18
 - including in test reports, 29
 - location in reports, 31

User Defined Mode, 21

V

View test parameters, 21

View test reports, 31