

TekExpress®
C-PHY Tx Essentials
Printable Application Help



TekExpress®
C-PHY Tx Essentials
Printable Application Help

Supports C-PHY-Tx Product Firmware V1.0 and above.

Copyright © Tektronix. All rights reserved. Licensed software products are owned by Tektronix or its subsidiaries or suppliers, and are protected by national copyright laws and international treaty provisions. Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specifications and price change privileges reserved.

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

Contacting Tektronix

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

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

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

Table of Contents

| | |
|---------------|---|
| Welcome | v |
|---------------|---|

Getting help and support

| | |
|--------------------------------|---|
| Related documentation | 1 |
| Conventions used in help | 1 |
| Technical support | 1 |

Getting started

| | |
|---|---|
| Installing the software | 3 |
| Install the software | 3 |
| Activate the license | 3 |
| View software version and license information | 3 |
| Application overview | 5 |
| C-PHY Tx Essentials key features | 5 |
| Supported oscilloscopes and probes | 5 |
| Minimum system requirements | 6 |
| Application directories and files | 7 |
| File name extensions | 7 |

Operating basics

| | |
|--------------------------------------|----|
| Run the application | 9 |
| Exit the application | 9 |
| Application controls and menus | 10 |
| Application controls | 10 |
| Options menu | 11 |
| Application panels | 14 |
| Application panel overview | 14 |
| Setup panel | 15 |
| Status panel overview | 22 |
| Results panel | 23 |
| Reports panel | 25 |

Setting up and configuring tests

| | |
|-----------------------------------|----|
| About setting up tests | 29 |
| Instrument connection setup | 29 |

| | |
|----------------------------------|----|
| View connected instruments | 31 |
| Test setup overview | 32 |

Running tests

| | |
|------------------------------|----|
| About running tests | 33 |
| Before you click start | 33 |
| Pre-test checklist | 35 |

Saving and recalling test setups

| | |
|--|----|
| Test setups overview | 37 |
| Save a test setup | 37 |
| Recall a saved test setup | 38 |
| Create a new test setup based on an existing one | 38 |

TekExpress programmatic interface

| | |
|---|----|
| Programmatic interface overview | 39 |
| Requirements for developing TekExpress client | 40 |
| To enable remote access | 40 |
| Remote proxy object | 42 |
| Client proxy object | 43 |
| Client programmatic interface overview | 44 |
| Program remote access code example | 46 |
| C-PHY Tx Essentials command flow | 47 |
| Connect through an IP address | 48 |
| Lock the server | 49 |
| Disable popup messages | 50 |
| Set or get the DUT ID | 51 |
| Set the configuration parameters for a suite or measurement | 52 |
| Query the configuration parameters for a suite or measurement | 55 |
| Select a test | 58 |
| Select a channel | 59 |
| Configure the selected measurement | 62 |
| Run with set configurations or stop the run operation | 63 |
| Handle error codes | 64 |
| Get or set the timeout value | 65 |
| Wait for the test to complete | 66 |
| After the test is complete | 70 |
| Save recall or query a saved session | 76 |

| | |
|----------------------------------|----|
| Unlock the server | 78 |
| Disconnect from the server | 79 |

Reference

| | |
|---|----|
| C-PHY Tx Essentials Parameters | 81 |
| General parameters | 81 |
| Test parameters | 82 |
| Map my TekExpress folder | 83 |
| Rise time | 83 |
| Fall time | 83 |
| Eye measurement | 83 |
| DC common mode measurement | 83 |
| AC common mode mismatch measurement | 83 |
| AC common mode level variation $\Delta VCPTX(LF)$ between (50 MHz to 450 MHz) | 84 |
| AC common mode level variation $\Delta VCPTX(HF)$ (Above 450 MHz) | 84 |
| Intra-pair skew | 84 |

Welcome

The Tektronix C-PHY Tx Essentials Automated Test software runs on Tektronix real-time oscilloscopes that are based on Windows 7 computer operating systems. C-PHY Tx Essentials delivers the TekExpress 3.0 based C-PHY automation solution with Tx measurement. It verifies the Eye diagram and performs width, and height measurement for C-PHY signals.

Getting help and support

Related documentation

The following information is available as part of the TekExpress C-PHY Tx Essentials 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 c-phy tx', type = manual; sort by date for latest release) |

See also

[Technical support](#)

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.

Getting started

Installing the software

Install the software

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

To download and install the files:

1. Close the TekScope application.
2. Go to the www.tek.com Web site and search for C-PHY Tx to locate the installation file. Download the file TekExpressC-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 7 instruments: C:\Program Files (x86)\Tektronix\TekExpress\TekExpress C-PHYTx Essentials

5. To run the application, open the TekScope application and select **Analyze > C-PHY Tx Essentials**.

See also. [Minimum system requirements](#)

[Supported oscilloscopes and probes](#)

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 C-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](#)

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.

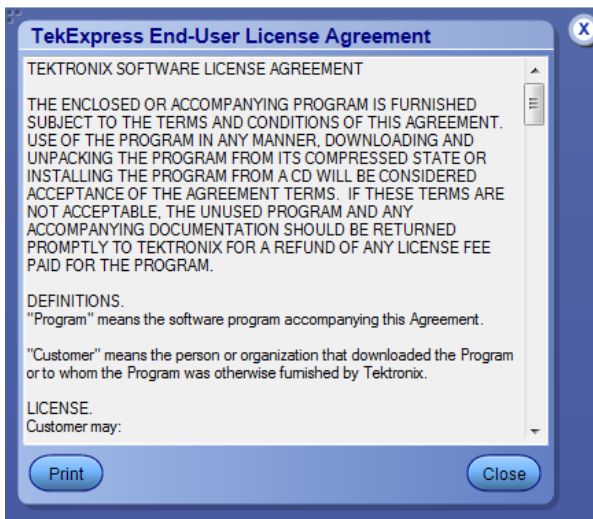
To view the version information:

1. Select **Options > About TekExpress**.
2. You can view the version numbers of the installed test suites. Click **OK** to close the dialog box when finished.



To view the end-user license agreement:

1. Select **Options > About TekExpress**.
Click **View End-User Agreement**. You can view the TekExpress End-User License Agreement.
2. Click **Print** to print the software license agreement.
3. Click **Close**, to close the dialog box.



To view the license information:

1. In the TekScope application, select **Help > About TekScope**.
The Options drop-down displays a list of installed options, including C-PHY Tx Essentials: C-PHY Tx Essentials solution.
2. Look in the **Option Installation Key** section for option key information.

See also. [Activate the license](#)

[Options menu overview](#)

Application overview

C-PHY Tx Essentials key features

The Tektronix C-PHY TX Essentials Automated Test software runs on Tektronix real-time oscilloscopes that are based on Windows 7 computer operating systems. C-PHY Tx Essentials delivers the TekExpress 3.0 based C-PHY automation solution with Tx measurement.

- Supports De embed and Embed software for three port on either side (6-port parameter support for de-embedding).
- Verifies the rise time and fall time of the DUT C-PHY signals.
- Performs both eye height and eye height measurements, and also verifies the eye diagram on C-PHY signals.
- Verifies that the static point common mode voltage VCPTX of the Trio signal are within the Transmitter limit.
- Verifies that the common-mode voltage mismatch (ΔV_{CMTPX}) of the DUT Data Lane HS Transmitter is less than the maximum conformance limit.
- Verifies that the common mode level variation is between 50 MHz to 450 MHz.
- Verifies that the common mode level variation is above 450MHz.
- Measures the Intra pair skew of the trio signal.

Supported oscilloscopes and probes

The C-PHY Tx Essentials application runs on the following Tektronix oscilloscopes:

- MSO/DPO70604C
- MSO/DPO70804C
- MSO/DPO/DSA 71254C
- MSO/DPO/DSA71604C
- MSO/DPO/DSA72004C
- DPO/DSA72504D
- DPO/DSA73304D
- MSO/DPO72304DX
- MSO/DPO72504DX
- MSO/DPO73304DX

You need the following accessories to run C-PHY Tx Essentials:

- Two Bias tees
- Three 100psec RT/FT filters
- Two BNC T connectors
- Three matched pair SMA cables
- Six male to male SMA connectors
- Three binding posts (Female to Female)
- Three TTC convertors
- Amplifier

Minimum system requirements

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

Table 2: System requirements

| Component | Requirement |
|------------------|---|
| Oscilloscope | Supported oscilloscopes and probes |
| Processor | Same as the oscilloscope |
| Operating System | Same as the oscilloscope: Win 7 64-bit |
| Memory | Same as the oscilloscope |
| Hard Disk | Same as the oscilloscope |
| Display | Same as the oscilloscope ¹ |
| Firmware | DPO/DSA/MSO TekScope v7.3.0.9 or later and DPOJET version is 6.2.1.8 |
| Software | <ul style="list-style-type: none"> ■ 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 | Supported oscilloscopes and probes |
| 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 |

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](#)

[Instrument connection setup](#)

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.

Application directories and files

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

Table 3: Application directories

| Directory names | Contains |
|---|---|
| InstallDir\TekExpress\TekExpress CPHYTX | 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 |
| Bin | Miscellaneous C-PHY Tx Essentials application libraries |
| Compliance Suites | Compliance-specific files |
| Examples | Various support files |
| ICP | Instrument and C-PHY Tx Essentials application-specific interface libraries |
| Lib | Utility files specific to the C-PHY Tx Essentials application |
| Report Generator | Excel Active X interface Library for Report Generation |
| Images | Images related to C-PHY Tx Essentials are saved |
| Tools | Instrument and C-PHY Tx Essentials application-specific files |

See also. [View test-related files](#)

[File name extensions](#)

File name extensions

The C-PHY Tx Essentials 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 .mhtml .pdf | Test result report file formats (.mht is the default format). |

See also. [Select report options](#)

[View test-related files](#)

[Application directories and files](#)

Operating basics

Run the application

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

- Select **Analyze > C-PHY Tx Essentials** from the TekScope menu, OR
- Double-click any saved C-PHY Tx Essentials 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 C-PHY Tx Essentials. 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](#)

Exit the application

Use the following method to exit the application:






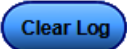


NOTE. Using other methods to exit the application results in abnormal termination of the application.

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, OR
 - A message box appears asking if you really want to exit TekExpress. To exit, click **Yes**.

Application controls and menus

Application controls

Table 4: Application control descriptions

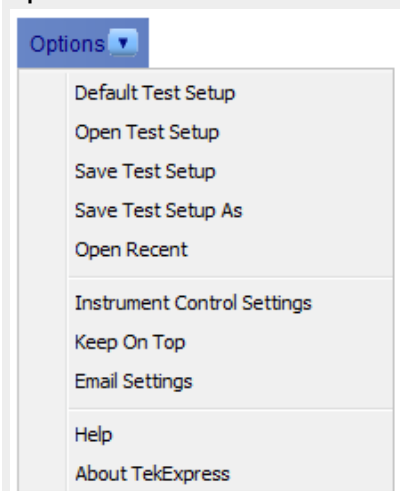
| Item | Description |
|------------------------------|---|
| Options menu | Opens the Options menu for access to global controls |
| Panels | Visual frames with sets of related options |
| Command buttons | Buttons that initiate an immediate action such as the Start, Stop, Pause, Continue, and Clear command buttons |
| Start button | <p>Start</p>  <p>Starts measurement acquisition and analysis.</p> |
| Stop button | <p>Stop</p>  <p>Stops (aborts) the current test run.</p> |
| Pause \ Continue button | <p>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>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.</p> |
| Clear Log | <p>Clear Log</p>  <p>This button is available only on the Status panel.</p> |
| Save | <p>Save...</p>  <p>This button is available only on the Status panel.</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

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

The Options menu has the following selections:

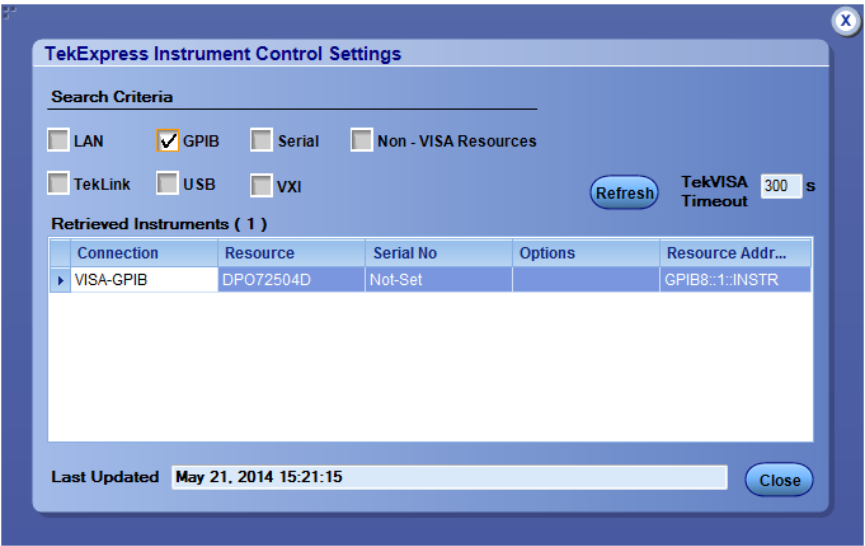
Options Menu



| Menu | Function |
|------------------------------------|--|
| Default Test Setup | Opens an untitled test setup with defaults selected |
| Open Test Setup | Opens a saved test setup |
| Save Test Setup | Saves the current test setup 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 |
| <i>Instrument control settings</i> | 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 C-PHY Tx utility on top of other open windows on the desktop |
| <i>Email settings</i> | Use to configure email options for test run and results notifications |
| 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 for your C-PHY Tx installation ■ Provides a link to the Tektronix Web site |

See also. [Application controls](#)

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](#) for details.

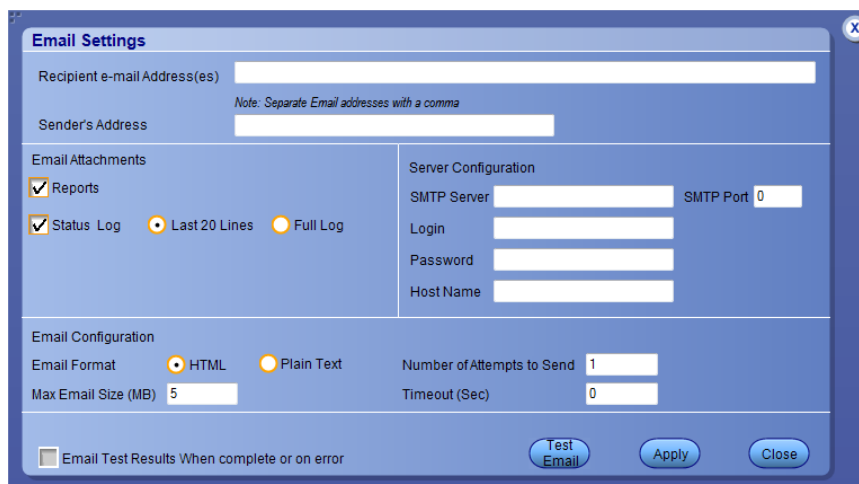
See also. [View connected instruments](#)

[Options menu overview](#)

Email settings. Use the Email Settings dialog box to enable C-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.

The image shows the 'Email Settings' dialog box. It has a title bar with a close button. The dialog is divided into several sections. The top section has two text input fields: 'Recipient e-mail Address(es)' and 'Sender's Address'. Below the first field is a note: 'Note: Separate Email addresses with a comma'. The middle section is split into two columns. The left column is titled 'Email Attachments' and contains three checked checkboxes: 'Reports', 'Status Log', and 'Last 20 Lines'. The right column is titled 'Server Configuration' and contains four text input fields: 'SMTP Server', 'SMTP Port' (with a value of 0), 'Login', 'Password', and 'Host Name'. The bottom section is titled 'Email Configuration' and contains two rows of settings. The first row has 'Email Format' with radio buttons for 'HTML' (selected) and 'Plain Text'. The second row has 'Max Email Size (MB)' with a value of 5 and 'Number of Attempts to Send' with a value of 1. Below these are 'Timeout (Sec)' with a value of 0 and a checkbox for 'Email Test Results When complete or on error' which is currently unchecked. At the bottom right are three buttons: 'Test Email', 'Apply', and 'Close'.

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](#)

[Options menu overview](#)

[Set test notification preferences](#)

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.

Application panels

Application panel overview

Panels group related configuration, test and results settings.

The TekExpress C-PHY Tx Essentials panels are:

Table 5: Application panels

| Panel name | Description |
|-------------------------|---|
| Setup | Set the DUT, test, acquisition, and report parameters. |
| Status | View the progress and analysis status of the selected tests, and view test logs. |
| Results | View a summary of test results and select results viewing preferences. |
| Reports | 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](#)

[About setting up tests](#)

Setup panel

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

[Set the DUT parameters](#)

[Select test\(s\)](#)

[Set lane acquisition source](#)

[Configure tests](#)

[Select test notification preferences](#)

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.

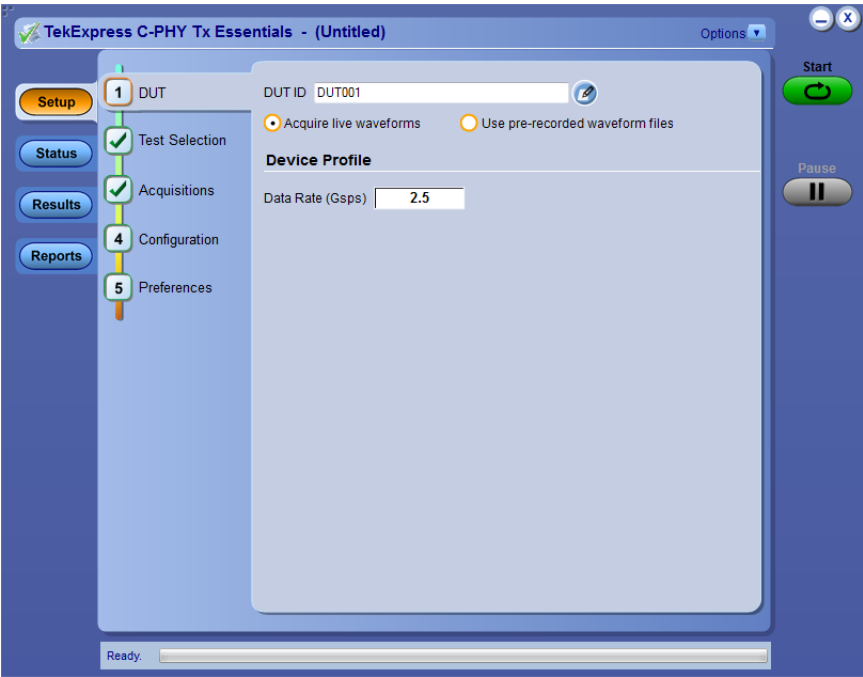


Figure 1: DUT tab settings


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

Table 6: DUT tab settings

| 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. |
| Data Rate | Enter the data rate to include in the tests |
| | NOTE. Values should be between 0.05 - 4.5 |

See also. [About setting up tests](#)

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.

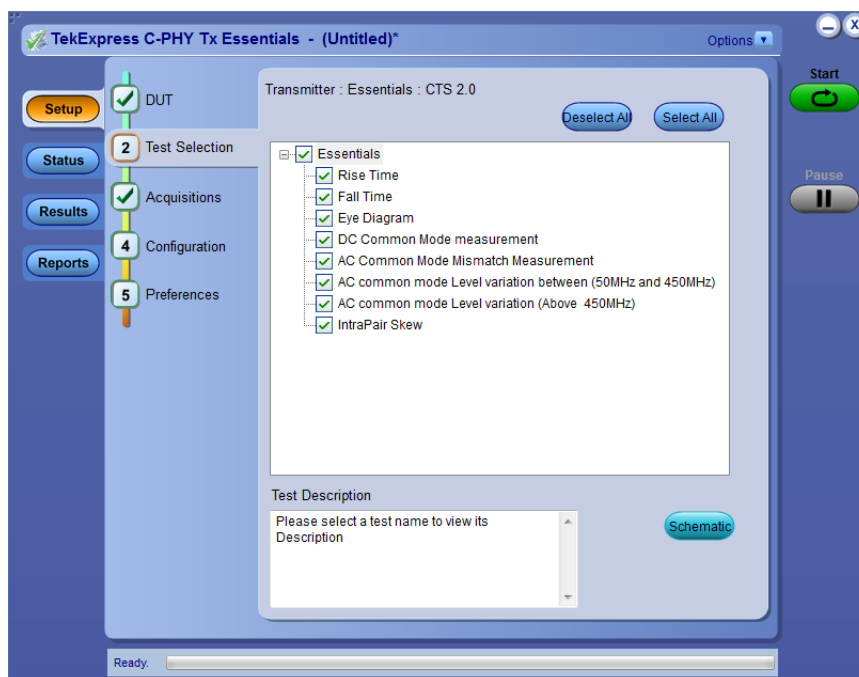


Table 7: Test Selection tab settings

| Control or field | Description |
|------------------------|--|
| Deselect All | Deselects (clears) all tests. |
| 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 (Rise Time, Fall Time, Eye Diagram, DC Common Mode, AC common mode mismatch, AC common mode level variation between 50 MHz and 450 MHz, AC common mode level variation above 450 MHz, IntraPair Skew). |
| 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. |

See also. [About setting up tests](#)

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.

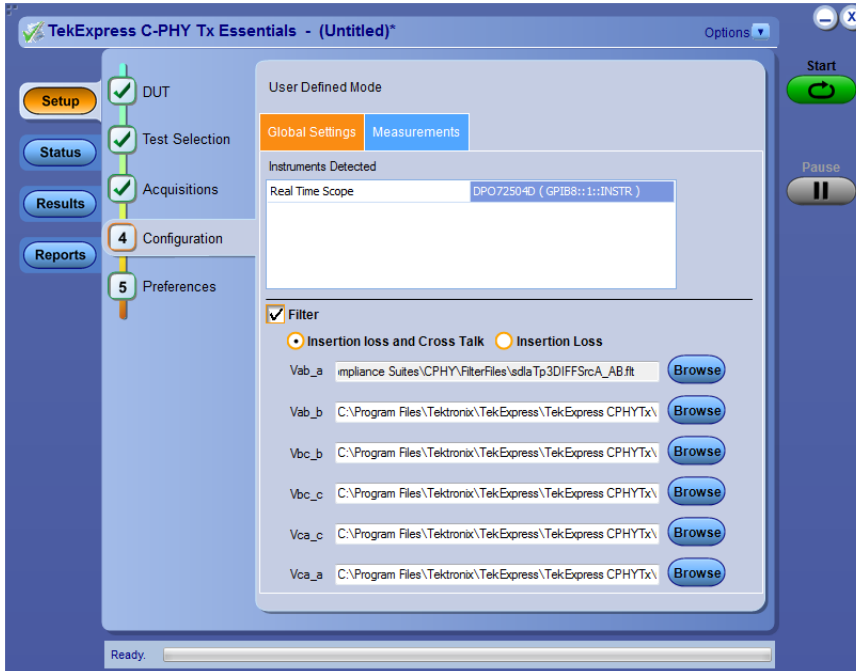


Figure 2: Configuration tab - Global Settings

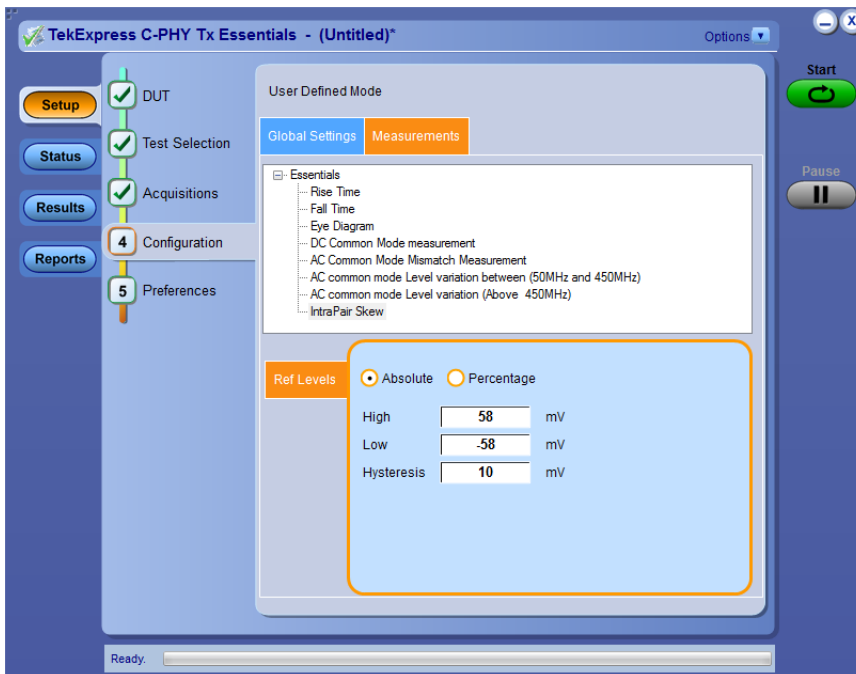


Figure 3: Configuration tab - Measurements

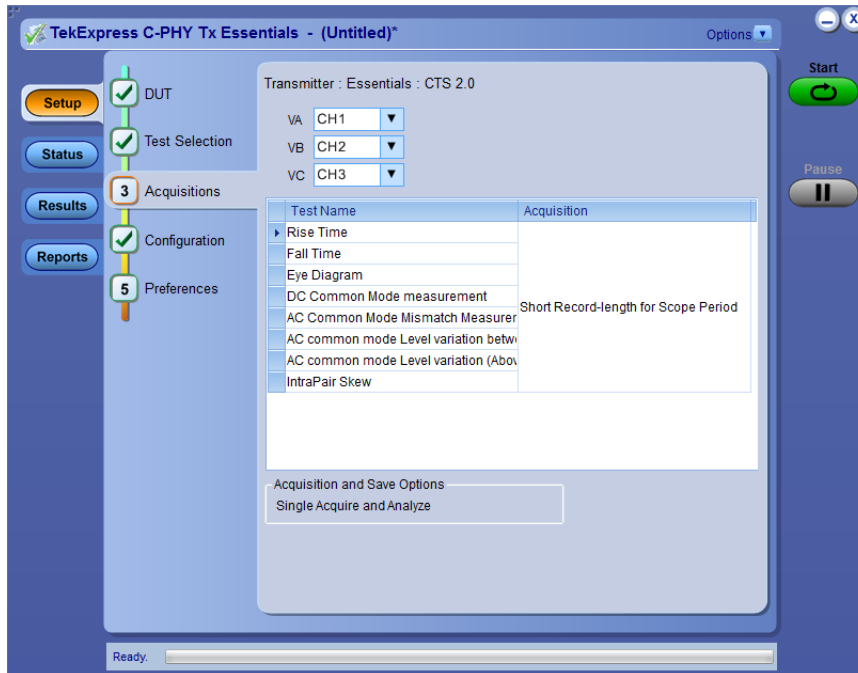
Table 8: Test configuration settings

| Control or field | Description |
|-------------------|---|
| 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. |
| 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. 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> |
| Filter | <ul style="list-style-type: none"> ■ Insertion Loss and Cross Talk ■ Insertion Loss |
| Measurements | <p>Lists all tests which are selected in the Test Selection tab.</p> <p>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> |

See also. [Configure tests](#)

[About running tests](#)

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. Select the Channels for VA, VB, and VC signals.

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

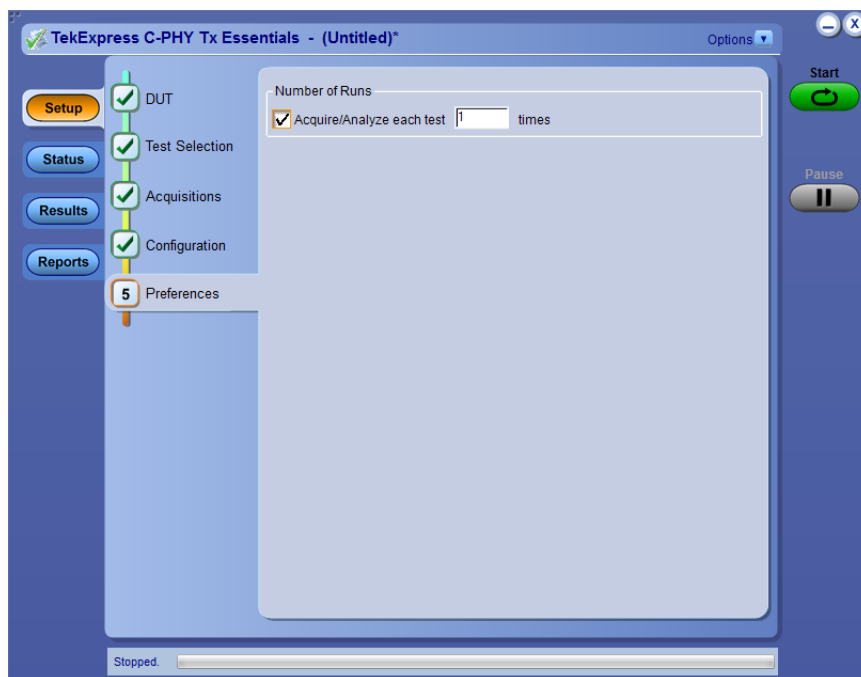


Figure 4: Preferences tab

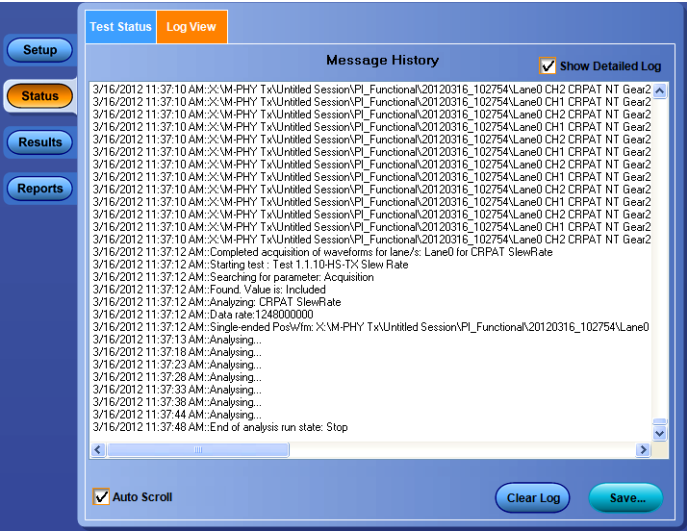
1. Click **Setup > Preferences**.
2. Select the **Acquire/Analyze each test** check box to select the lanes.
3. Specify the number of times, each selected test to run.

See also. [About setting up tests](#)

[Select report options](#)

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 9: 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 startedIn progressTestingCompletedAborted |

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

Table 10: 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](#)

Results panel

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](#).

| Test Name | Details | Value | Units | Pass/Fail | Margin |
|-----------|---------------------|----------|-------|-----------|----------|
| Rise Time | RiseTime(S-W) of AB | 204.1108 | ps | Pass | 174.1108 |
| Rise Time | RiseTime(W-S) of AB | 204.1851 | ps | Pass | 174.1851 |
| Rise Time | RiseTime(W-W) of AB | 202.5722 | ps | Pass | 172.5722 |
| Rise Time | RiseTime(S-S) of AB | 201.0968 | ps | Pass | 171.0968 |
| Rise Time | RiseTime(S-W) of BC | 201.3647 | ps | Pass | 171.3647 |
| Rise Time | RiseTime(W-S) of BC | 201.4214 | ps | Pass | 171.4214 |
| Rise Time | RiseTime(W-W) of BC | 196.0394 | ps | Pass | 166.0394 |
| Rise Time | RiseTime(S-S) of BC | 198.7608 | ps | Pass | 168.7608 |
| Rise Time | RiseTime(S-W) of CA | 230.7740 | ps | Pass | 200.7740 |
| Rise Time | RiseTime(W-S) of CA | 226.3131 | ps | Pass | 196.3131 |
| Rise Time | RiseTime(W-W) of CA | 222.3979 | ps | Pass | 192.3979 |
| Rise Time | RiseTime(S-S) of CA | 225.9991 | ps | Pass | 195.9991 |
| Fall Time | FallTime(S-W) of AB | 205.8499 | ps | Pass | 175.8499 |
| Fall Time | FallTime(W-S) of AB | 203.7134 | ps | Pass | 173.7134 |

View test results. When a test finishes, the application switches to the [Results panel](#), 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:

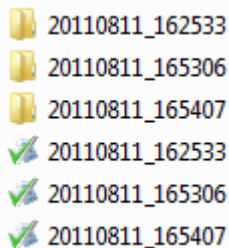
- To expand all tests listed, select **View Results Details** from the Preferences menu in the upper right corner.
- To expand and collapse tests, click the plus and minus buttons.
- To collapse all expanded tests, select **Preferences > View Results Summary**.
- To remove or restore the Pass/Fail column,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

View test-related files. Files related to tests are stored in the My TekExpress\C-PHY Tx Essentials folder. In the C-PHY Tx Essentials 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 C-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.



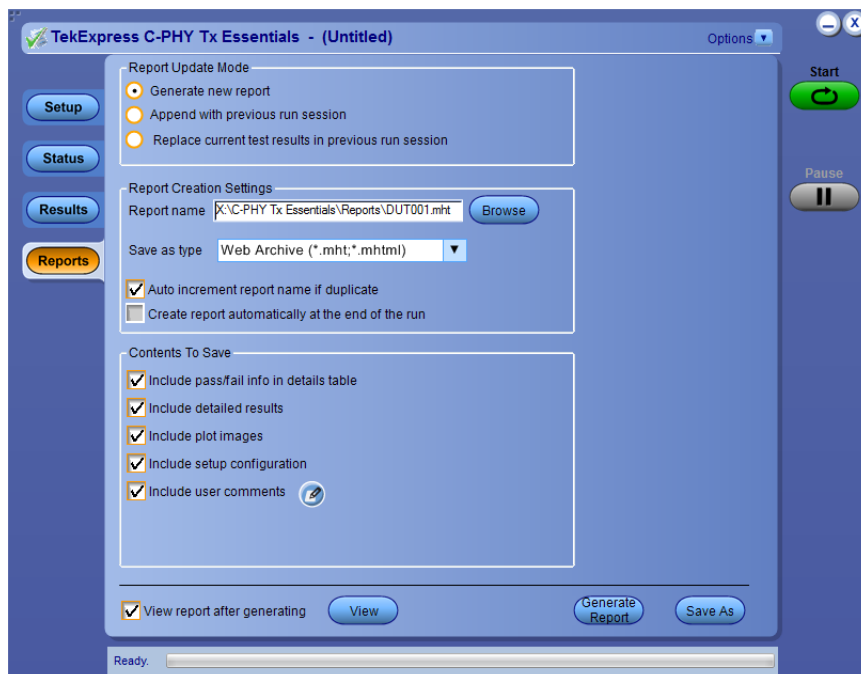
The first time you run a new, unsaved session, the session files are stored in the Untitled Session folder located at . . \My TekExpress\C-PHY Tx Essentials. When you name and save the session, the files are placed in a folder with the name that you specify.

See also. [File name extensions](#)

Reports panel

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



Select report options. Use the [Reports panel](#) 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 set your 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 11: Report options

| Setting | Description |
|--|--|
| Generate new report | Creates a new report. |
| Append with previous run session | Appends the last run session and the previous run session. |
| Replace current test results in previous run session | Replaces the current test results in the previous run session. Newly added tests results are appended to the end of the report. |
| Report Name | <p>Displays the name and location from which to open a report. The default location is at \My TekExpress\ C-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\C-PHY Tx \DUT001_Test_211.mht.</p> <p>NOTE. You cannot change the file location using the Browse button.</p> <p>Open an existing report.</p> <p>Click Browse, to 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. 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. The application will not automatically do this when the file is saved.</p> |

| Setting | Description |
|---|---|
| Auto increment report name if duplicate | Automatically increment the name of the report file if the application finds a file with the same name as the one being generated. For example: DUT001, DUT002, DUT003. This option is enabled by default. |
| Create report automatically at the end of the run | Creates the report automatically at the end of the run. |
| Include Pass/Fail Results Summary | 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 . |
| Include Detailed Results | Includes parameter limits, execution time, and test-specific comments generated during the test. |
| Include Plot Images | Includes waveform plots in the report. |
| Include Setup Configuration | 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. |
| Include User Comments | Select to include any comments about the test that you or another user added in the DUT tab of the Setup panel. Comments appear in the Comments section under the summary box at the beginning of each report. |
| View Report After Generating | Automatically opens the report in your Web browser when the test completes. This option is selected by default. |

See also. [View a report](#)

[About setting up tests](#)

View a report. The application automatically generates a report when test analysis is completed and displays the report in the 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](#).

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.

| <div> <div>Tektronix</div> <div> TekExpress Automated Test Solution TekExpress C-PHY Tx </div> </div> | | | |
|---|---|--------------------------|----------------|
| Setup Information | | | |
| DUT ID | DUT001 | Scope Model | DPO70604C |
| Device Type | Transmitter | Scope Serial Number | Q130114 |
| TekExpress Version | C-PHY Tx: 1.0.0.91 Framework: 3.0.1.88 | SPC, Factory Calibration | 7.3.0 Build 9* |
| Execution Mode | Live | Scope F/W Version | PASS, PASS |
| Overall Compliance Mode | No | DPOJET Version | 6.2.1.8 |
| Over All Test Result | Pass | Probe1 Model | 1X |
| Overall Execution Time | 0:02:15 | Probe1 Serial Number | N/A |
| | | Probe2 Model | TCA292D |
| | | Probe2 Serial Number | N/A |
| | | Probe3 Model | 1X |
| | | Probe3 Serial Number | N/A |
| DUT COMMENT: | General Comment - CPHY_Tx | | |

DUT Comment

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 DUT Comment section directly below the summary box.

Test Name Summary Table

This table lists the tests results. The test results for the selected tests are displayed.

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, and so on).

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

| Test Name Summary Table | |
|---|-------------|
| Rise Time | Pass |
| Fall Time | Pass |
| Eye Diagram | Pass |
| DC Common Mode measurement | Pass |
| AC Common Mode Mismatch Measurement | Pass |
| AC common mode Level variation between (50MHz and 450MHz) | Pass |
| AC common mode Level variation (Above 450MHz) | Pass |
| IntraPair Skew | Informative |

See also. [View test results](#)

[View test-related files](#)

Setting up and configuring tests

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

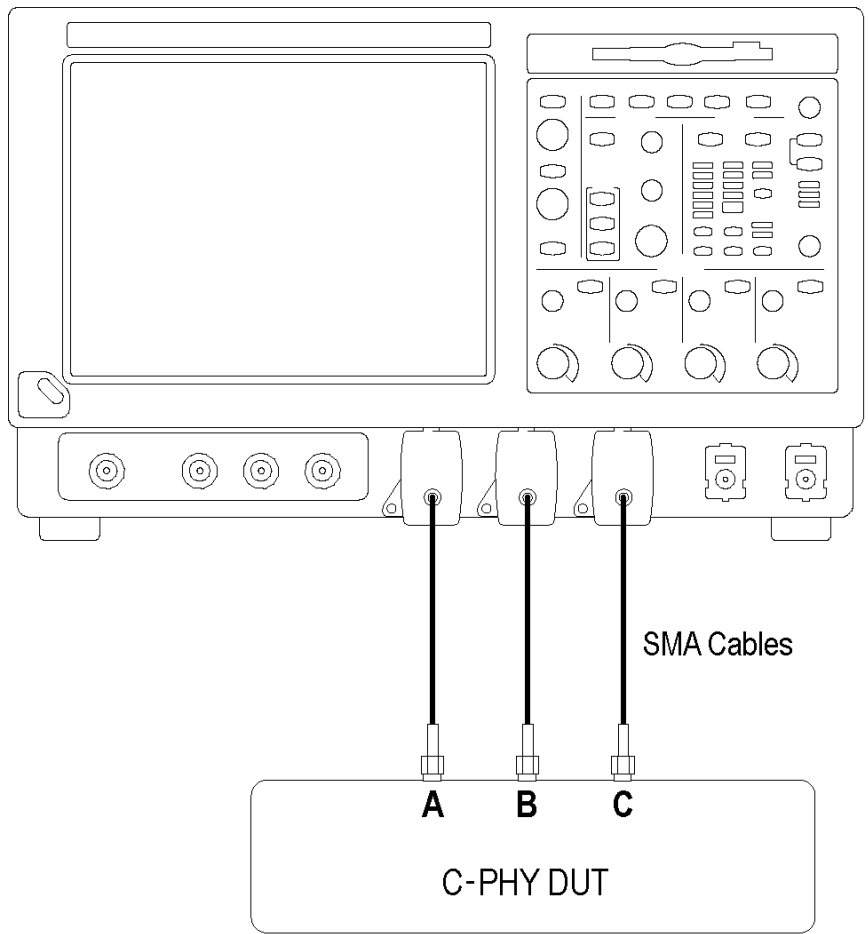
[About test setups](#)

[Before you click Start](#)

[About running tests](#)

Instrument connection setup

The following diagram shows how to connect the DUT to the oscilloscope for all the C-PHY Tx Essentials measurements.



See also

[Minimum system requirements](#)

[View connected instruments](#)

[About setting up tests](#)

View connected instruments

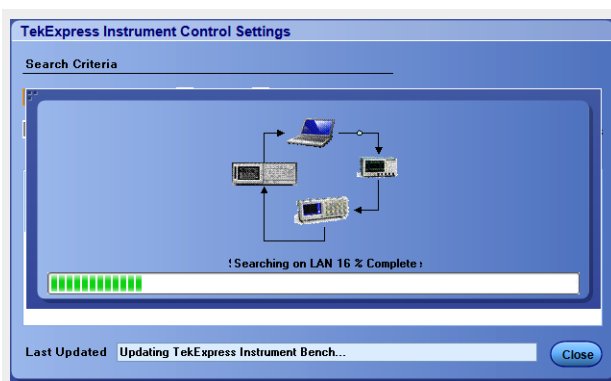
Use the Instrument Control Settings dialog box to view or search for connected instruments required for the tests. The application uses TekVISA to discover the connected instruments.

To refresh the list of connected instruments:

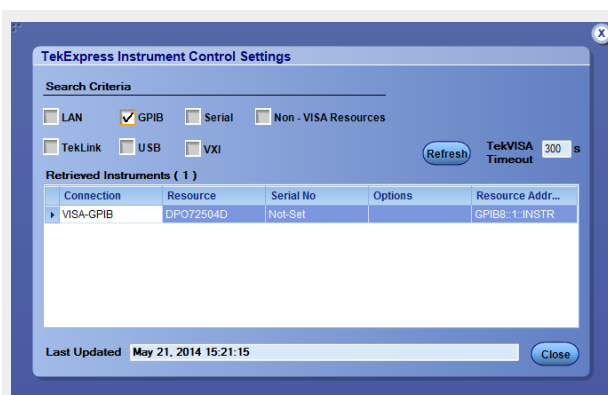
1. From the Options menu, select **Instrument Control Settings**.
2. In the Search Criteria section of the Instrument Control Settings dialog box, select the connection types of the instruments to search for.

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](#)

[Instrument connection setup](#)

Test setup overview

A test setup includes configuration parameters and report options. Use the options in the Setup panel and [Reports panel](#) to select and configure tests.

1. [Select the DUT parameters.](#)
2. [Select one or more tests.](#)
3. [Configure test parameters.](#)
4. [Select test notification preferences.](#)
5. [Select report options.](#)

See also

[Test setups overview](#)

[Pre-run checklist](#)

[Before you click Start](#)

[About running tests](#)

Running tests

About running tests

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

See also

[Configure test parameters](#)

[About setting up tests](#)

[Before you click Start](#)

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

- \Program Files\Tektronix\TekExpress\TekExpress CPHYTX

NOTE. The Program Files folder for Windows 7 is for 64-bit.

- \My Documents\My TekExpress\C-PHY Tx Essentials
- \My Documents\My TekExpress\C-PHY Tx Essentials\Untitled Session

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

**CAUTION.**

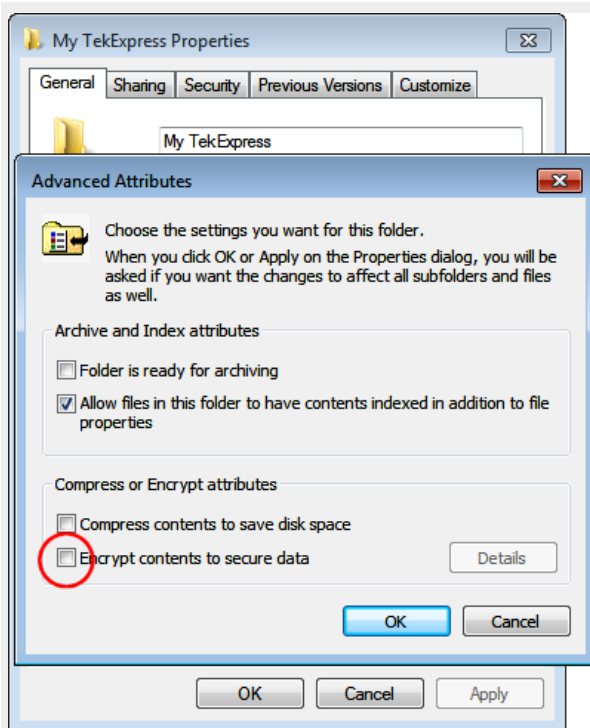
Do not directly edit or modify any of the session files or folders because this may result in loss of data or corrupted session files. Each session has multiple files associated with it. When you save a session, the application creates a .TekX file, and a folder named for the session that contains associated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

2. [Map my TekExpress folder](#) 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](#) before you run a test.

See also

[View test-related files](#)

[Application directories and file types](#)

[File name extensions](#)

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](#).

On the oscilloscope:

- Make sure that all the required instruments are properly warmed up (approximately 20 minutes).
- Perform Signal Path Compensation (SPC).
 1. On the oscilloscope main menu, select the **Utilities** menu.
 2. Select **Instrument Calibration** and run the SPC utility.
- Perform deskew on any cables.
- Make sure that all the required instruments are properly warmed up (approximately 20 minutes).
- Perform Signal Path Compensation (SPC).
 1. On the oscilloscope main menu, select the **Utilities** menu.
 2. Select **Instrument Calibration** and run the SPC utility.
- Perform deskew on any cables.

In the C-PHY Tx Essentials application:

- Verify that the application is able to find the instrument. If it cannot, [perform a search for connected instruments](#).
1. Select **Setup > Test Selection**. Select any test and then click **Configure**.
 2. In the Configuration section, click **Global Settings**.
 3. 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.

See also

[Instrument connection setup](#)

Saving and recalling test setups

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 C-PHY Tx Essentials.

See also

[About setting up tests](#)

[Save a test setup](#)

[Recall a saved test setup](#)

[Create a new test setup based on an existing one](#)

Save a test setup

Save a test setup before or after running a test using the parameters you want to save. 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](#).
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:\C-PHY Tx Essentials\<session_name>.

See also

[About setting up tests](#)

[Configure test parameters](#)

[Test setup overview](#)

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](#)

[Create a new test setup based on an existing one](#)

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](#)

TekExpress programmatic interface

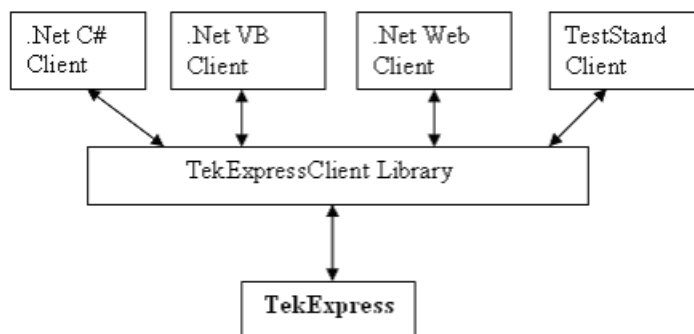
Programmatic interface overview

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](#)

[Remote proxy object](#)

[Client proxy object](#)

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 *Ildglib.dll* and *IRemoteInterface.dll*.
- *Ildglib.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.
- *Ildglib.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*, *Ildglib.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.

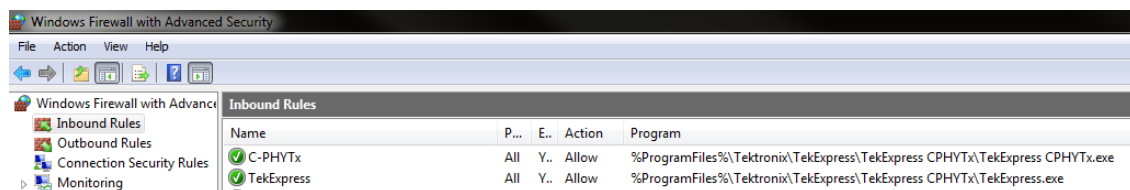
If any other client tries to access a server that is locked, it will receive a notification that the server is locked by another client.

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.

To enable remote access

To access and remotely control an instrument using the TekExpress programmatic interface, you need to change specific firewall settings as follows:

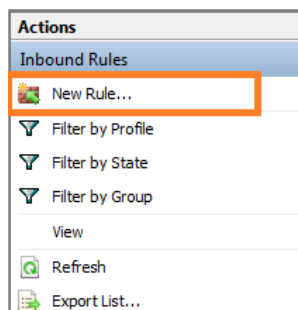
1. Access the Windows Control Panel and open the Windows Firewall tool (**Start > Control Panel > All Control Panel Items > Windows Firewall**).
2. Click **Advance Settings > Inbound Rules**.
3. Scroll through the Inbound Rules list to see if the following items (or with a similar name) are shown:
 - TekExpress C-PHY Tx Essentials
 - TekExpress



4. If both items are shown, you do not need to set up any rules. Exit the Windows Firewall tool.
5. If one or both are missing, use the following procedure to run the New Inbound Rule Wizard and add these executables to the rules to enable remote access to the TekExpress application.
6. On the client side include the Client application.exe through which TekExpress application is remotely controlled. For example if the application is controlled using python scripts then "ipy64.exe" should be included as part of Inbound rules.

Run the New Inbound Rule Wizard

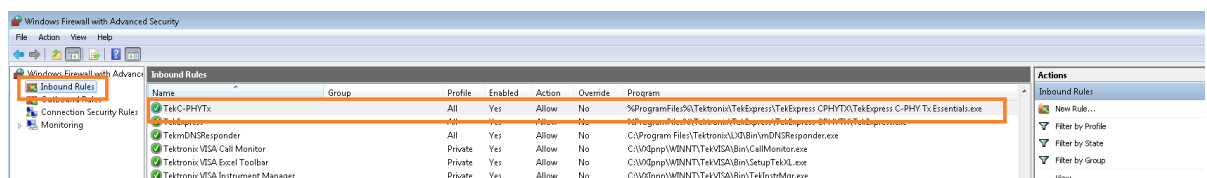
1. Click on **New Rule** (in Actions column) to start the **New Inbound Rule Wizard**.



2. Verify that **Program** is selected in the Rule Type panel and click **Next**.
3. Click **Browse** in the Program panel and navigate to and select one of the following TekExpress applications (depending on the one for which you need to create a rule):
4. TekExpress C-PHY Tx Essentials.exe
5. TekExpress.exe

NOTE. See [Application directories and content](#) for the path to the application files.

6. Click **Next**.
7. Verify that **Allow the connection** is selected in the Action panel and click **Next**.
8. Verify that all fields are selected (**Domain**, **Private**, and **Public**) in the Profile panel and click **Next**.
9. Use the fields in the Name panel to enter a name and optional description for the rule. For example, a name for the TekExpress C-PHY Tx Essentials application could be **TekExpress C-PHY Tx Essentials**. Add description text to further identify the rule.
10. Click **Finish** to return to the main Windows Firewall screen.
11. Scroll through the Inbound Rules list and verify that the list shows the rule that you just entered.



12. Repeat steps 1 through 11 to enter the other TekExpress executable if it is missing from the list. Enter **TekExpress PI** as the name.

13. Scroll through the Inbound Rules list and verify that the list shows the rule that you just entered.
14. Exit the Windows Firewall tool.

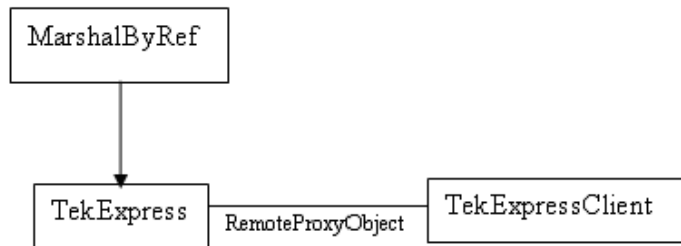
To use the remote access:

1. Obtain the IP address of the instrument on which you are running TekExpress C-PHY Tx Essentials. For example, 134.64.235.198.
2. On the PC from which you are accessing the remote instrument, use the instrument IP address as part of the C-PHY Tx Essentials PI code to access that instrument. For example:

```
object obj = piClient.Connect("134.64.235.198",out clientid);
```

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.

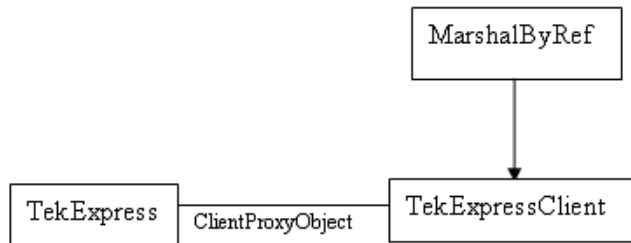
For example,

```
//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:

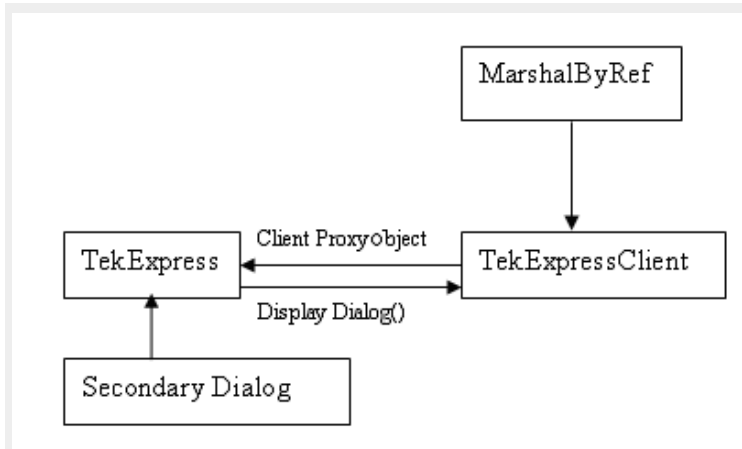
- To get the secondary dialog messages from the server.
- To get the file transfer commands from the server while transferring the report.

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.

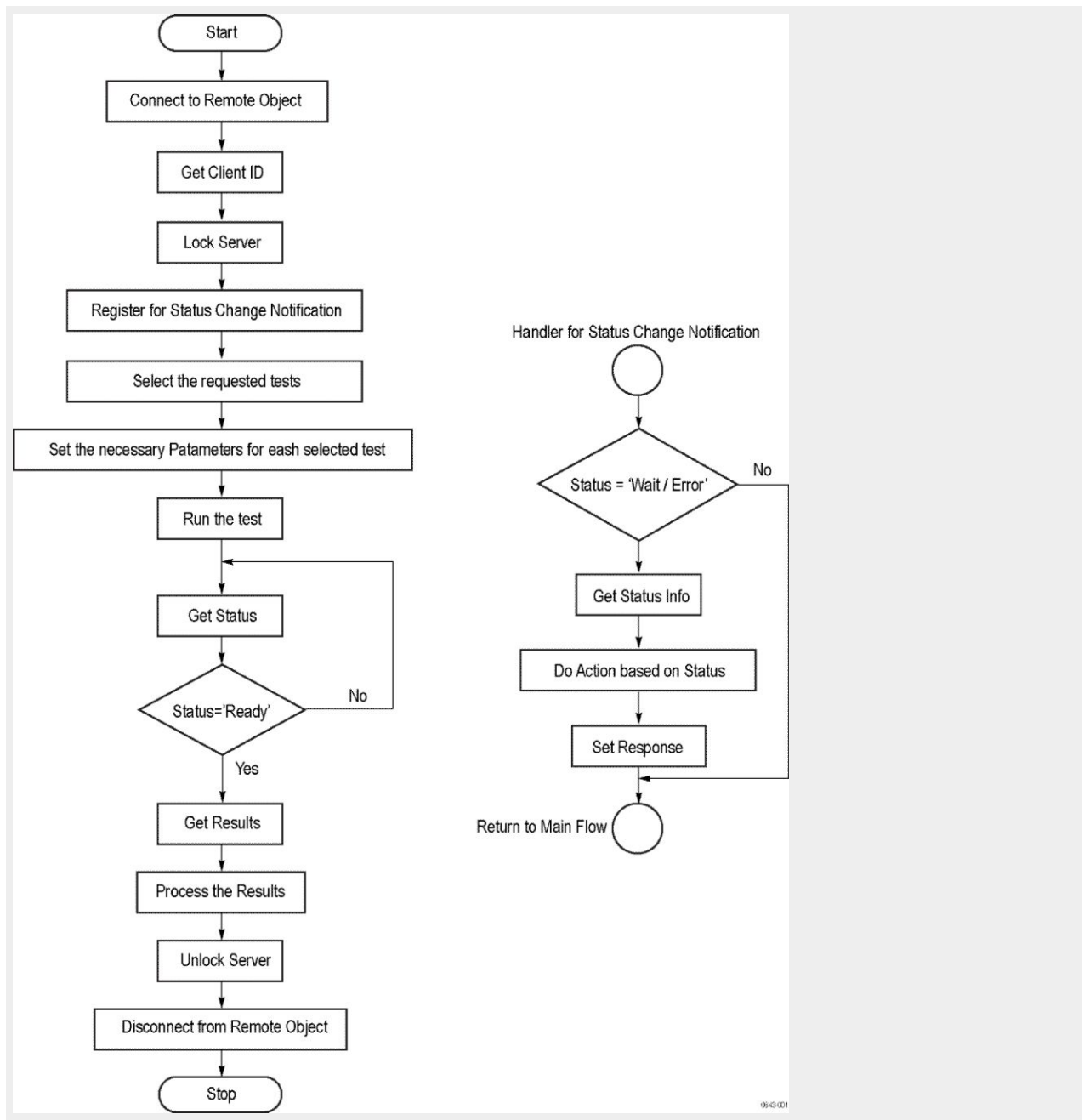
File Transfer Events

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



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](#).

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

[Program remote access code example](#)

Program remote access code example

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

Table 12: Remote access code example

| Task | Code |
|--------------------------------|---|
| Start the application | |
| Connect through an IP address. | <code>m_Client.Connect("localhost") 'True or False 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.selectsingleTest(clientID, "Transmitter", "Essentials", "", "Rise Time", true)</code> |

| Task | Code |
|-----------------------------------|--|
| Select an amplitude | <code>mClient.SetGeneralParameter(clientID, "Transmitter", "Essentials", " Rise Time ", "Amplitude\$Large Amplitude")</code> |
| Select a termination | <code>mClient.SetGeneralParameter(clientID, "Transmitter", "Essentials", " Rise Time ", "Termination\$Unterminated")</code> |
| Run with set configurations | <code>m_Client.Run(clientID)</code> |
| Wait for the test to complete. | <code>Do Thread.Sleep(500) m_Client.Application_Status(clientID) Select Case status 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) End Select Loop Until status = "Ready"</code> |
| Save results | <code>'Save all results values from folder for current run m_Client.TransferResult(clientID, logDirname)</code> |
| Unlock the server | <code>m_Client.UnlockServer(clientID)</code> |
| Disconnect from server | <code>m_Client.Disconnect()</code> |
| Exit the application | |

C-PHY Tx Essentials 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](#)

[Lock the server](#)

[Disable popup messages](#)

[Set or get the DUT ID](#)

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

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

[Select a test](#)

[Select a channel](#)

[Run with set configurations or stop the run operation](#)

[Handle error codes](#)

[Get or set the timeout value](#)

[After the test is complete](#)

Wait for the test to complete

Save, recall, or query a saved session

Unlock the server

Disconnect from the server

Connect through an IP address

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

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. 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. 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 | This method locks the server. Note 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) |

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. 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.

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 bool _verbose | This method sets the verbose mode to either true or false. When the value is set to true, any message boxes that appear during the application will be routed to the client machine that is controlling TekExpress. When the value is set to false, all the message boxes are shown on the server machine. | String that gives the status of the operation after it has been performed When Verbose mode is set to true, the return value is "Verbose mode turned on. All dialog boxes will be shown to client". When Verbose mode is set to false, the return value is "Verbose mode turned off. All dialog boxes will be shown to server". | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string Verbose mode is turned on return=m_Client.SetVerboseMode(clientID, true) Verbose mode is turned off returnval=m_Client.SetVerboseMode(clientID, false) |

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

bool _verbose

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

NOTE. The Fail condition for 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 string dutName | This method changes the DUT ID of the setup. The client must provide a valid DUT ID. | String that gives the status of the operation after it has been performed Return value is "DUT Id Changed" on success | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string return=m_Client.SetDutId(clientID,desiredDutId) Note |
| GetDutId() | string clientID string dutId | This method gets the DUT ID of the current setup. | String that gives the status of the operation after it has been performed | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string return=m_Client.GetDutId(clientID, out DutId) |

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

string dutName

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

string dutId

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

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

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

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 string device string suite string test string parameterString | This method sets the number of video lanes for the selected measurement. NOTE. Use this command to select a lane, channel, or source type. | String that gives the status of the operation after it has been performed The return value is "" (an empty String) on success. | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string |
| SetAnalyzeParameter() | string clientID string device string suite string test string parameterString | This method sets the configuration parameters in the Analyze panel of the Configuration Panel dialog box for a suite or measurement. | String that gives the status of the operation after it has been performed The return value is "" (an empty String) on success. | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string |
| SetAcquireParameter() | string clientID string device string suite string test string parameterString | This method sets the configuration parameters in the Acquire panel of the Configuration Panel dialog box for a suite or measurement. | String that gives the status of the operation after it has been performed The return value is "" (an empty String) on success. | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnVal = remoteObject.SetAcquireParameter(id, device, suite, test, parameterString) if ((OP_STATUS) returnVal != OP_STATUS.SUCCESS) return CommandFailed(returnVal) |

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

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

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

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

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

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

string device

| Name | Type | Direction | Description |
|--------|--------|-----------|----------------------------------|
| device | string | IN | Specifies the name of the device |

string suite

| Name | Type | Direction | Description |
|-------|--------|-----------|---------------------------------|
| suite | string | IN | Specifies the name of the suite |

string test

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

string parameterString

| Name | Type | Direction | Description |
|-----------------|--------|-----------|-----------------------------|
| parameterString | string | IN | Selects or deselects a test |

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 |

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 |

Query the configuration parameters for a suite or measurement

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

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

string device

| Name | Type | Direction | Description |
|--------|--------|-----------|----------------------------------|
| device | string | IN | Specifies the name of the device |

string suite

| Name | Type | Direction | Description |
|-------|--------|-----------|---------------------------------|
| suite | string | IN | Specifies the name of the suite |

string test

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

string parameterString

| Name | Type | Direction | Description |
|-----------------|--------|-----------|-----------------------------|
| parameterString | string | IN | Selects or deselects a test |

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 |

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 |

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

NOTE. The Fail condition for 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", "Essentials", "Rise Time", "Scope Channel for Trigger$CH1")
```

Select a test

| Command name | Parameters | Description | Return value | Example |
|--------------|--|---|--|---|
| SelectTest() | string clientID string device string suite string test bool isSelected | This method selects or deselects a specified test. If this Setting parameter is set to true, you can select a measurement. If this Setting parameter is set to false, you can deselect a measurement. | String that displays the status of the operation after it has been performed The return value is "" (an empty String) on success. | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string Select test example |

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

string device

| Name | Type | Direction | Description |
|--------|--------|-----------|----------------------------------|
| device | string | IN | Specifies the name of the device |

string suite

| Name | Type | Direction | Description |
|-------|--------|-----------|---------------------------------|
| suite | string | IN | Specifies the name of the suite |

string test

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

bool isSelected

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

NOTE. The Fail condition for 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", "Essentials", "Rise Time", true)
```

Select a channel

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

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

string device

| Name | Type | Direction | Description |
|--------|--------|-----------|----------------------------------|
| device | string | IN | Specifies the name of the device |

string suite

| Name | Type | Direction | Description |
|-------|--------|-----------|---------------------------------|
| suite | string | IN | Specifies the name of the suite |

string test

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

string parameterString

| Name | Type | Direction | Description |
|-----------------|--------|-----------|-----------------------------|
| parameterString | string | IN | Selects or deselects a test |

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

string device

| Name | Type | Direction | Description |
|--------|--------|-----------|----------------------------------|
| device | string | IN | Specifies the name of the device |

string suite

| Name | Type | Direction | Description |
|-------|--------|-----------|---------------------------------|
| suite | string | IN | Specifies the name of the suite |

string test

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

string parameterString

| Name | Type | Direction | Description |
|-----------------|--------|-----------|-----------------------------|
| parameterString | string | IN | Selects or deselects a test |

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

string device

| Name | Type | Direction | Description |
|--------|--------|-----------|----------------------------------|
| device | string | IN | Specifies the name of the device |

string suite

| Name | Type | Direction | Description |
|-------|--------|-----------|---------------------------------|
| suite | string | IN | Specifies the name of the suite |

string test

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

string parameterString

| Name | Type | Direction | Description |
|-----------------|--------|-----------|-----------------------------|
| parameterString | string | IN | Selects or deselects a test |

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 data rate

```
returnval = mClient.SetGeneralParameter(clientID, "Transmitter", "Essentials", "Data Rate (Gbps)$4.5")
```

Configure the selected measurement

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

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

string device

| Name | Type | Direction | Description |
|--------|--------|-----------|----------------------------------|
| device | string | IN | Specifies the name of the device |

string suite

| Name | Type | Direction | Description |
|-------|--------|-----------|---------------------------------|
| suite | string | IN | Specifies the name of the suite |

string test

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

string parameterString

| Name | Type | Direction | Description |
|-----------------|--------|-----------|-----------------------------|
| parameterString | string | IN | Selects or deselects a test |

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 | Runs the selected tests Note 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 | Stops the currently running tests Note | String that gives the status of the operation after it has been performed The return value is "Stopped..." on success. | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.Stop(clientID) |

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

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

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

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 | 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 string time | 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) |

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

string time

| Name | Type | Direction | Description |
|------|--------|-----------|---|
| time | string | IN | The time in seconds 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.

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

NOTE. *The Fail condition for 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 | This method gets the status of the server application. The states are Ready, Running, Paused, Wait, or Error. | 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 out string[] status | 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 |
| GetCurrentStateInfo() <i>NOTE. This command is used when the application is running and is in the wait or error state.</i> | string clientID out string WaitingMsbBxCaption out string WaitingMsbBxMessage out string[] WaitingMsbBxButtontexts | This method gets the additional information of the states when the application is in Wait or Error state. Except client ID, all the others are Out parameters. | This command does not return any value. This function populates the Out parameters that are passed when invoking this function. | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL mClient.GetCurrentStateInfo(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxButtontexts) |
| SendResponse() <i>NOTE. This command is used when the application is running and is in the wait or error state.</i> | string clientID out string WaitingMsbBxCaption out string WaitingMsbBxMessage string WaitingMsbBxResponse | After receiving the additional information using the method GetCurrentStateInfo(), the client can decide which response to send and then send the response to the application using this function. The response should be one of the strings that was received earlier as a string array in the GetCurrentStateInfo function. The _caption and _message should match the information received earlier in the GetCurrentStateInfo function. | This command does not return any value. | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL mClient.SendResponse(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxResponse) |

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

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

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

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

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

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

out string[] status

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

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

out string 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 |

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

out string 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[] 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 button Texts) |

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

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 string device string suite string test | <p>This method gets the pass or fail status of the measurement after test completion.</p> <p>NOTE. Execute this command after completing the measurement.</p> | <p>String that gives the status of the operation after it has been performed</p> <p>Returns the pass or fail status in the form of a string</p> | <p>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.</p> <p>returnval as string</p> <p>returnval=m_Client.GetPass-FailStatus(clientID, device, suite, test)</p> |
| GetResultsValue() | string clientID string device string suite string test string parameterString | This method gets the result values of the measurement after the run. | <p>String that gives the status of the operation after it has been performed</p> <p>Returns the result value in the form of a string</p> | <p>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.</p> <p>returnval as string</p> <p>returnval=m_Client.GetResultsValue(clientID, device, suite, test, "Measured Value")</p> |
| GetResultsValueForSubMeasurements() | string clientID string device string suite string test string parameterString int rowNr | This method gets the result values for individual sub-measurements after the run. | <p>String that gives the status of the operation after it has been performed</p> <p>Returns the result value in the form of a string</p> | <p>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.</p> <p>returnval as string</p> |

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

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

string device

| Name | Type | Direction | Description |
|--------|--------|-----------|----------------------------------|
| device | string | IN | Specifies the name of the device |

string suite

| Name | Type | Direction | Description |
|-------|--------|-----------|---------------------------------|
| suite | string | IN | Specifies the name of the suite |

string test

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

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

string device

| Name | Type | Direction | Description |
|--------|--------|-----------|----------------------------------|
| device | string | IN | Specifies the name of the device |

string suite

| Name | Type | Direction | Description |
|-------|--------|-----------|---------------------------------|
| suite | string | IN | Specifies the name of the suite |

string test

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

string parameterString

| Name | Type | Direction | Description |
|-----------------|--------|-----------|-----------------------------|
| parameterString | string | IN | Selects or deselects a test |

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

string device

| Name | Type | Direction | Description |
|--------|--------|-----------|----------------------------------|
| device | string | IN | Specifies the name of the device |

string suite

| Name | Type | Direction | Description |
|-------|--------|-----------|---------------------------------|
| suite | string | IN | Specifies the name of the suite |

string test

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

string parameterString

| Name | Type | Direction | Description |
|-----------------|--------|-----------|-----------------------------|
| parameterString | string | IN | Selects or deselects a test |

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

string device

| Name | Type | Direction | Description |
|--------|--------|-----------|----------------------------------|
| device | string | IN | Specifies the name of the device |

string suite

| Name | Type | Direction | Description |
|-------|--------|-----------|---------------------------------|
| suite | string | IN | Specifies the name of the suite |

string test

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

string parameterString

| Name | Type | Direction | Description |
|-----------------|--------|-----------|-----------------------------|
| parameterString | string | IN | Selects or deselects a test |

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 C-PHY Tx Essentials version |

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

string filePath

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

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

string 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.

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 |

int rowNr

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

Save recall or query a saved session

| Command name | Parameters | Description | Return value | Example |
|---------------------|-----------------------------------|--|---|--|
| CheckSessionSaved() | string clientID out bool saved | This method checks whether the current session is saved. | Return value is either True or False | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.CheckSessionSaved(m_clientID, out savedStatus) |
| RecallSession() | string clientID string name | Recalls a saved session. The client provides the session name. | String that gives the status of the operation after it has been performed The return value is "Session Recalled..." | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.RecallSession(clientID, savedSessionName) |
| SaveSession() | string clientID string name | 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..." | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.SaveSession(clientID, desiredSessionName) |
| SaveSessionAs() | string clientID string name | Saves the current session under a different name every time this method is called. The client provides the session name. | String that gives the status of the operation after it has been performed The return value is "Session Saved..." | m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.SaveSessionAs(clientID, desiredSessionName) |

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

out 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.

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

string name

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

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

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.

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

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

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

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

out string clientID

| Name | Type | Direction | Description |
|----------|--------|-----------|---|
| clientid | string | OUT | Identifier of the client that is connected to the server clientid = unique number + ipaddress of the client. For example, 1065+192.157.98.70 |

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

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...".

Reference

C-PHY Tx Essentials Parameters

General parameters

This table lists the general parameters that can be changed in the User Defined Mode.

Table 13: General parameters list

| Parameter | Return Values | Default | Minimum | Maximum |
|-----------------------|---|--|---------|---------|
| Data Rate (Gbps) | NA | 2.5 | 0.05 | 4.5 |
| VA | CH1,CH2,CH3,CH4,REF1,REF2,REF3,REF4,MATH1,MATH2,MATH3,MATH4 | CH1 | NA | NA |
| VB | CH1,CH2,CH3,CH4,REF1,REF2,REF3,REF4,MATH1,MATH2,MATH3,MATH4 | CH2 | NA | NA |
| VC | CH1,CH2,CH3,CH4,REF1,REF2,REF3,REF4,MATH1,MATH2,MATH3,MATH4 | CH3 | NA | NA |
| Enable DeEmbed Filter | Yes No | No | NA | NA |
| Filter Type | Insertion Loss Insertion Loss and Cross Talk | Insertion Loss and Cross Talk | NA | NA |
| DeEmbed Filter 1 | NA | C:\Program Files\Tektronix\TekExpress\TekExpress CPHYTx\Compliance Suites\CPHY\FilterFiles\sdlaTp3DIFFSrcA_AB.fl | NA | NA |
| DeEmbed Filter 2 | NA | C:\Program Files\Tektronix\TekExpress\TekExpress CPHYTx\Compliance Suites\CPHY\FilterFiles\sdlaTp3DIFFSrcB_AB.fl | NA | NA |
| DeEmbed Filter 3 | NA | C:\Program Files\Tektronix\TekExpress\TekExpress CPHYTx\Compliance Suites\CPHY\FilterFiles\sdlaTp3DIFFSrcB_BC.fl | NA | NA |

| Parameter | Return Values | Default | Minimum | Maximum |
|------------------|---------------|--|---------|---------|
| DeEmbed Filter 4 | NA | C:\Program Files \Tektronix\TekExpress \TekExpress CPHYTx \Compliance Suites \CPHY\FilterFiles \sdlTp3DIFFSrcC_BC.fl t | NA | NA |
| DeEmbed Filter 5 | NA | C:\Program Files \Tektronix\TekExpress \TekExpress CPHYTx \Compliance Suites \CPHY\FilterFiles \sdlTp3DIFFSrcC_CA.fl t | NA | NA |
| DeEmbed Filter 6 | NA | C:\Program Files \Tektronix\TekExpress \TekExpress CPHYTx \Compliance Suites \CPHY\FilterFiles \sdlTp3DIFFSrcA_CA.fl t | NA | NA |

Test parameters

This table lists which test parameters can be changed in the User Defined Mode.

Table 14: Test parameters list (User Defined mode)

| Parameter group | Parameter name | Range | Default | Units |
|-----------------|--------------------------------|--|-----------------------------|--------|
| Ref levels | Reference levels | Absolute, Percentage | Percentage | NA |
| | Reference level-High (%) | 70 to 90 (in %) 40 to 60 (in Absolute) | 80 (in %) 58 (in Absolute) | % or V |
| | Reference level-Low (%) | 10 to 30 (in %) -60 to -40 (in Absolute) | 20 (in %) -58 (in Absolute) | % or V |
| | Reference level-Hysteresis (%) | 5 to 15 (in %) 5 to 25 (in Absolute) | 10 (in %) 10 (in Absolute) | % or V |
| Clock Settings | Clock recovery method | EXPEDGE EXPPLL | EXPEDGE | NA |
| | Signal Type | CLOCK DATA AUTO | CLOCK | NA |
| | Clock Edge | RISE FALL BOTH | RISE | NA |
| | Loop bandwidth (MHz) | 1 to 10 | 4 | MHz |
| | MaskHitType | Auto Manual | Auto | NA |

| Parameter group | Parameter name | Range | Default | Units |
|-----------------|-----------------------|---------------|---------|-------|
| Others | Accumulation | True False | True | NA |
| | Eye Height Percentage | 10 to 90 | 50 | % |
| | Hysteresis | 5 to 15 | 10 | % |

Map 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 C-PHY Tx Essentials 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**.

Rise time

Verifies that the rise time of the DUT C-PHY Tx HS (80% - 20%) signals.

Fall time

Verifies the fall time of the DUT C-PHY Tx HS (80% - 20%) signals.

Eye measurement

Verifies the eye diagram and performs both eye widths and eye height measurement on C-PHY Tx Essentials signals.

DC common mode measurement

Verifies that the static point common mode voltage $V_{CP_{TX}}$ of the Trio signal are within the transmitter limit.

AC common mode mismatch measurement

Verifies that the common-mode voltage mismatch ($\Delta V_{CMT_{PX}}$) of the DUT Data Lane HS Transmitter is less than the maximum conformance limit.

AC common mode level variation $\Delta V_{CPTX(LF)}$ between (50 MHz to 450 MHz)

Verifies that the AC common-point signal level variations between 50 and 450 MHz ($\Delta V_{CPTX(LF)}$) of the transmitter are less than the maximum allowable limit.

AC common mode level variation $\Delta V_{CPTX(HF)}$ (Above 450 MHz)

Verifies that the AC common-mode signal level variations above 450 MHz ($\Delta V_{CPTX(HF)}$) of the DUT HS transmitter are less than the maximum allowable limit.

Intra-pair skew

Measures the intra pair skew of the trio signal.

Index

A

- Acquire status, 22
- Acquisitions tab, 20
- Activating the license (C-PHY Tx), 3
- Analysis Status, 22
- Application
 - exit, 9
 - license, 9
 - version information, 4
- ApplicationStatus(), 67
- Auto Scroll, 23

C

- CheckSessionSaved(), 76
- Clear test log, 23
- Client (TekExpress), 39
- Client proxy object, 43
- Client requirements, TekExpress, 40
- Code example, remote access, 46
- Command buttons, 10
- Commands
 - ApplicationStatus(), 66
 - error codes, 64
 - GetCurrentStateInfo(), 66
 - QueryStatus(), 66
 - SendResponse(), 66
 - SetVerboseMode(), 50
- Configure email, 14
- Configure email setting, 13
- Configuring tests, 18
- Connect(), 48
- Connected instrument(s), 19
- Connected instruments, 31
- Contact Tektronix, 1
- Conventions, 1

D

- Demo mode, 9
- Description of main application panels, 14
- Disable Popups command, 50

- Disconnect(), 79
- Download the software, 3
- DUT
 - definition, 1
 - settings, 16
 - tab, 16

E

- Email notifications, 14
- Email settings, 13
- Error codes, 64
- Evaluation mode, 9
- Exiting the application, 9
- Extensions, file name, 7

F

- File name extensions, 7

G

- Gear selection, 16
- GetAcquireParameter(), 55
- GetAnalyzeParameter(), 55
- GetCurrentStateInfo(), 67
- GetDutId(), 51
- GetGeneralParameter(), 55
- GetPassFailStatus(), 70
- GetReportParameter(), 71
- GetResultsValue(), 70
- GetResultsValueForSubMeasurements(), 70
- GetTimeOut(), 65
- Global settings, 18
- Global Settings tab, 19

I

- Installing the software, 3
- Instruments
 - connected, 19
 - selecting, 19

- viewing connected, 12, 31
- Instruments detected, 19
- Interface error codes, 64

K

- Keep On Top, 9

L

- License, 9
- License activation (C-PHY Tx), 3
- License information, 4
- LockSession(), 49
- Log files, 23
- Log View tab (Status panel), 22

M

- Main application panels, 14
- Measurements tab, 19
- Menus
 - Options, 11
 - Preferences, 24
- Message History, 23
- Minimum system requirements, 6
- My TekExpress folder
 - locating, 33

O

- Options menu
 - Instrument Control Settings, 12
 - Keep On Top, 9

P

- Panels
 - Reports, 25
 - Results, 23
 - Setup, 15
 - Status, 22
- Panels overview, 14
- Pass/Fail summary, 27
- Pre-run checklist, 33, 35
- Program example, 46

- Programmatic interface
 - client, 39
 - client proxy object, 43
 - client requirements, 40
 - example, 46
 - overview, 39
 - remote proxy object, 42
 - server, 39

Q

- QueryStatus(), 67

R

- Recall a test setup, 38
- RecallSession(), 76
- Related documentation, 1
- Remote proxy object, 42
- Reports
 - file type (.mht, .html, .csv, .xls), 26
 - main panel, 25
 - names, 26
 - options, 26
 - selecting contents, 26
 - viewing, 27
- Results panel, 23, 24
- Run(), 63
- Running tests, 33

S

- SaveSession(), 76
- SaveSessionAs(), 76
- Saving test setups, 37
- Select Channel, 20
- Selecting a connected instrument, 19
- SelectTest(), 58
- SendResponse(), 67
- Server (TekExpress), 39
- Server control, 40
- Session files, 24
- Session folders, 24
- Set DUT parameters, 16
- Set email parameters, 13

SetAcquireParameter(), 52
 SetAnalyzeParameter(), 52, 62
 SetDutId(), 51
 SetGeneralParameter(), 52, 59
 SetTimeout(), 65
 Setting up tests, 32
 Setup panel, 15
 Setup tabs
 Acquisitions, 20
 DUT, 16
 SetVerboseMode(), 50
 Show application version, 4
 Show Detailed Log, 23
 Signal Path Compensation (SPC), 33
 Signal source validation
 setting, 20
 Software download and installation, 3
 Status panel, 22
 Stop(), 63
 System requirements, 6

T

Tab
 Acquisitions, 20
 Configuration, 18
 DUT, 16
 Tabs
 Log View, 22
 Test Status, 22
 Technical support, 1
 Tektronix support, 1
 Test configuration, 18

Test groups, 29
 Test Results, 24
 Test Selection button, 19
 Test setup files, 24
 Test setup overview, 29
 Test setup steps, 32
 Test setups
 creating based on existing, 38
 recalling, 38
 saving, 37
 Test setups overview, 37
 Test Status commands, 66
 Test Status tab (Status panel), 22
 Tests
 running, 33
 setting up, 29
 TransferImages(), 71
 TransferReport(), 71

U

UnlockSession(), 78
 Untitled Session folder, 33
 User Comments
 including in test reports, 26
 location in reports, 27
 User Defined Mode, 19

V

View test parameters, 18
 View test reports, 27

