TekExpress
DisplayPort Tx Automated Solution
Printable Application Help

077-0621-02

Tektronix

TekExpress
DisplayPort Tx Automated Solution
Printable Application Help

www.tektronix.com
077-0621-02

Tektronix

Contacting Tektronix

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

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

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

Table of Contents

Saving and recalling test setups

Setting up and configuring tests

Running tests and viewing results

Algorithms

TekExpress programmatic interface

Reference

Welcome

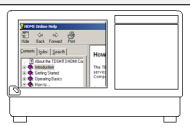
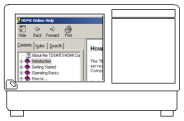
The Tektronix DP12 software, combined with a DisplayPort AUX controller automates DisplayPort physical layer source compliance testing. DP-AUX eliminates the need for user interaction during testing and the need to use vendor-specific proprietary software. Engineers can simply select the desired tests to run and work on other tasks while the tests are being executed. The DP-AUX stand-alone GUI can also be used in validation, debug, and characterization of DisplayPort Source and Sink devices. The new TEK-GRL-DP-SINKSW automates the DP Rx testing.

Getting help and support

Related documentation

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

Table 1: Product documentation

Item	Purpose	Location
Online Help	In-depth operation and UI help	
PDF of the Online Help	In-depth operation and UI help	

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 term “select” is a generic term that applies to the two mechanical methods of choosing an option: using a mouse or using the touch screen.

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
- If possible, save the waveform on which you are performing the measurement as a .wfm file

Getting started

Installing the software

Compatibility

The TekExpress DisplayPort Tx application runs on the following Tektronix oscilloscopes:

- DPO/DSA/MSO71254, DPO/DSA/MSO71604, and DPO/DSA/MSO72004 Series Digital Oscilloscopes
- DPO/DSA71254B, DPO/DSA71604B and DPO/DSA72004B Series Digital Oscilloscopes
- DPO/DSA/MSO71254C, DPO/DSA/MSO71604C and DPO/DSA/MSO72004C Series Digital Oscilloscopes
- DPO/DSA72504D and DPO/DSA73304D Series Digital Oscilloscopes
- DPO/MSO72304/DX72504DX/73304DX Series Digital Oscilloscopes

See Also. *Minimum system requirements*

Minimum system requirements

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

Table 2: System requirements

Component	Requorement
Oscilloscope	■ MSO70000/C/DX, DPO70000/B/C/D/DX, and DSA70000/B/C/D series oscilloscopes with at least 8 GHz for Compliance Mode for RBR and HBR, and noncompliance for HBR2 (12.5 GHz) speeds For list of compatible oscilloscopes, see *Compatibility*.
Processor	Same as the oscilloscope
Operating System	Same as the oscilloscope (Windows XP (32-bit) SP2 and higher, Windows 7 64-bit and 32-bit)
Memory	Same as the oscilloscope
Hard Disk	Same as the oscilloscope
Display	Same as the oscilloscope

Component	Requorement
Software	■ DPOJET, Jitter and Eye Diagram Analysis Tool, 3.4.0.17 or later ■ National Instruments LabVIEW Runtime 5.0.1(32 bit) or 6.0.1(64 bit) or later ■ National Instruments TestStand engine 4.2.1 ■ Microsoft Internet Explorer 6.0 SP1 or later ■ Microsoft Photo Editor 3.0 or equivalent software for viewing image files ■ Adobe Reader 7.0 or equivalent software for viewing portable document format (PDF) files ■ SDLA software for creating CTLE filters
Probes	TCA-SMA (P7380SMA/P7313SMA), differential and single-ended
Test Fixtures	Wilder DisplayPort fixtures (DP-TPA-P, DP-TPA-R, DP-TPA-C, DPI-TPA-PA, DPI-TPA-PRRCA, MDPI-TPA-PA and MDPI-TPA-PRRCA)
RF Switch	(Optional) Keithley S46 Microwave Switch System
Other Devices	■ Microsoft compatible mouse or compatible pointing device ■ Four USB ports (two USB ports minimum) ■ PCI-GPIB or equivalent interface for instrument connectivity [1] ■ Wilder CIC boards (optional)

See Also. *Compatibility*

[1] The virtual GPIB port is used by TekExpress for communicating with oscilloscope applications. Make sure the GPIB configuration is set to “Controller” mode on the TekScope. 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 oscilloscope GPIB menu. For ease of use, connect to an external (secondary) monitor.

Install the software

Install the software on a compatible Tektronix Real-Time oscilloscope running Windows XP or Windows 7. For a list of compatible oscilloscopes, see *Compatibility*.

1. Go to the www.tek.com Web site and search for DisplayPort Tx to locate the installation file. Download the file DisplayPort TXWebInstaller.exe.
2. Close all applications, including any TekVISA applications.
3. Double-click the executable file to extract the installation files.

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

   - Windows XP location: C:\Program Files\Tektronix\TekExpress \TekExpress DisplayPort Tx
   - Windows 7 location: C:\Program Files (x86)\Tektronix\TekExpress \TekExpress DisplayPort Tx

See Also. *Minimum system requirements*

Activate the license

Activate the license using the option installation wizard on the oscilloscope. Follow these steps to activate the TekExpress DisplayPort Tx license:

1. From the oscilloscope menu bar, click **Utilities** > **Option Installation**.

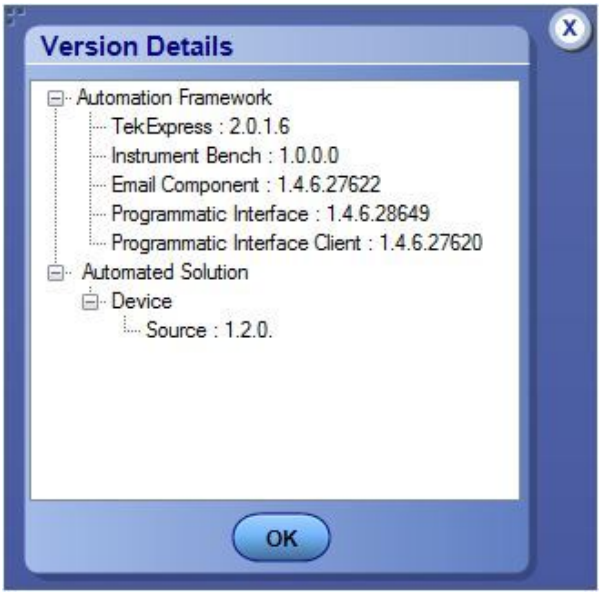
   The TekScope Option Installation wizard opens.
2. Instructions for using the Options Installation window to activate licenses for installed applications is provided in the oscilloscope online help. Press the **F1** key on the oscilloscope keyboard to open the Option Installation help topic. Follow the directions in the topic to activate the license.

See Also. *View version and license information*

View version and license information

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

To view version information for application, click the **Options** button in TekExpress and select **About TekExpress**. Click **View Version Details**to view version details.



NOTE. *This example shows a typical Version Details dialog box, and may not reflect the actual values as shown when you open this item in the application.*

To view license information:

1. From the oscilloscope Help menu, select **About TekScope**.

   The Options section in the dialog box displays a list of installed options, including DisplayPort Tx.

2. To view the Option key, see the Option installation key section. When finished, click **OK** to close the dialog box.

See Also. *Activate the license*

Options menu

Introduction to the application

DisplayPort Tx application overview

NOTE. *The skew between Data+ and Data- in the signal path from DUT to the oscilloscope is computed and compensated before running the measurements.*

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

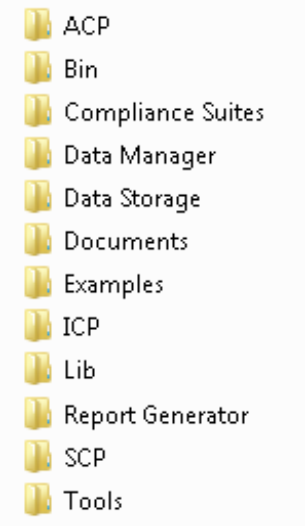
Key Features. DisplayPort Tx has the following key features:

- Fully-automated compliance testing of DP1.2 Source devices
- Support for Dual Mode (DP++) and AUX channel tests
- Integration with DP-AUX and RF switch matrix (optional)
- Automatic insertion of modeled channel losses and CTLE equalization per the CTS
- User-Defined mode for characterization, margin testing and debugging (in addition to compliance testing)
- Support for P7313SMA and P7380SMA differential probes (optional)
- DP-AUX provides DUT control for full automation, saving hours of manual DUT settings.
- Greater bandwidth allows for increased resolutions, higher refresh rates, and greater color depth.
- Automated testing:
  - Minimizes user intervention when conducting time-consuming testing
  - Reduces the time required to conduct testing
- Selective testing:
  - Performs fully-automated testing for transmitter measurements
  - Allows selecting test individually or in groups
  - Avoids repeated testing through accurate and reliable results from a single run
- Quick testing:
  - One-button selection of multiple tests ensures faster testing.
  - Test margins and statistical information promote quick analysis.
  - One-button MHT report format saves time.
- Complete Solution: Complete compliance solution with an elaborate test fixture and signal sources support for SMA cable-based solution provides a cost-effective way to perform compliance testing.

- Customize the setup:
  - Modify the test setup according to the DUT configuration.
  - Run tests using different record lengths.
  - Run test measurements using more than one signal.
- Detailed test reporting:
  - Provides a Pass/Fail summary table
  - Provides margin details on each test
  - Provides a consolidated report for all tests
  - Provides additional information such as skew, alignment, and signal type selected for each measurement

Application directories and usage

The application directory and associated files are organized as follows:



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

Table 3: Application directories and usage

Directory names	Usage
InstallDir\TekExpress\TekExpress DisplayPort Tx	Contains the application and associated files
TekExpress DisplayPort Tx\ACP	Contains instrument and DisplayPort Tx application-specific interface libraries
TekExpress DisplayPort Tx\Bin	Contains the miscellaneous DisplayPort Tx application libraries
TekExpress DisplayPort Tx\Compliance Suites	Contains compliance-specific files

Directory names	Usage
TekExpress DisplayPort Tx\Data Manager	Contains the result management specific libraries of the DisplayPort Tx application
TekExpress DisplayPort Tx\Data Storage	Contains libraries needed for storing data
TekExpress DisplayPort Tx\Documents	Contains the technical documentation for the application
TekExpress DisplayPort Tx\Examples	Contains various support files
TekExpress DisplayPort Tx\ICP	Contains instrument and DisplayPort Tx application-specific interface libraries
TekExpress DisplayPort Tx\Lib	Contains utility files specific to the DisplayPort Tx application
TekExpress DisplayPort Tx\Report Generator	Contains Excel Active X interface Library for Report Generation
TekExpress DisplayPort Tx\SCP	Contains instrument and DisplayPort Tx application-specific interface libraries
TekExpress DisplayPort Tx\Tools	Contains instrument and DisplayPort Tx application-specific files

See Also. *View test-related files*

File name extensions

The TekExpress DisplayPort Tx application uses the following file name extensions:

File name extension	Description
.TekX	Session files are saved in this format but the extensions may not be displayed.
.seq	The test sequence file
.xml	The encrypted XML file that contains the test-specific configuration information The log file extension is also xml
.wfm	The test waveform file
.mht	Test result reports are saved in this format by default. Test reports can also be *saved in HTML format*.
.pdf	Test result reports Application help document

See Also. *View test-related files*

Application directories and usage

Before you click start

Operating basics

Start the application

The first time you open the application after installation, you are required to enter a license key. The application also 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 DisplayPort Tx. The Resources.xml file contains information regarding instruments available on your network.

NOTE. *When the application starts, it checks for the appropriate license key. If the valid license key is not present, the application switches to the Evaluation mode. You are allowed 10 free trials in Evaluation mode. Each time you open the application without supplying a valid license key, one of the free trials is used. The application continues to run in Evaluation mode until you supply a key or until you have used up the 10 free trials.*

To run the DisplayPort Tx application, do any of the following:

- From the TekScope Analyze menu, select **Analyze** > **TekExpress DisplayPort Tx**.
- Double-click any saved session file.

While running the application, you can switch between the oscilloscope screen and DisplayPort Tx by clicking the desired window. To keep the application window on top, select **Keep On Top** from the *Options menu*.

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

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 will be asked to save it before exiting. To save it, click **Yes**. Otherwise click **No**. The application closes.
   - A message box appears asking if you really want to exit TekExpress. To exit, click **Yes**.

Application controls and menus

Application controls

Table 4: Application controls descriptions

Item	Description
Options menu	Opens the Options menu for access to global controls
Panel	Visual frame with a set of related options. Each panel has a shortcut tab.
Command buttons	Button that initiates an immediate action such as the Start, Stop, Pause, and Clear command buttons
Start button	 Use the Start button to continuously acquire and accumulate measurements. If prior acquired measurements have not been cleared, the new measurements are added to the existing set.
Stop button	 Use the Stop button to abort the test.
Pause \ Continue button	 Use the Pause button to temporarily interrupt the current acquisition. When a test is paused, the button name changes to Continue.

Item	Description
Clear button	 Use the Clear button to clear all existing measurement results. Adding or deleting a measurement, or changing a configuration parameter of an existing measurement, also clears measurements. This prevents the accumulation of measurement statistics or sets of statistics that are not coherent. This button is available only on the *Results panel*.
Application window move icon	 Place the cursor over the three-dot pattern in the upper left corner of the application window. When the cursor changes to a hand, drag the window to the desired location.

Options menu

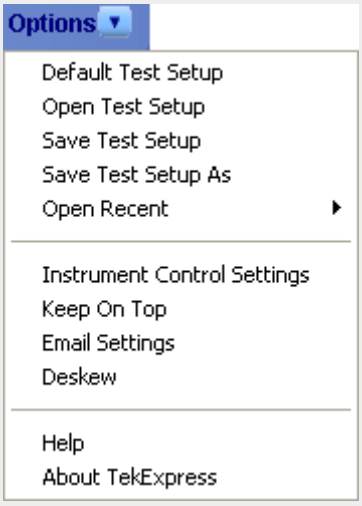
The Options menu is located in the upper right corner of the application.

The Options menu has the following selections:

Menu	Function
Default Test Setup	Opens an untitled test setup with defaults selected
Open Test Setup	Opens a saved test setup
Save Test Setup	Saves the current test setup selections
Save Test Setup As	Creates a new test setup based on an existing one
Open Recent	Displays a menu of recently opened test setups to select from
Instrument control settings	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 DisplayPort Tx utility on top of other open windows on the desktop
Email settings	Use to configure email options for test run and results notifications
Deskew	Use to *deskew channels* after entering skew values in the TekScope Vertical menu

Menu	Function
Help	Displays TekExpress Help
About TekExpress	■ Displays application details such as software name, version number, and copyright ■ Provides access to *license information* for your DisplayPort Tx installation ■ Provides a link to the Tektronix Web site

Options menu

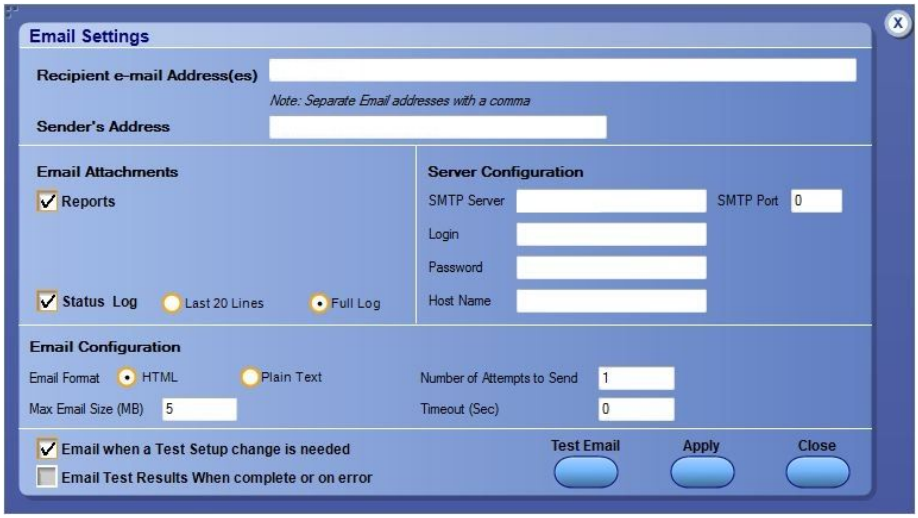


See Also. *Application controls*

Email settings

Use the Email Settings utility to *configure email notifications* if you want DisplayPort Tx to notify you when a tests completes, produces an error, or fails. Select the type of test run information to be included in the notification, such as test reports and test logs, the email message format, and the email message size limit.

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



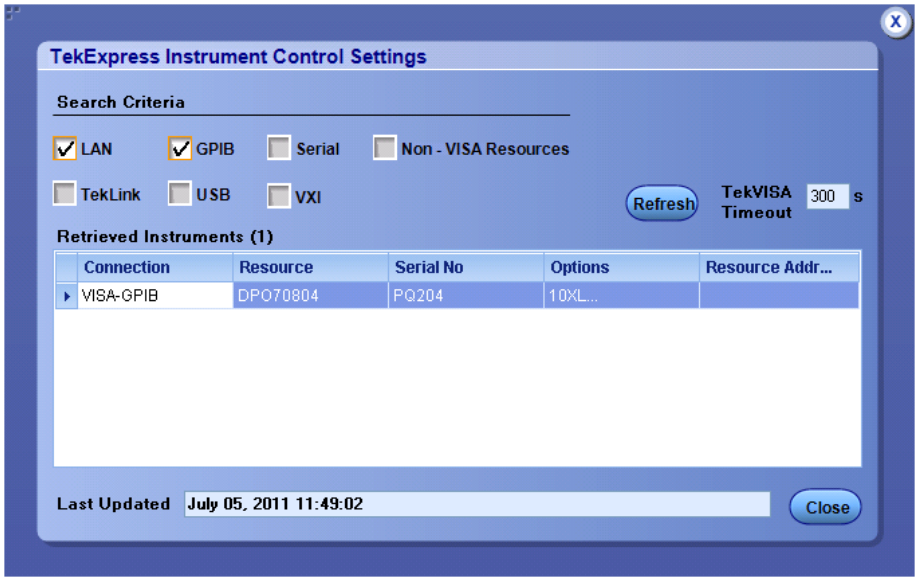
See Also. *Options menu*

Select analysis and test notification preferences

Instrument control settings

The Instrument Control Settings dialog box shows the list of resources found on different connections. It serves two purposes:

- Discovers the connected instruments
- Confirms the instrument connection setup



Use the Instrument Control Settings feature to *search for connected instruments* and view instrument connection details. Connected instruments displayed here can be selected for use in the Global Settings section of the Setup panel's Configuration section. See step 2 of *Configure tests* for details.

See Also. *Options menu*

Panels

About panels

TekExpress DisplayPort Tx has the following main panels:

Table 5: Application panels

Panel name	Purpose
Setup	The Setup panel allows you to configure the test setup. Use this panel to: ■ *Select device parameters*. ■ *Select the test(s)*. ■ *Select acquisitions parameters* for selected tests. ■ *Configure the selected tests*. ■ *Select analysis and test notification preferences*.
Status	View the acquisition status and analysis status of the selected tests, and view test logs.
Results	View a summary of test results and select result viewing preferences.
Reports	Browse for reports, save reports, specify report naming conventions, select content to include (such as summary information, detailed information, user comments, plot images, setup configuration), and select other report viewing options.

See Also. *Application controls*

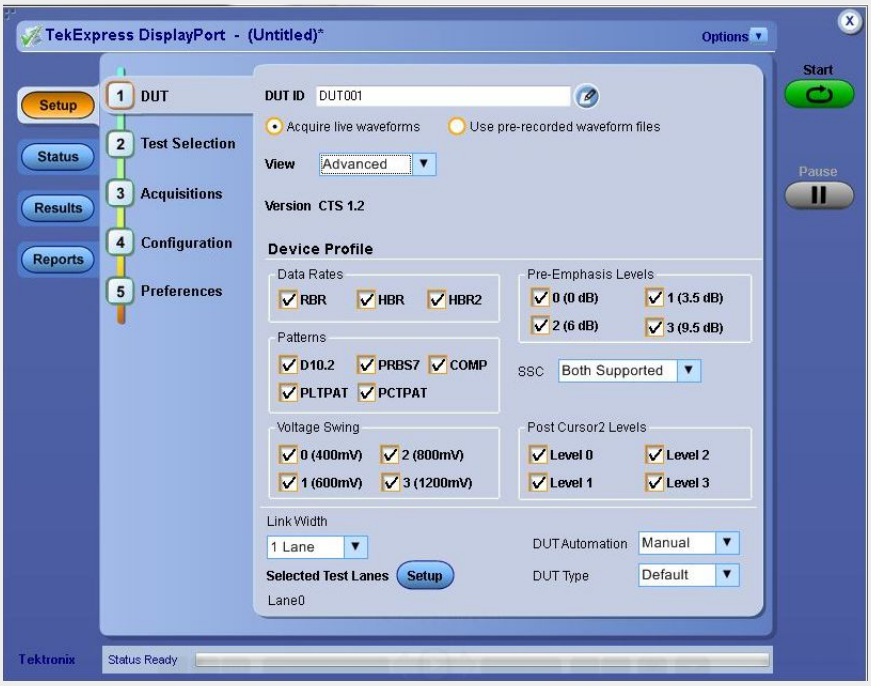
Setup panel

The Setup panel has numbered tabs to guide you through the test setup process:

1. DUT

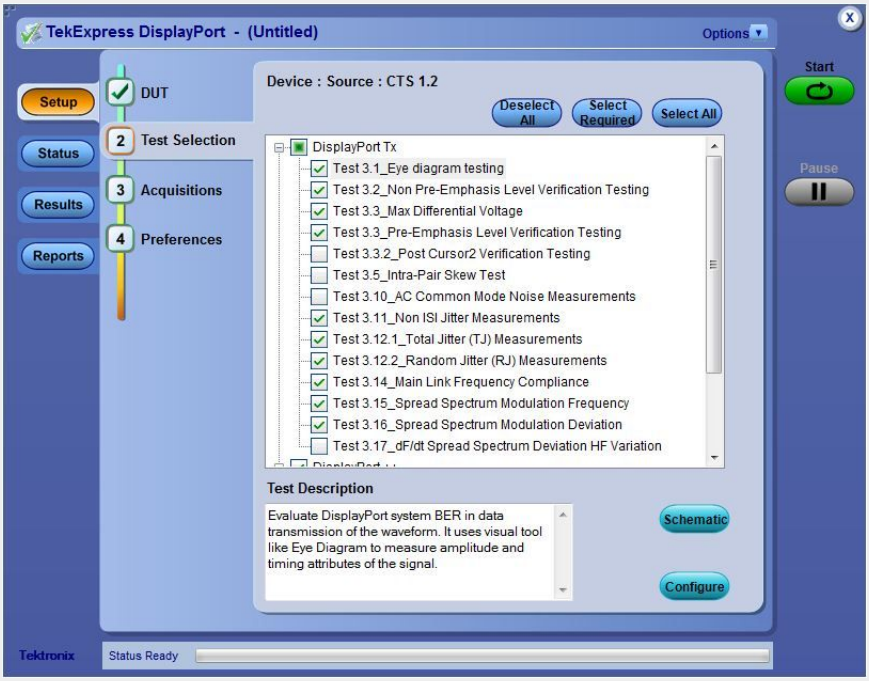
DUT tab, Compliance View

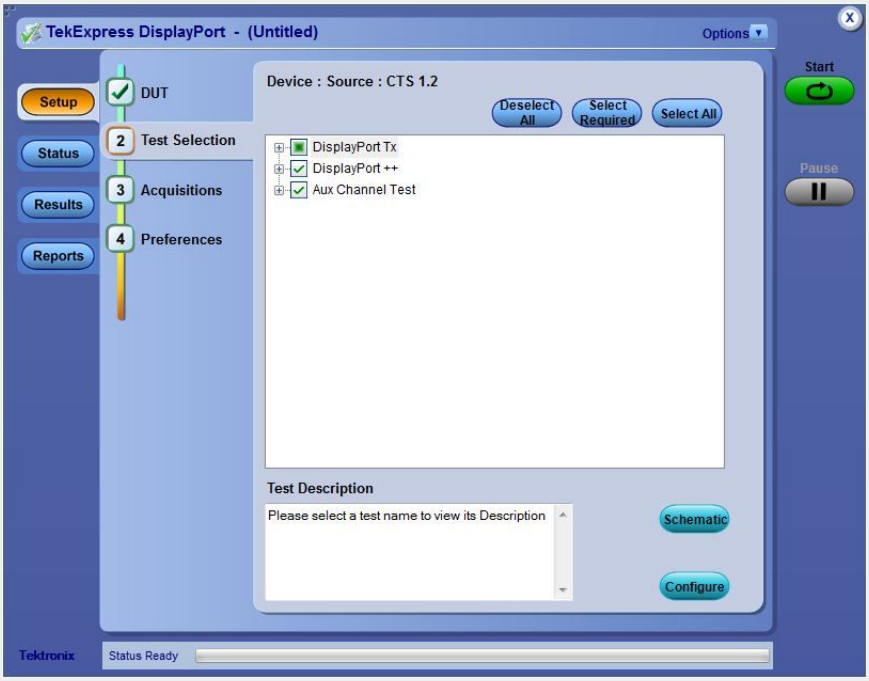
Advanced View



2. Test Selection

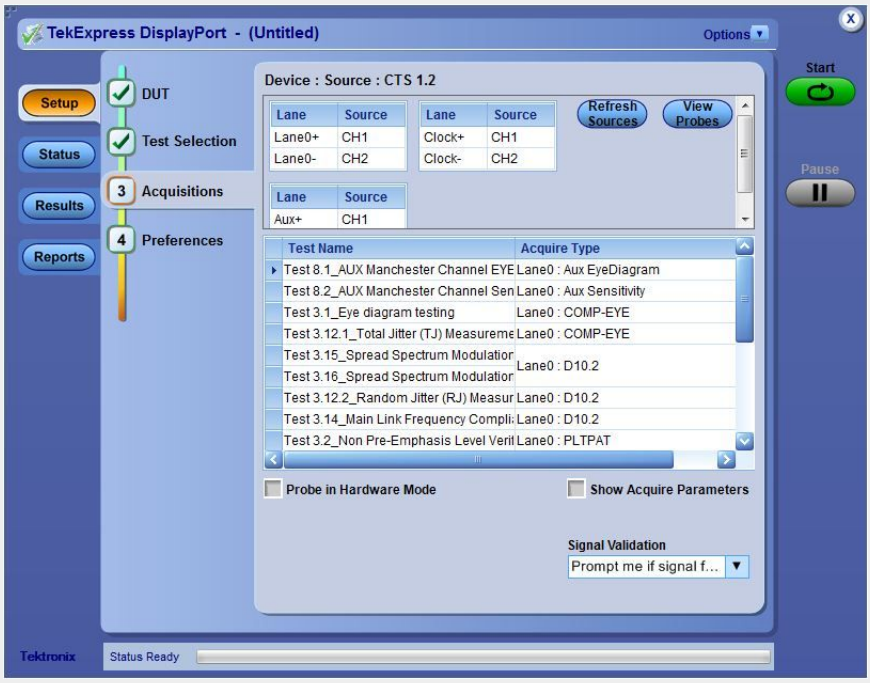
Test Selection tab





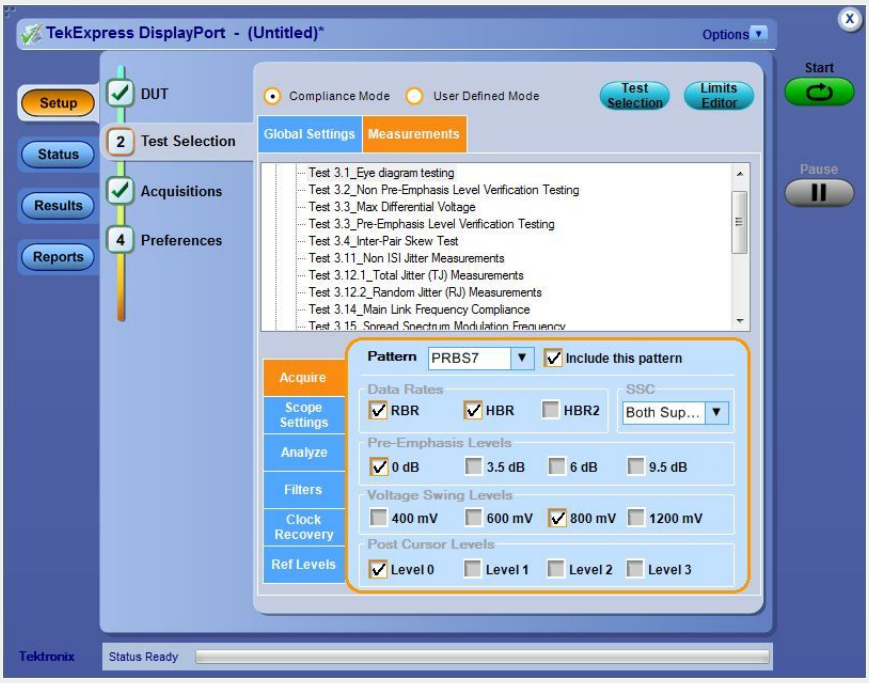
3. Acquisitions

Acquisitions tab



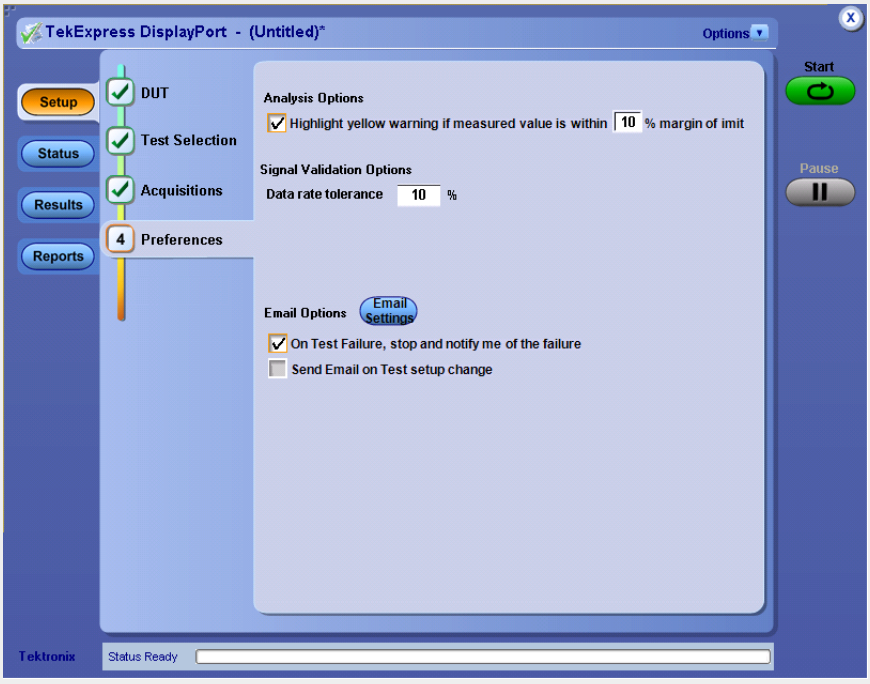
4. Configuration

Configuration tab, Measurements



5. Preferences

Preferences tab



By default, only the DUT, Test Selection, Acquisitions, and Preferences tabs are displayed. The Configuration tab is only available if you select the Advance View option in the DUT tab. Otherwise, configuration settings for tests selected in the Test Selection tab are available from the Configure button on the Test Selection tab.

Options selected in a preceding tab affect options available in the next tab down. For example, Device Profile settings affect the list of tests available for selection in the Test Selection tab. However, you can switch between the tabs in any order if you need to modify your test parameters.

When you finish making selections in a tab and move to the next tab, the number in the tab title changes to a check mark. If you make changes on a tab that significantly affect the settings set in a preceding tab, the check mark in the tab title of the affected tab is replaced with the number again to indicate that you should check the settings on that tab.

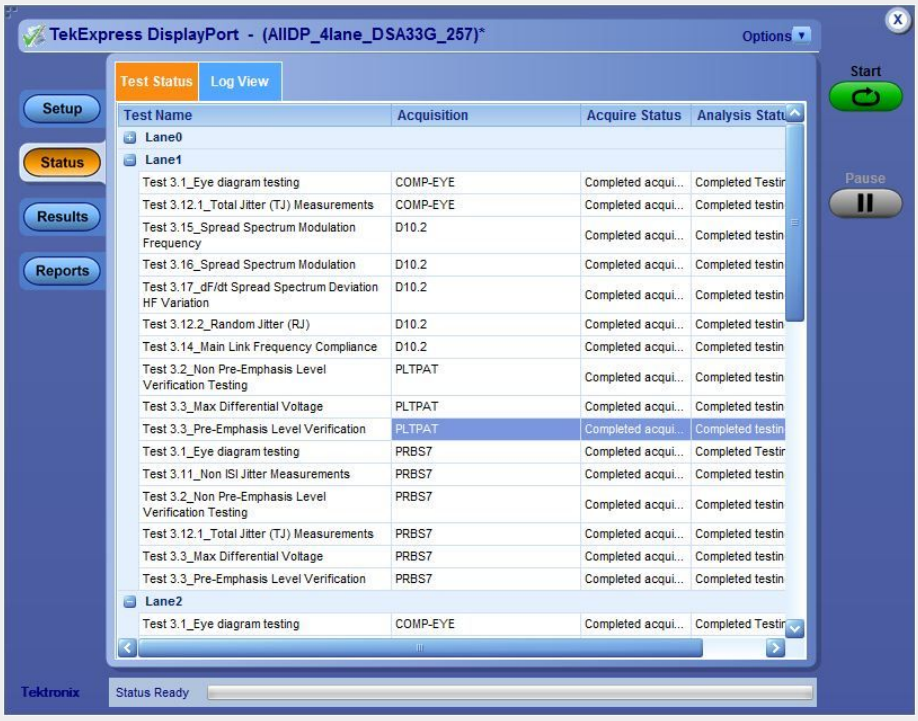
For more information on using the Setup panel, see *About setting up Tests*.

See Also. *About panels*

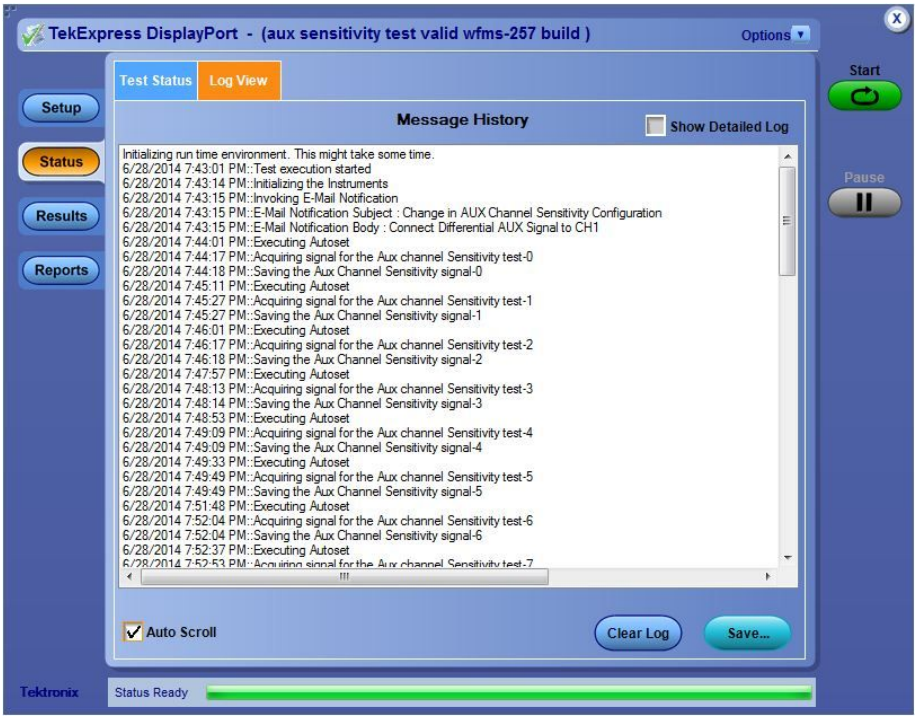
Status panel

Use the Status panel to track the progress of the test measurements. The Status panel has two tabs: the Test Status tab displays the acquire status and the analysis status of the tests as they are being executed; the Log View tab (shown below) displays a view of the test log.

Test Status view



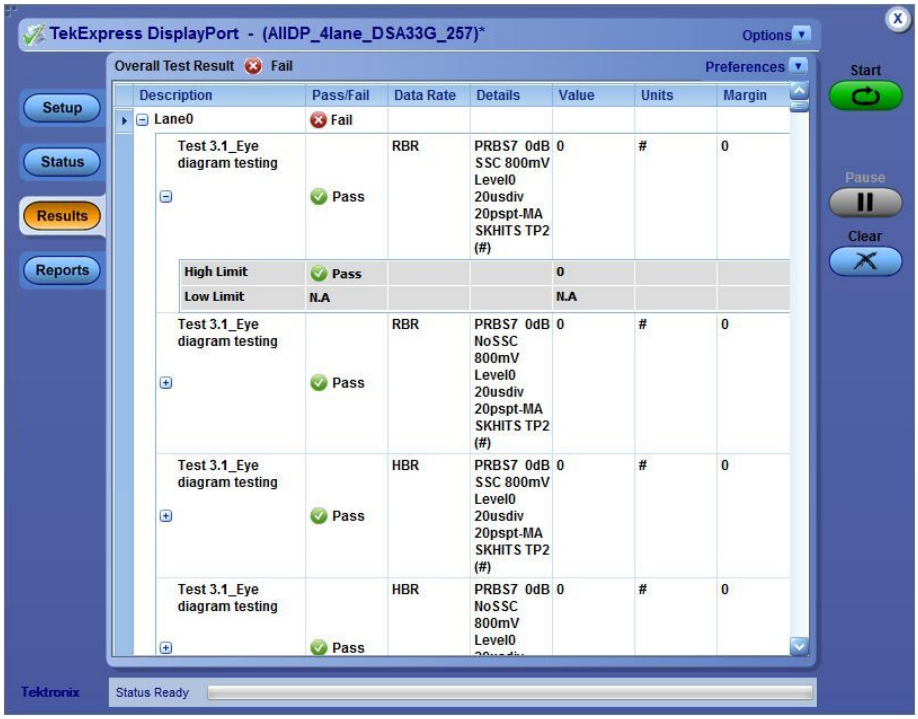
For more information on using the Status panel, see *View the progress of analysis*.



See Also. *About panels*

Results panel

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

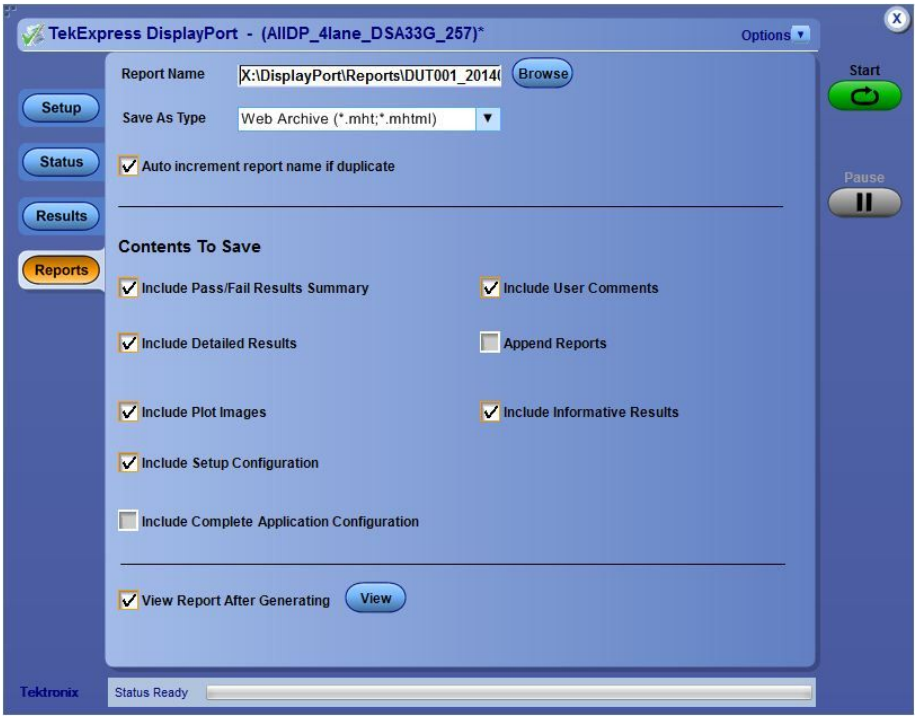


See Also. *About panels*

Reports panel

Use the Reports panel to browse for 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*.



See Also. *About panels*

Saving and recalling test setups

About test setups

TekExpress DisplayPort Tx opens with the default setup selected. Run a test before or after saving a setup. When you save a setup, the test information, such as the selected oscilloscope, general parameters, acquisition parameters, measurement limits, waveforms, and other configuration settings, are all saved under the setup name. Use test setups to:

- *Run a saved test in prerecorded mode*.
- View all the information associated with a saved test, including the Log file, the history of the test status as it executed, and the results summary. For details, see *Recall a saved test setup*.
- *Create a new test setup based on an existing one*.

See Also

About setting up tests

Save a test setup

Delete a test setup

Save a test setup

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

The following instructions start from the default setup:

1. From the Options menu, select **Default Test Setup**.
2. Select the desired options in the Setup panel.
3. Select the desired *report options*.
4. If desired, run the test to be sure that it captures the information you want. If it does not, edit the parameters.
5. From the Options menu, select **Save Test Setup**.

See Also

About setting up tests

Test setup overview

About configuring tests

Delete a test setup

Recall a saved test setup

These instructions are for recalling saved test setups:

1. From the Options menu, select **Open Test Setup**.
2. In the File Open dialog box, select the desired setup from the list and then click **Open**.

See Also

About test setups

Create a new test setup based on an existing one

Delete a test setup

Create a new test setup based on an existing one

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

1. From the Options menu, select **Open Test Setup**.
2. In the File Open dialog box, select the desired setup from the list and then click **Open**.
3. Modify the parameters as desired.
4. From the Options menu, select **Save Test Setup As**.
5. In the File Save As dialog box, enter a test setup name and then click **Save**.

See Also

About test setups

Delete a test setup

Delete a test setup

Delete test setups that you no longer need. This also removes them from the test setup list in the Options menu.

1. Make sure the setup you want to delete is not currently selected in TekExpress DisplayPort. To save any of the test information in the test setup folder, save it to a location other than the DisplayPort folder in the My TekExpress folder.
2. Navigate to the DisplayPort on the X: drive and locate the test setup files you want to delete.
3. Delete both the test setup folder and the test setup file (for example,  314-317 and 314-317, respectively).

See Also

View test-related files

About test setups

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.

Test options include two views, Compliance and Advanced (selected in the DUT tab of the Setup panel), and two modes, Compliance and User Defined (selected in the configuration settings of the Setup panel). The selected view determines where the test configuration settings are displayed. The selected mode determines whether you can change test configuration settings to include those that are noncompliant. The level of user intervention required depends on the DUT Automation setting and Acquisition parameters.

- **Compliance View** selected with **Compliance Mode**: View configuration options in the Test Selection tab of the Setup panel. Tests will run automatically with little or no user intervention. You will not be able to change test parameters to anything that deviates from the compliance standards. The only test configuration parameters that you can change in this mode are the Real Time Scope and the Record Length, both under Global Settings.
- **Compliance View** selected with **User Defined Mode**: View configuration options in the Test Selection tab of the Setup panel. Tests will run automatically but you will be able to change some test parameters before starting the test.
- **Advanced View** selected with **Compliance Mode**: View configuration options in the Configuration tab of the Setup panel. Tests will run automatically with little or no user intervention. You will not be able to change test parameters to anything that deviates from the compliance standards. The only configuration parameters you can change in this mode are the selected Real Time Scope and the Record Length, both under Global Settings.
- **Advanced View** selected with **User Defined Mode**: View configuration options in the Configuration tab of the Setup panel. Tests will run automatically but you will be able to change some test parameters before starting the test.

DisplayPort Supported tests

- 3.1 Eye Diagram:

  This test evaluates the DisplayPort system Bit Error Rate in the data transmission of the waveform. It uses visual tools such as the Eye Diagram to measure the signal's amplitude and timing attributes.

- 3.2 Non Pre-Emphasis Level Verification:

  This test evaluates the reliability of the source on level settings. Ensures that a sink can rely on the source to incrementally increase upon request by the sink.

- 3.3 Pre-Emphasis Level Verification and Maximum Peak-to-Peak Differential Voltage Testing:

  This test evaluates the accuracy of pre-emphasis of the source waveform (on both Transition and Non-transition bits) by measuring the peak differential amplitude.

- 3.3.2 Post Cursor2 Verification:

  This test evaluates the effect of adding Post-Cursor2 in a source waveform by measuring the peak differential amplitude.

- 3.4 Inter-Pair Skew Test:

  This test evaluates the skew, or time delay, between differential data lanes in the DisplayPort interface.

- 3.5 Intra-Pair Skew Test:

  This test evaluate the skew, or time delay, between respective sides of a differential data lane in a DisplayPort interface.

- 3.10 AC Common Mode Noise Measurement:

  This test evaluates the effect of adding Post-Cursor2 in a Source waveform by measuring the peak differential amplitude.

- 3.11 Non ISI Jitter Measurements:

  This test evaluates the amount of Non-ISI jitter accompanying the data transmission. Non-ISI jitter cannot be compensated at the receiver and therefore should be limited in magnitude.

- 3.12.1 Total Jitter (TJ) Measurements:

  This test evaluates the total jitter accompanying the data transmission at a given Bit Error Rate. This measurement is a data time interval error (Data-TIE) jitter measurement.

- 3.12.2 Random Jitter (RJ) Measurements:

  This test evaluates the random jitter accompanying the data transmission at a given Bit Error Rate. This measurement is a data time interval error (Data-TIE) jitter measurement.

- 3.14 Main Link Frequency Compliance:

  This test ensures that the average data rate does not deviate beyond the VESA DisplayPort 1.2 Standard.

- 3.15 Spread Spectrum Modulation Frequency:

This test evaluates the frequency of the SSC modulation and validates that it is within specification limits.

- 3.16 Spread Spectrum Modulation Deviation:

  This test evaluates the range of SSC down-spreading of the transmitter signal in PPM.

- 3.17 dF/dt Spread Spectrum Deviation HF Variation:

  This test ensures that the SSC profile does not include any frequency deviations that would exceed 1250 ppm/uSec.

NOTE. *For tests that involve both single-ended and differential measurements, when both the single-ended and differential measurements are selected, the application does the single-ended measurements first and the differential measurements next.*

Dual-Mode tests

- 3.18 Dual Mode TMDS Clock

  Confirm that the TMDS clock waveform of a source operating in dual-mode does not exceed acceptable limits.

- 3.19 Dual Mode Eye Diagram Testing

  To evaluate the waveform ensuring that timing variables and amplitude trajectories meet the requirements for a dual-mode source device.

Aux Channel tests

- 8.1 AUX Channel Eye Diagram Testing

  Checks if AUX channel waveform timing variables and amplitude trajectories support DisplayPort system.

- 8.2 AUX Channel Eye Sensitivity Test

  Evaluates the sensitivity to the AUX CH EYE opening of a DUT.

See Also

Test setup overview

Before you click start

Deskew channels

About running tests

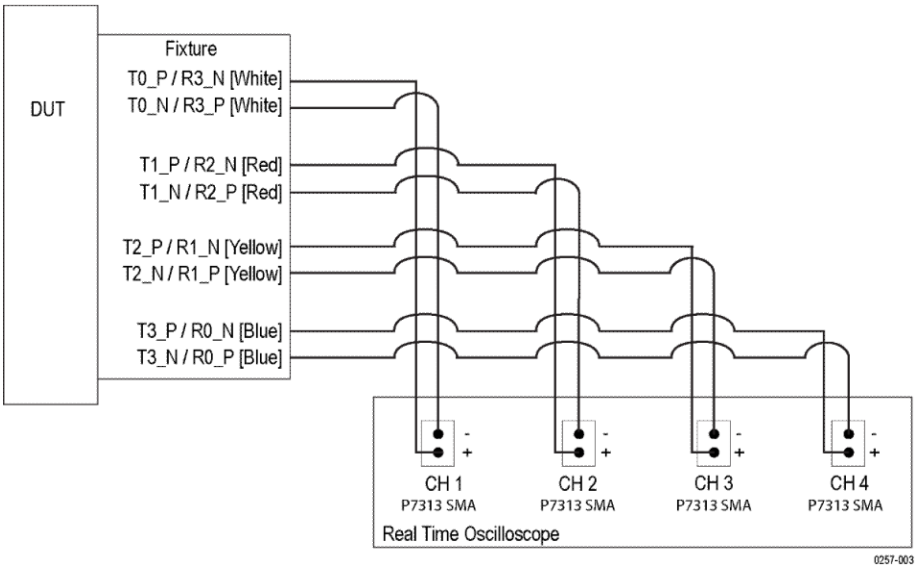
Equipment connection setup

To run tests, you need the following equipment (for details, see *Minimum System Requirements*):

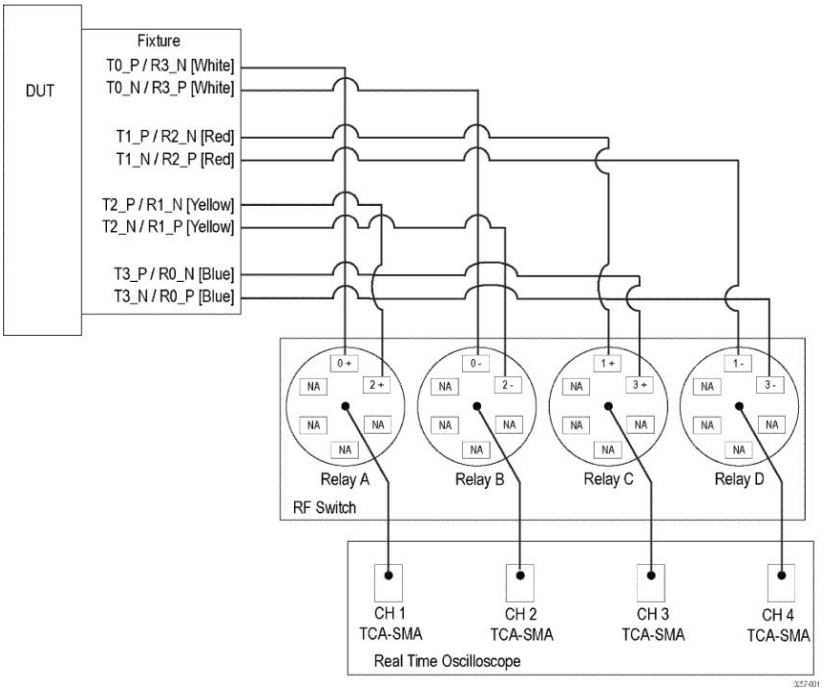
- A *supported Tektronix oscilloscope*
- TCA-SMA or Differential Probes (for example, P7313SMA)
- The device under test
- Test fixtures

The following diagrams show different setups. To view a diagram, click the one for the desired setup.

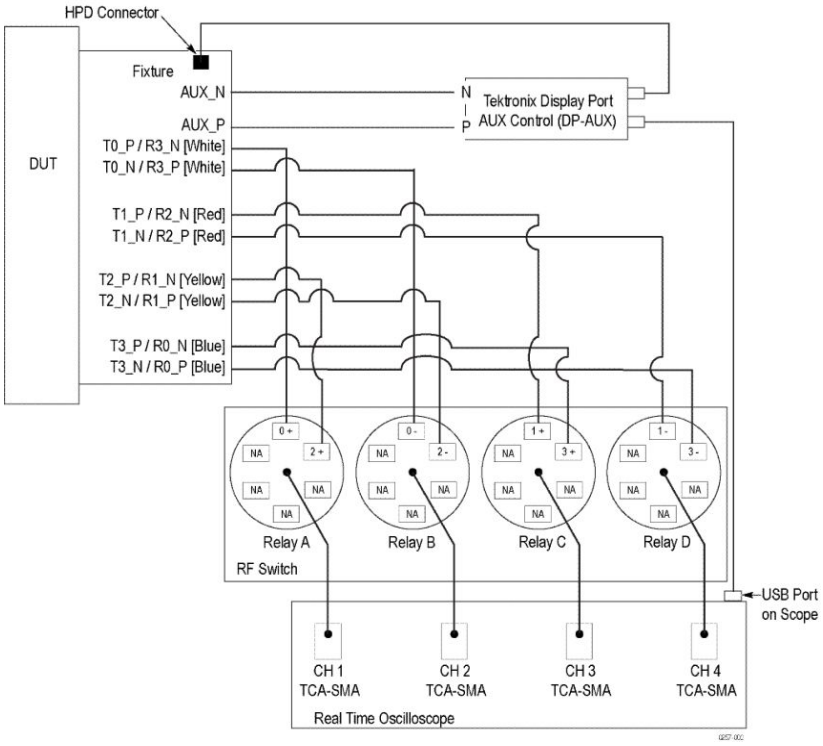
Setup using Differential probes without an RF switch or Tektronix DP-AUX



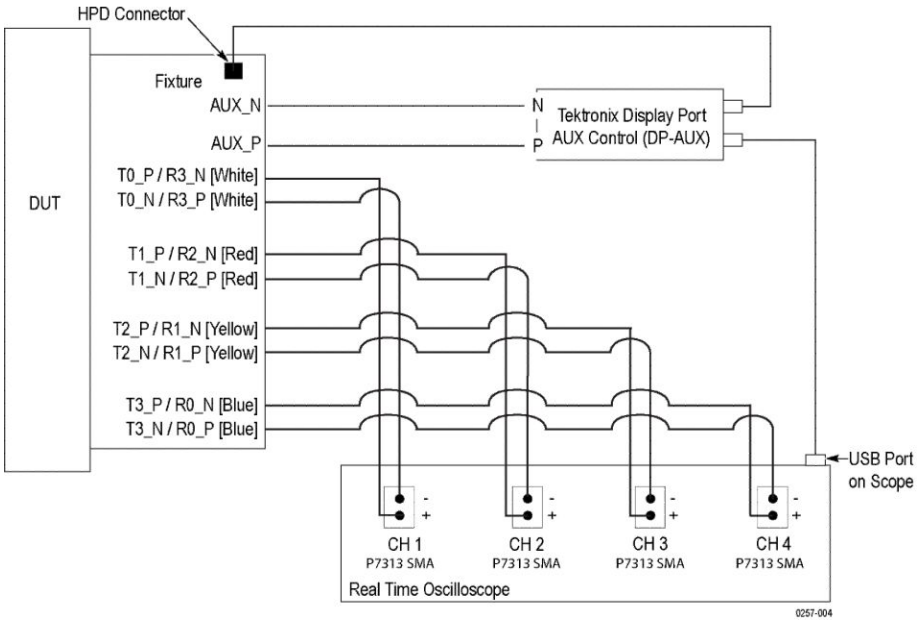
Setup using an RF Switch without Tektronix DP-AUX



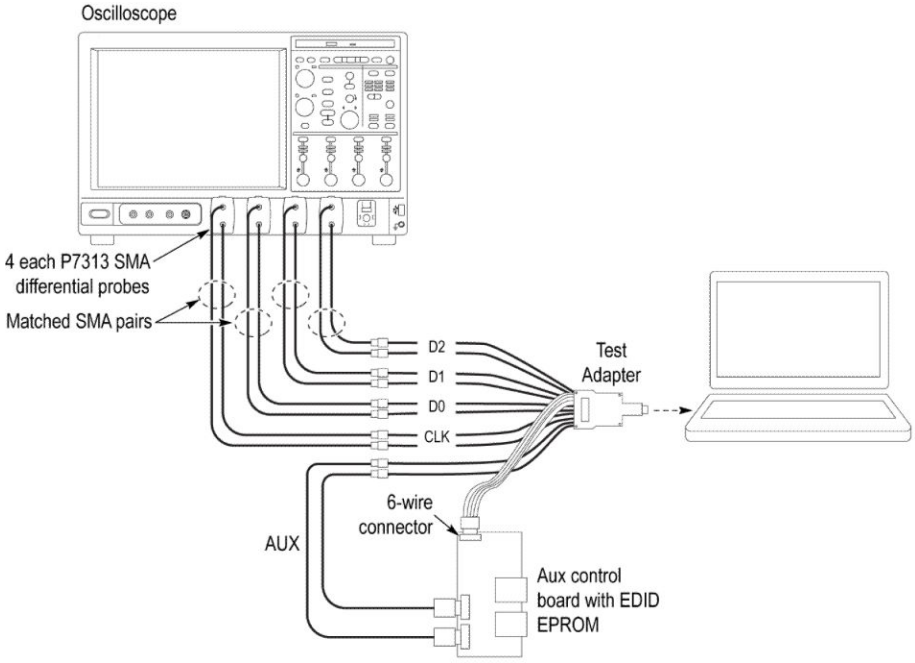
Setup using RF Switch and Tektronix DP-AUX Controller



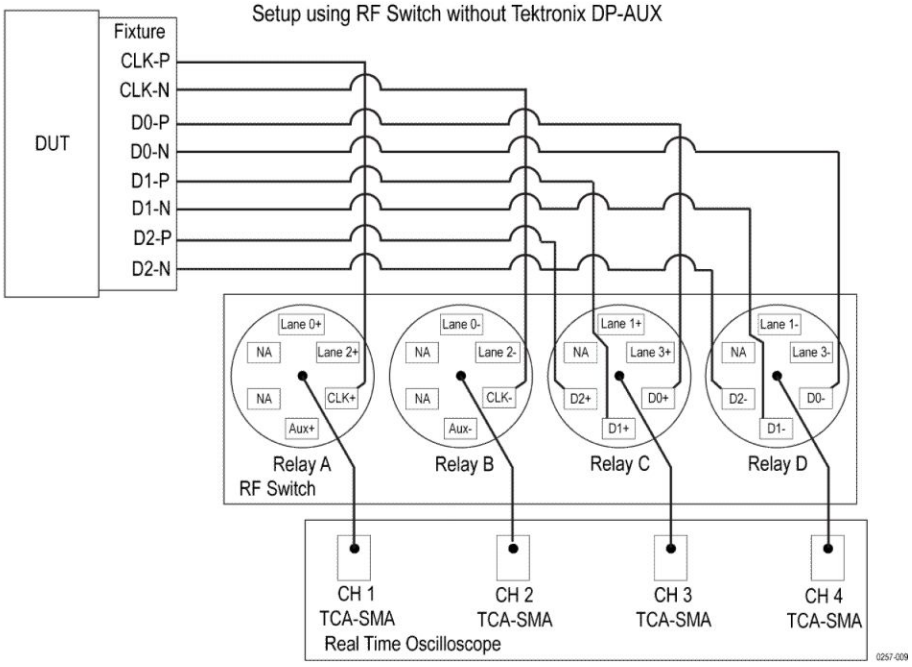
Setup using Tektronix DP-AUX Controller without an RF Switch



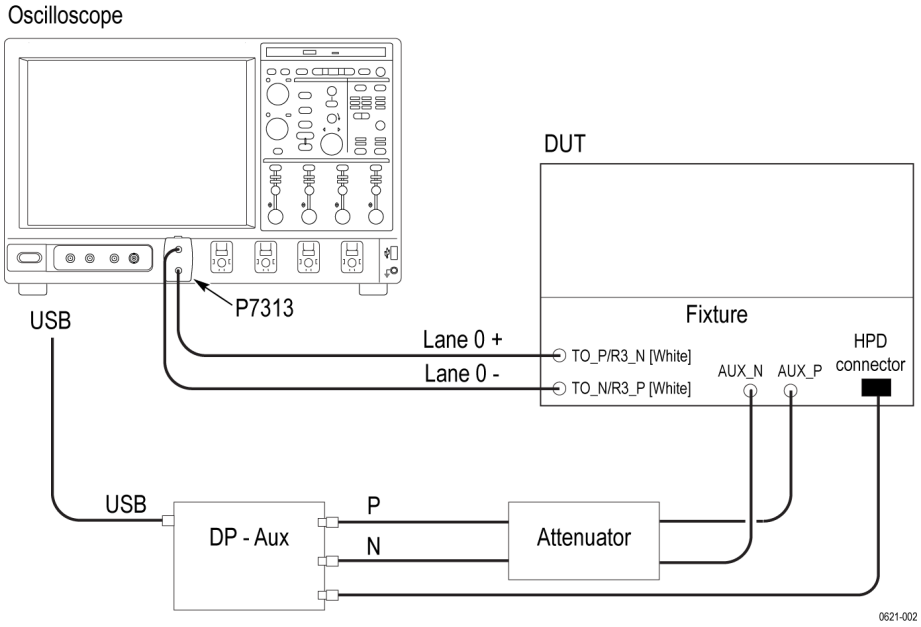
Setup for dual mode



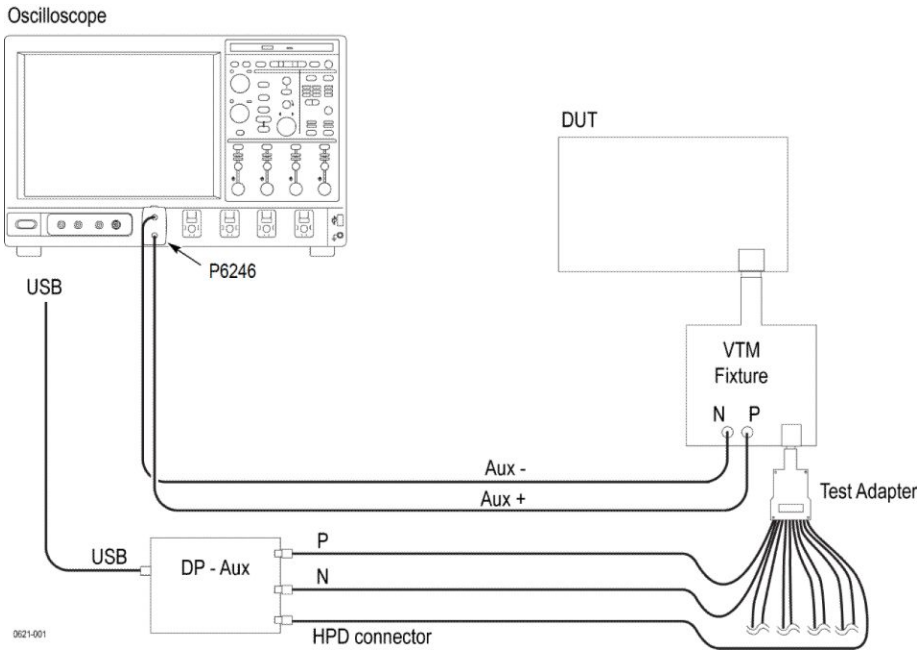
Setup using RF Switch without Tektronix DP-AUX



Setup for Aux Sensitivity test



Setup for Aux Channel test



See Also

View connected instruments

About setting up tests

Pre-Run checklist

Deskew channels

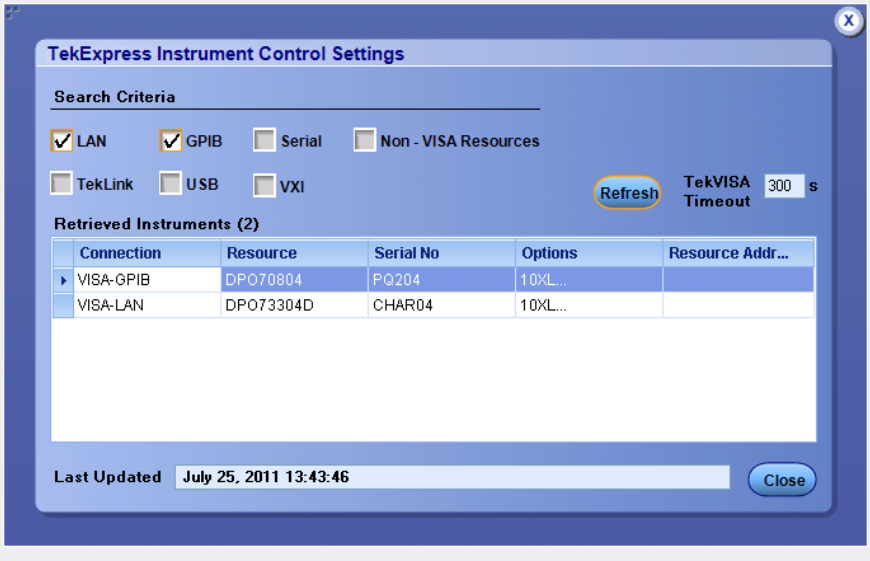
View connected instruments

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

See Also

Configure tests

Equipment connection setup

Test setup overview

Test setup includes acquisition and configuration parameters, but you can also select report options when you set up tests. Below are listed the main test setup steps. Most test setup options are located in the *Setup panel*, but options specific to reports are located in the *Reports panel*.

NOTE. *The Start button is disabled until you are done selecting any required settings necessary to run the test.*

---

1. *Select settings for the device under test*.
2. *Select one or more tests*.
3. *Select acquisitions*.
4. *Configure tests*.
5. *Select analysis and test notification preferences*.
6. *Select report options*.

See Also

Before you click start

Pre-Run checklist

About running tests

Select device parameters

Use the Setup panel DUT tab settings to select settings for the device under test. These options affect the list of available tests in the Test Selection tab.

NOTE. *If TekExpress DisplayPort determines that the connected DUT does not have certain capabilities, then any software options applying exclusively to such capabilities will not be shown as options. For example, if the DUT does not support SSC, then SSC tests will not be shown.*

---

1. In the Setup panel, click the **DUT** tab.
2. Select the global DUT settings that will apply to all tests for the current session:
   a. If needed, in the DUT ID field, enter the ID for the device. The default value is DUT001. The name in this field appears on test reports.
   b. (Optional) To add comments that will appear at the top of the test report (if the Include User Comments option is selected in the *Report options*), click the note pad icon () to the right of the DUT ID field and specify a comment of up to 256 characters.
   c. Select whether to use a prerecorded waveform file or acquire a live waveform for testing.
   d. For View, select **Compliance** or **Advanced**.

      NOTE. *If you select Advanced View, the configuration settings are included in the Setup steps as step 4. If you select Compliance View, the configuration step is not included as a separate step. Access configuration parameters for selected tests from the Configure button in the Test Selection tab.*

      ---

   e. By default, displays the latest CTS version.
3. For the Device Profile section, select from the measurement options described in the following table. The list of tests, and test parameters available in the test configuration section, are filtered based on the characteristics of the DUT and the parameters you select here. For example, if you deselect an option here, such as HBR2, then that option will not be available in the test configuration settings.

Table 6: Device Profile options:

Setting	Description / Option
Data Rates	Select the data rates to include in the tests. ■ RBR: Reduced Bit Rate (1.62G) ■ HBR: High Bit Rate (2.7G) ■ HBR2: High Bit Rate 2 (5.4G)
Patterns	All patterns are selected by default. To exclude a pattern from a test, clear the check boxes for that pattern.
Voltage Swing	All voltage levels are selected by default. At least one voltage swing level must be selected at all times.
Pre-Emphasis Levels	All pre-emphasis levels are selected by default. The 0 (0 dB) level is necessary for all tests and cannot be disabled.
SSC	Only one of the following options can be selected at a time for Spread Spectrum Clocking: ■ Always On: When selected, only the acquisitions that support the SSC On condition will be tested. ■ Always Off: When selected, only the acquisitions that support the SSC Off condition will be tested. ■ Both Supported: When selected, the tests will run under both conditions for those acquisitions that support both. If this is selected for acquisitions that do not support both, then only the supported condition will be tested.
Post Cursor2 Levels	Selected by default based on DUT capability
Link Width	Select the desired number of test lanes to use for this test session. The length width shown here determines the number of test lanes you can select. ■ 1 Lane: When selected, only Lane 0 can be used ■ 2 Lanes: When selected, only lanes 0 and 1 can be used ■ 4 Lanes: When selected, all lanes will be used

Setting	Description / Option
Selected Test Lanes	Displays the test lanes selected for the test session To change lanes selected for testing, click **Setup**. **a.** In the Test Lane Setup dialog box, select the desired number of lanes from the Link Width drop-down list. **b.** To select the lanes to use, click the corresponding lane buttons: ■ To select all four lanes at once, click **Select All**. If you select this, select **4 Lanes** from the Link Width drop-down list. ■ To deselect all selected lanes, click **Deselect All**. ■ If you select 1 Lane, select the **Lane0** button. ■ If you select 2 Lanes, select **Lane0** and **Lane1**. ■ If you select 4 Lanes, select all four Lane buttons. **a.** Click **OK**. Your selections display in the Link Width section of the DUT tab.
DUT Type	The type of vendor DUT. This option only displays if DisplayPort detects a connected DP-AUX controller.

4. The DUT Automation field displays the type of DUT control currently in use. DUT controls are detected by DisplayPort through the Instrument Control Settings feature.
   - **Manual** (default): You will be prompted to configure the DUT manually during the test. You can change this to Custom if needed.
   - **DP-AUX**: You can change the DP-AUX setting to Manual if you want to control the DUT manually during testing.
   - **Custom**: Select this option if you are using a custom utility to manage the DUT.

See Also

Select a test

About setting up tests

Use prerecorded waveforms for analysis

Acquire live waveforms for analysis

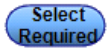
Select a test

Use these instructions to select the tests to run on the connected DUT.

1. In the Setup panel, click the **Test Selection** tab.
2. All tests are selected by default. Click **Deselect All** if you do not want to run all tests.
3. Select the desired test(s):
   - To select all required tests, click the **Select Required** button.
   - To select all tests in the list, click the **Select All** button.
   - To select one or more tests, select the check boxes for the tests.

Test selection controls

Table 7: Setup panel test controls

Button	Description
	When the View type selected in the DUT tab is Compliance, this button opens a configuration section for the selected test. If the View type is Advanced, this button is not displayed.
	Displays the schematic document for the selected test. Use to verify the test setup before running the test
	Deselects all tests in the table
	Selects tests required for compliance and deselects all other tests currently selected
	Selects all tests in the table except 3.3.2 Post Cursor 2 Verification Test

NOTE. *For HBR2 speed: Total Jitter, Random Jitter and Zero Cable tests are dependent on selection of Eye diagram tests.*

See Also

Select acquisitions

About setting up tests

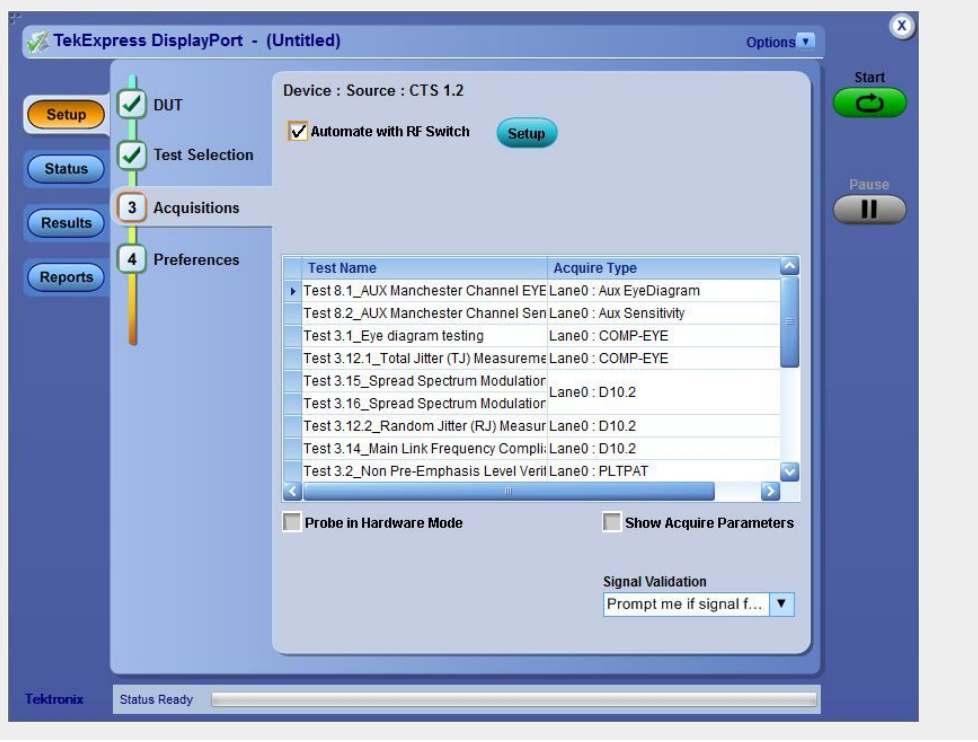
Selecting acquisitions

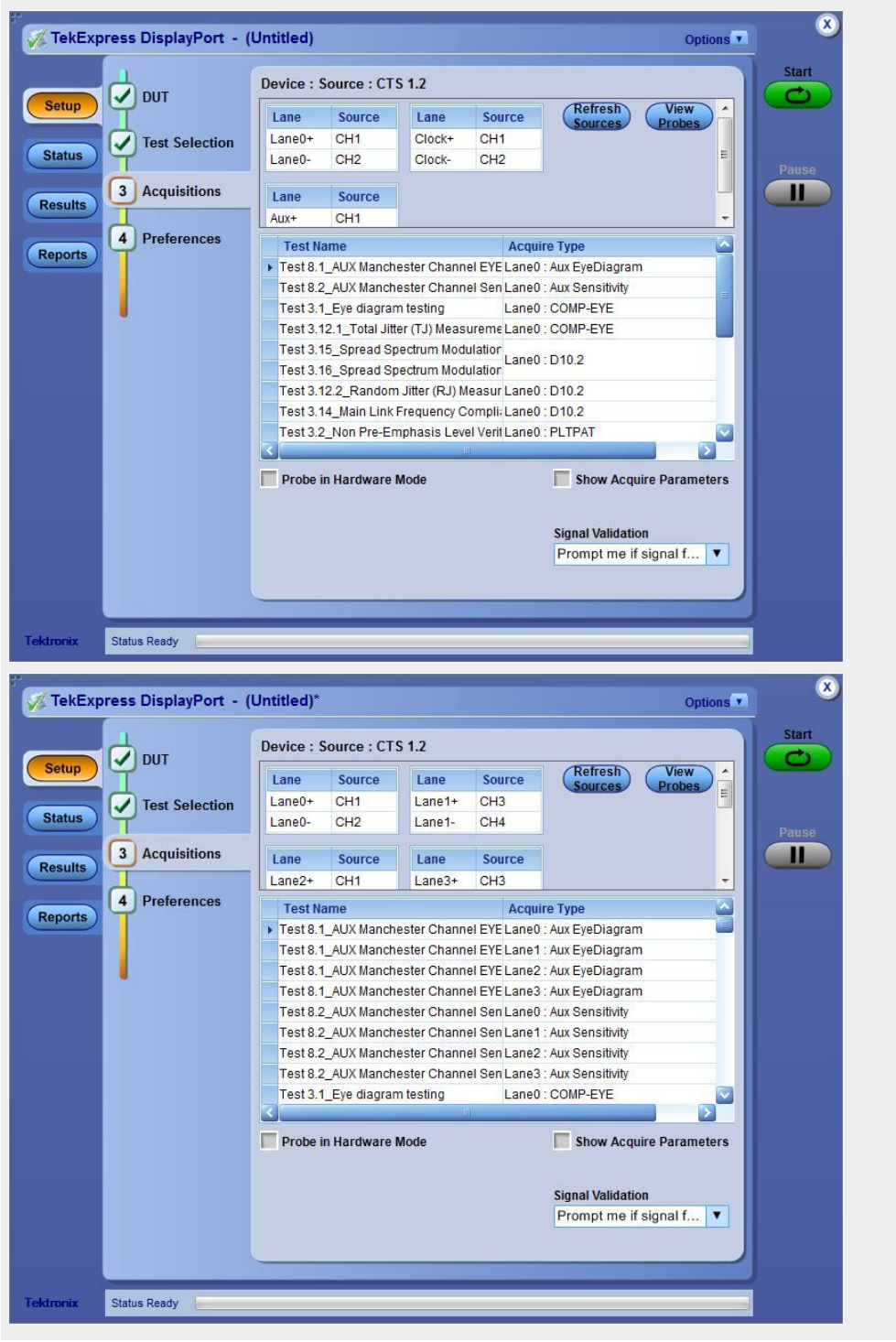
About acquisitions

Use the Acquisitions tab in the Setup panel to view *acquisitions parameters* for the selected tests.

The information on the Acquisitions tab is specific to the tests selected in the Test Selection tab and gets updated every time you change the selected tests. Options available also depend on whether you selected to *acquire live waveforms* or *use prerecorded waveform files* in the DUT tab.

The top section of the Acquisitions tab is the source selection area. The information that appears here depends in part on which type of setup you are using. For example, if you are using differential probes without RF Switch automation, the source area will display tables of the lanes and their sources, and you can view probe information. If you are using an RF Switch setup, the Automate with RF Switch check box is displayed with a Setup button for accessing the RF Switch setup window.





Below the source area is the test acquisition and mapping table. If the same acquisition can be used for multiple tests, then the test acquisitions table will show these rows as merged. Additional acquisition options appear below the table.

Table 8: Acquisitions table

Column name	Function
Test Name	Displays the name of the selected test that the acquisitions apply to. One or more tests can perform the same acquisitions.
Acquire Type	Lists the source of the acquisition. For example, Lane0: PRBS7–InterPair means the data is from Lane0 and the signal type is PRBS7–InterPair.
Acquisition	Updates the location of the named acquisition
(Parameter Name)	Displays a column for each parameter available in the test. Cells in the column might display the status of whether the measurement was included in the measurement, supported by the test, or the parameter setting.
Waveform FileName	Lists the name of the waveform files used for the test. Select waveform files by clicking the button in the row. This allows you to select any waveform file using the standard File Open window. This applies to prerecorded waveforms only.

NOTE. *For DP++ tests, you must select sources for Clock lane and Data lanes. For Aux tests, select sources for AUX lane.*

See Also. *Select acquisitions*

About setting up tests

Select acquisitions

Use the Acquisitions tab in the Setup panel to view and set acquisition parameters for selected tests. Options available depend on whether you are using a live waveform or a prerecorded waveform file, and whether you are using acquisition automation (RF switch).

1. In the Setup panel, click the **Acquisitions** tab.
2. If you selected to use a pre-recorded waveform file, in the Acquisitions table, scroll to the Waveform FileName column. For each acquire type row, click the ellipsis button () and select the desired waveform file or files.

3. If you selected to use live waveforms and no RF switch automation, then the lane and channel selection tables are displayed near the top of the tab. The selections displayed are based on the available probe setup.
   - To see which probes are connected to which channels, click **View Probes**.
   - To refresh the probe configuration information displayed after changing any probes, click **View Probes** and then click **Refresh** in the Probe Configuration dialog box.
   - To change a lane source, click in the Source column and select a channel from the drop-down list.
4. If you selected to use live waveforms with RF switch automation, then the Automate with RF Switch check box will be visible and selected. Map the lanes to the relay locations.

   a. Click the **Setup** button.

   b. In the RF Switch Setup dialog box, map the lanes to the Relay locations and then click **OK**.

5. Select any other desired *acquisition options*.

See Also. *Configure tests*

About setting up tests

Test setup overview

Acquisitions parameters

Available parameters depend on the selected mode (Compliance or User Defined) and the type of waveform being used, live or prerecorded.

Table 9: Acquisitions options

Option	Description
Probe in Hardware Mode	Disables DSP filtering and enables raw data acquisition at Scope Bandwidth
Show Acquire Parameters	When selected, the acquisition parameters for each test display in the Acquisitions table columns
Signal Validation	Determines what the application does if it is unable to validate the signal ■ Prompt me if signal fails ■ Skip test if signal fails ■ Use signal as is–don't check

See Also. *About acquisitions*

Select acquisitions

Use prerecorded waveforms for analysis

Use prerecorded waveform files to run saved tests or to run a new test using a saved waveform from another test.

Run a test in prerecorded mode.

Run a saved test or new test using a prerecorded waveform file.

1. Open a saved test setup or create a new one.

   NOTE. *If using a saved test and you do not want to overwrite the previous data, save this test setup using the Save As option before proceeding.*

2. In the Setup panel, select the **DUT** tab and then select **Use pre-recorded waveform files**. (This selection adds the Waveform FileName column to the Acquisitions table in the Acquisitions tab.)
3. *Change any other DUT parameters* as desired.
4. In the Test Selection tab, select the desired test(s).
5. In the Acquisitions tab, locate the row for the desired test. In the Waveform FileName column, click the ellipsis button () and then select a single waveform file or multiple waveform files.
6. Perform any additional test needed, such as *test configuration*, and then click **Start**.

See Also. *Before you click start*

Pre-Run checklist

About acquisitions

Acquire live waveforms for analysis

Use these instructions to set up acquisition for a live waveform.

1. *Open a saved test setup* or *create a new one*.
2. In the Setup panel, select the **DUT** tab, enter the desired DUT ID in the corresponding field and then select **Acquire live waveforms**.
3. *Select other DUT parameters* as desired.
4. In the Test Selection tab, *select the desired test(s)*.
5. In the Acquisitions tab, do any of the following that apply:
   - Select the data sources for the lanes.
   - If you are using an RF switch, *map the lanes to the relay locations*.
   - Select other desired *acquisitions parameters*.
6. *Configure the tests* if you have not done so already, and then click **Start** to run the test.

NOTE. *For DP++ tests, you must select sources for Clock lane and Data lanes. For Aux tests, select sources for AUX lane.*

See Also. *About acquisitions*

Select acquisitions

Configuring tests

About configuring tests

Use configuration settings to view and edit the analysis parameters for selected tests.

The Configuration tab shows Global parameters and Measurement parameters. Global parameters are common for all tests. Measurement parameters are specific to the test selected in the tree view section of the configuration settings. Test-specific parameters can include acquisition, analysis, and limit parameters, and certain oscilloscope settings. Whether you can configure settings depends on which mode you selected, Compliance or User Defined. For more information, see *About setting up tests*.

NOTE. *If you selected the Compliance View in the DUT tab, you will access test configuration settings using the Configure button in the Test Selection tab. If you selected Advanced View in the DUT tab, access test configuration settings from the Configuration tab in the Setup panel.*

See Also. *Configure tests*

About running rests

Measurement parameter descriptions

Configure tests

Use these instructions for configuring tests.

1. Do one of the following:
   - If you selected Compliance View in the DUT tab, then in the Test Selection tab, select the desired test in the list and then click **Configure**.

     NOTE. *To return to test selection from the configuration section, click the button.*

   - If you selected Advanced View, click the **Configuration** tab in the Setup panel.

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

   ---

2. Modify *parameter settings* as desired:

- To use Compliance Mode values, select **Compliance Mode**.

  NOTE. *You cannot change test parameters in Compliance mode, but you can view the selected parameter settings.*

- To select the instrument for testing, click **Global Settings**. In the Instruments Detected section, click the drop-down arrow and then select the desired instrument from the list. If you do not see the desired instrument in the list, *refresh the list*.

See Also. *Measurement parameter descriptions*

Select analysis and test notification preferences

Before you click start

About setting up tests

Common test parameters and values

The following table lists the measurement parameters common to most tests.

Table 10: Common parameters and values

Parameter type	Parameter and default value
Mode	Determines whether test parameters are in compliance or can be edited ■ Compliance: Most test parameter values cannot be edited. ■ User Defined: Most test parameter values can be edited.
Global Settings	These settings apply to all tests selected for the current session. ■ RF Switch: Specifies the RF Switch instrument being used. If RF switching will not be used, the default is “Do not use.” ■ Real Time Scope: Specifies the oscilloscope used for tests.
Measurements	Shows the parameter settings for the test selected in the tree view section. For details, see *Measurement parameter descriptions*.
Limits	Shows the upper and lower limits for the applicable measurement using different types of comparisons. In Compliance Mode, allows you to view the measurement high and low limits used for the test selected in the tree view of the Measurements tab. When running tests in User Defined Mode, you can edit the limit settings in the Limits Editor.  **Figure 1: When running tests in User Defined Mode, the cells in the Limits Editor table are active for editing so you can change parameters.**

See Also. *Configure tests*

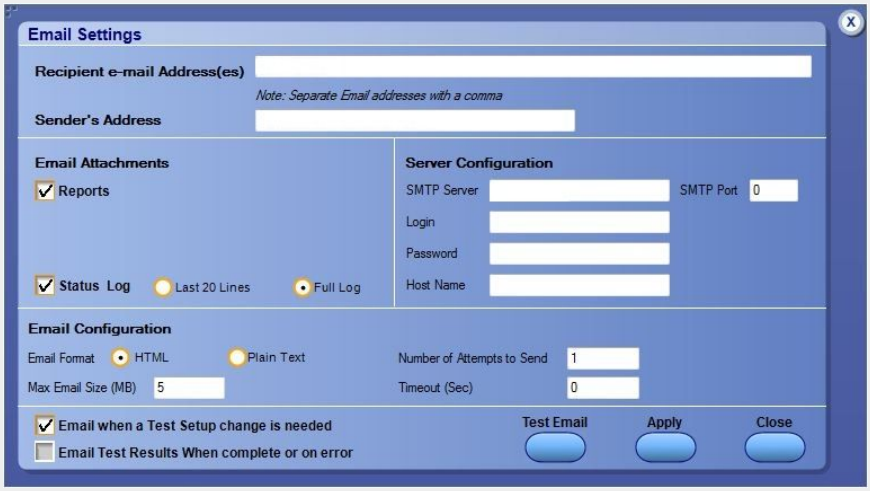
Select acquisitions

Measurement parameter descriptions

Configure email notification

To be notified by email when a test completes, fails, or produces an error, configure email settings. Configure these settings from the Setup panel Preferences tab or from the application Options menu.

Email Settings



1. From the Options menu, select **Email Settings** to open the Email settings dialog box.
2. (Required) For Recipient email Address(es), enter your email address. To include multiple addresses, separate the addresses with commas.
3. (Required) For Sender's Address, enter the email address used by the instrument. This address consists of the instrument name followed by an underscore followed by the instrument serial number, then the @ symbol and the email server 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**: Select to receive the test report with the notification email
   - **Status Log**: Select to receive the test status log with the notification email. If you select this option, then also select whether you want to receive the full log or just the last 20 lines.

6. In the Email Configuration section, select as desired:
   - Select the message file format to send: HTML (the default) or plain text.
   - Enter a maximum file size for the email message. Messages with attachments larger than this limit will not be sent. The default is 5 MB.
   - To limit the number of attempts the system makes to send a notification, enter the number in the Number of Attempts to Send field. The default is 1. You can also specify a timeout.
7. Select the conditions for which you want to be notified by email:
   - **Email when a Test Setup change is needed**. Select this option to receive an email notification to change the probe settings on the test setup.
   - **Email Test Results When complete or on error**. Use this check box to enable or disable email notifications.
8. To test your email settings, click **Test Email**.
9. To apply your settings, click **Apply**.
10. Click **Close** to exit the Email Settings dialog box.

See Also. *Configure tests*

Select analysis and test notification preferences

Select analysis and test notification preferences

Select how you want to be notified by the system when a test measurement fails:

1. In the Setup panel, click the **Preferences** tab.
2. To have the application highlight any test results that are within a certain margin, in the Analysis Options section, select the **Highlight yellow warning if measured value is within __% margin of limit** check box. Enter a limit margin in the corresponding field.
3. To set a data rate tolerance for signal validation, enter a percentage in the corresponding field.
4. In the Email Options section, select any of the following options:
   - **On Test Failure, stop and notify me of the failure**: Select this option to have the test stop and notify you by email if the test fails.
   - **Send Email on Test setup change**: Select this option to have the application email you a reminder to change the probe settings in the test setup.
5. If you select an email option in step 4, then click **Email Settings**. In the Email Settings dialog box, select the **Email Test Results When complete or on error** check box. *Configure the other email settings* if you have not already done so.

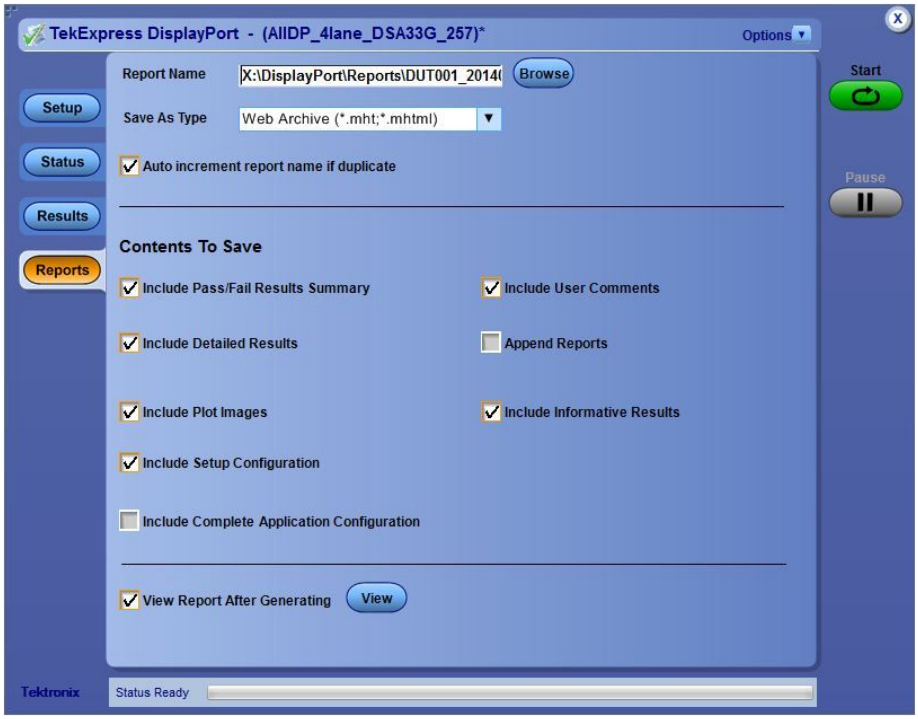
See Also *Select report options*

About setting up tests

Select report options

Select which test information to include in the report and the naming conventions to use for the report. For example, if you do not want your test results to be overwritten each time you run any test, always give the report a unique name or select to have the name increment each time you run a particular test. Generally, you would select report options before running a test or when creating and saving test setups. Report settings are included in saved test setups.

1. Select the **Reports** panel.



2. Select from the report options. Refer to the following Report Options table for details.

Table 11: Report options

Setting	Description
Report Name	Displays the default report name and location where the report is saved when generated. The report and related files are stored in the Untitled Session folder under the DisplayPort Tx folder by default. The report file in this folder gets overwritten each time you run a test unless you save the test under a unique name after running it, or select to auto increment the report name. Change the report name or location. In the Report Name field, type over the current folder path and name and then save the test setup. 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\DisplayPort Tx\DUT001_group1.mht. ***NOTE.*** *You cannot change the file location using the Browse button.* Open an existing report. Click **Browse**, locate the report file, then click **View** at the bottom of the panel.
Save As Type	Saves a report in a file type different from the default. Lists supported file types to choose from ***NOTE.*** *If you select a file type different from the default, be sure to change the report file name extension in the Report Name field to match.*
Auto increment report name if duplicate	If the application finds a report with the same name as the one being generated, the application automatically increments the name of the report. For example: DUT001, DUT002, DUT003.This option is enabled by default.
Include Pass/Fail Results Summary	The colored block labeled Test Result indicating whether the test passed or failed will be included in the report. For details, see Report Contents in *View a report*.
Include Detailed Results	Report will include parameters measured, such as Margin, Low Limit, High Limit, Compliance Mode Yes/No, Execution Time, and Comments
Include Plot Images	Screen shots captured from the oscilloscope during test execution that show the waveform and measurement data

Setting	Description
Include Setup Configuration	Information about hardware and software used in the test will be included in the summary box at the beginning of the report. Information includes: the oscilloscope model and serial number, probe model and serial number, firmware version, SPC and factory calibration status, and software versions for applications used in the measurements.
Include Complete Application Configuration	A table listing general, common, and acquired parameters used in the test will be included in the report. This option is disabled by default.
Include User Comments	Select to include any comments about the test that you or another user added in the DUT tab of the Setup panel. Comments appear in the Comments section under the summary box at the beginning of each report.
View Report After Generating	Automatically opens the report in your Web browser when the test completes. This option is selected by default.
Append Reports	This option adds new report data to the end of an existing report of the same name.
Include informative results	Include or exclude informative results in reports. This needs to be selected before you run the tests for it to get applied.

See Also

View a report

About setting up tests

Running tests and viewing results

About running tests

After selecting and configuring tests, review the *pre-run checklist* and then click **Start** to perform the tests. The application acquires and analyzes the data, then 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 using the Alt+Tab key combination. To keep the TekExpress DisplayPort Tx application on top, select **Keep On Top** from the DisplayPort Tx Options menu.

Waveforms generated during the test are automatically saved when you save the related test setup, so you can *run tests in prerecorded mode*.

See Also

Before you click start

About configuring tests

About setting up tests

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 DisplayPort Tx, it creates the following folders on the oscilloscope:

- Windows XP: \Program Files\Tektronix\TekExpress**TekExpress DisplayPort Tx**
- Windows 7: \Program Files (x86)\Tektronix\TekExpress**TekExpress DisplayPort Tx**
- \My Documents\My TekExpress**DisplayPort Tx**
- \My Documents\My TekExpress\DisplayPort Tx**Untitled Session**

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

CAUTION.

Do not modify any of the session files or folders because this may result in lost data or corrupted session files. Each session has multiple files associated with it. When you save a session, a .TekX file, and a folder named for the session and containing the associated files, is created on the oscilloscope X: drive.

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

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

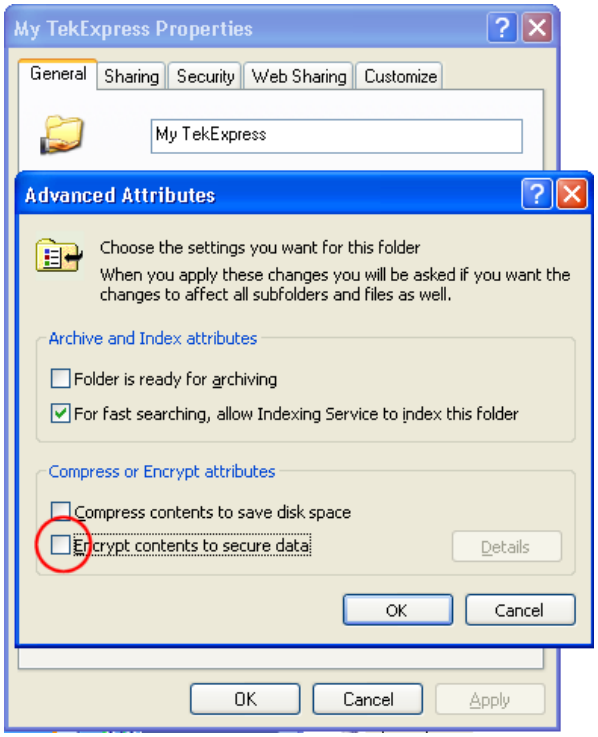
   NOTE. *If the X: drive is mapped to any other shared folder, the application will display a warning message asking you to disconnect the X: drive manually.*

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

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

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

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



4. See the *pre-run checklist* before you run a test.

See Also

View test-related files

Application directories and usage

File name extensions

Pre-Run checklist

Do the following before you click Start to run a test. If this is the first time you are running a test on a setup, refer to the information in *Before you click ctart*.

1. Make sure that all the required instruments are properly warmed up (approximately 20 minutes).
2. Perform the Signal Path Compensation (SPC).
   a. On the oscilloscope main menu, select the **Utilities** menu.
   b. Select **Instrument Calibration**.
3. *Deskew channels*.
4. Make sure that the application is able to find the DUT. If it cannot, perform a *search for connected instruments*.
   a. Launch the DisplayPort Tx application.
   b. Select the **Setup** panel and then click the **Test Selection** tab.
   c. Select any test and then click **Configure**.
   d. In the Configuration section, click **Global Settings**.
   e. In the Instruments Detected section, click the drop-down arrow to the right of Real Time Scope and make sure that the correct DUT is listed.

See Also *Equipment connection setup*

Run a test in prerecorded mode

This option allows you to rerun a complete test using just the oscilloscope and the saved test setup files.

NOTE. *Running the test in prerecorded mode does not replace the existing saved test results. Instead, if you save the test under the same name, the results are saved in a session folder named for the date and time of the session.*

1. *Open the desired saved test setup*.
2. In the Setup panel, select the **DUT** tab and then select **Use pre-recorded waveform files**.

   A Run session field displays, showing the session that ran for this test. If there is more than one session that contains test data (not an empty session folder) these are available in the Run session drop-down list. Select the desired session from the drop-down list.
3. Select the **Acquisitions** tab.
4. In the *Acquisitions table*, look in the Waveform FileName column.

5. If no waveform files are listed, click the ellipsis button () and then locate and select the desired waveform files. Skip to step 7.
6. If waveform files are listed, verify that the listed files are the ones you want to use. If they are not, or if you want to add a file, do any of the following as needed:
   - To remove a waveform file from the list, click the ellipsis button and select **Remove Waveform File**.
   - To replace a waveform file, click the ellipsis button, select **Replace Waveform File**, then locate and select the desired file.
   - To add a waveform file, click the ellipses button, select **Select Waveform Files**, then locate and select the waveform files.
7. Click **Start** to run the test.

See Also *Before you click start*

About acquisitions

View the progress of analysis

The *Status panel* displays a record of the test as it executes. By default, the application switches to this panel after you click the Start button. Choose from the following two views by selecting the named tab, even while a test is in progress.

Test Status

The Test Status tab presents a collapsible/expandable table with information about each test as it is running.

Table 12: Status tab table

Column	Description
Test Name	Name of the test
Acquisition	The type of pattern data being acquired
Acquire Status	Progress state of the acquisition: ■ To be started ■ Completed Acquisition ■ Prerecorded waveform
Analysis Status	Progress state of the analysis: ■ To be started ■ Testing ■ Completed ■ Aborted

Log View

The Log View tab provides a list of actions that happen as the test executes. Use this tab to monitor, review, or troubleshoot tests.

Table 13: Log View options

Item	Description
Message History	This window timestamps and displays all run messages.
Show Detailed Log	Select this check box to record a detailed history of test execution.
Auto Scroll	Select this check box to have the program automatically scroll down as information is added to the log during test execution.
Clear Log	Click this button to clear all messages in the Message History window.
Save	Use to save 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*

View test results

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

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

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

).

See Also *View a report*

View the progress of analysis

View a report

After the analysis, a report is automatically generated. By default, the report is displayed in your 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 you if want to view a different test report, then do the following:

1. In the Reports panel, click the **Browse** button to the right of the Report Name field and locate the desired report.

   The path to the report displays in the Report Name field.

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

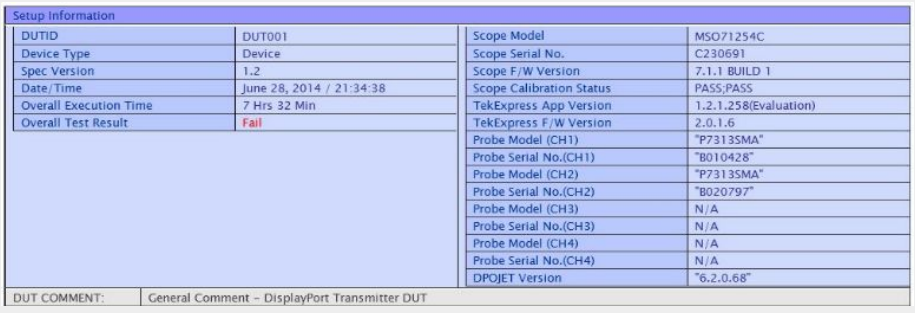
For information on changing the file type, file name, and other report options, see *Select report options*.

Report Contents

In the report, a table shows the test name, measurement details, test result (pass/fail), mode, and analysis time.

Setup configuration information

Setup configuration information is listed in the summary box at the beginning of the report. This information includes the oscilloscope model and serial number, probe model and serial number, software versions, and more. To exclude this information from a report, clear the **Include Setup Configuration** check box in the Reports tab before running the test. If you selected to include comments in the test report, any comments you added in the DUT tab are shown at the end of the setup information summary.



Setup Information			
DUTID	DUT001	Scope Model	MSO71254C
Device Type	Device	Scope Serial No.	C230691
Spec Version	1.2	Scope F/W Version	7.1.1 BUILD 1
Date/Time	June 28, 2014 / 21:34:38	Scope Calibration Status	PASS;PASS
Overall Execution Time	7 Hrs 32 Min	TekExpress App Version	1.2.1.258(Evaluation)
Overall Test Result	Fail	TekExpress F/W Version	2.0.1.6
		Probe Model (CH1)	"P7313SMA"
		Probe Serial No.(CH1)	"B010428"
		Probe Model (CH2)	"P7313SMA"
		Probe Serial No.(CH2)	"B020797"
		Probe Model (CH3)	N/A
		Probe Serial No.(CH3)	N/A
		Probe Model (CH4)	N/A
		Probe Serial No.(CH4)	N/A
		DPOJET Version	"6.2.0.68"
DUT COMMENT:	General Comment – DisplayPort Transmitter DUT		

Pass/Fail summary

The Pass/Fail Summary column indicates whether a test passed or failed. If the test passed, the column cell is green. If the test failed, it is red. To exclude this information from a report, clear the **Include Pass/Fail Results Summary** check box in the Reports tab before running the test.

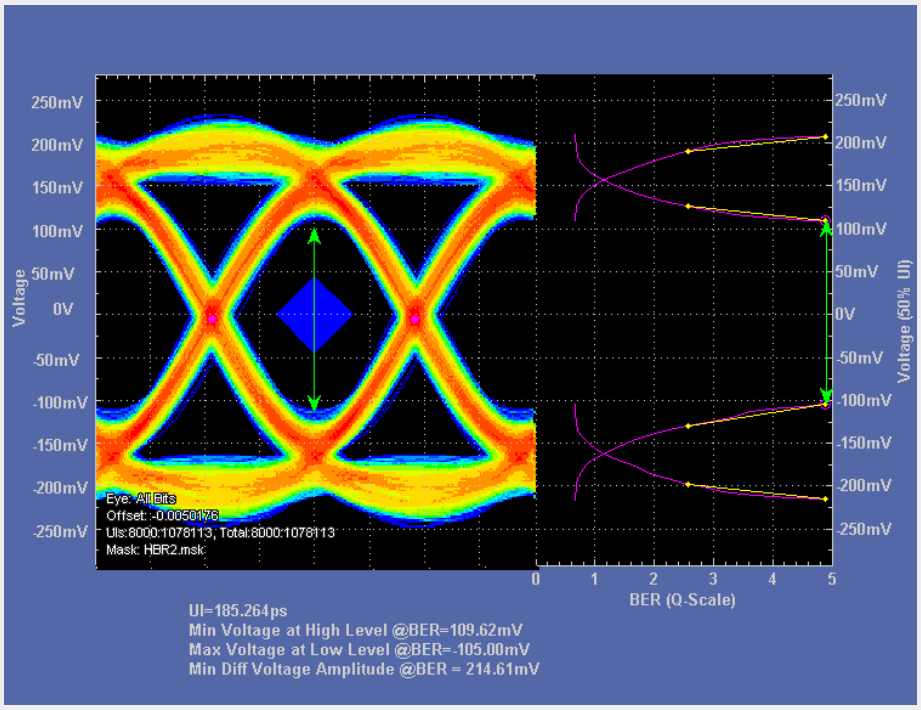
Detailed results

Includes detailed measurement information.

To exclude this information from the report, clear the **Include Detailed Results** check box in the Reports tab.

Plot images

Plot images are screen shots captured from the oscilloscope during test execution. They show waveform and measurement data.



To exclude plot images from a report, clear the **Include Plot Images** check box before running a test.

Application configuration

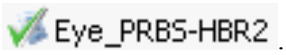
This table lists general, common, and acquired parameters used in the test. General parameters are those selected to be included in all tests (where applicable). General parameters, are set in the DUT tab. Acquire parameters are those that were actually used for the acquired pattern during the test. To exclude this information from a report, clear the **Include Complete Application Configuration** check box before running a test.

See Also *View test results*

View test-related files

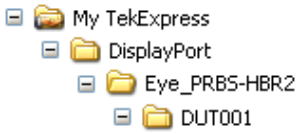
View test-related files

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

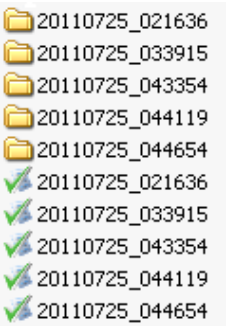
For example, 

.

Inside the test setup folder is another folder named for the DUT ID used in the test sessions. In the example below, the test setup folder is Eye_PRBS-HBR2 and the DUT 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.



Each session folder contains image files of any plots generated from running the test session, and any waveform files if prerecorded waveform files were used during the session.

The first time you run a new, unsaved session, the session files are stored in the Untitled Session folder located at ..\My TekExpress\DisplayPort Tx. Once you name and save the session, the Untitled Session folder name is changed to the one you specified.

NOTE. *By default, test report files are saved in the session folder. You can change where the report file is saved for a specific test session.*

See Also

File name extensions

Before you click start

Algorithms

About algorithms

For all measurements, use the following guidelines to set up the oscilloscope.

Oscilloscope Setup Guidelines

For all measurements, use the following guidelines to set up the oscilloscope:

- The signal is any channel, reference, or math waveform.
- Set the vertical scale for the waveform so that the waveform does not exceed the vertical range of the oscilloscope.
- Set the sample rate to capture enough waveform detail and avoid aliasing.
- Longer record lengths increase measurement accuracy but the oscilloscope takes longer to measure each waveform.

Acquire the signal in accordance with the SFF-8431 specification. For this reason, the oscilloscope setup differs for each measurement.

Oscilloscope Setup Procedure

Before making any measurements, perform the following steps to ensure accurate measurement:

1. Allow at least 20 minutes warm-up time for oscilloscope.
2. Run the oscilloscope SPC calibration routine. Remove all probes from the oscilloscope before running SPC.
3. If using probes, perform the probe calibration defined for the specific probes being used.
4. Perform deskew to compensate for skew between measurement channels. Select **Off** for the “Display Only” control on the Deskew setup window to make sure that the deskew data is stored with any waveforms that are stored.

See Also

Before You Click Start

3.1 Eye diagram testing (Normative)

This test evaluates the waveform to ensure that timing variables and amplitude trajectories support the overall DisplayPort system objectives of Bit Error Rate in data transmission.

Measurement Algorithm

Depending on the data rate selected, the points at which the measurement is taken are selected. For HBR/RBR, the point of measurement is TP2. For HBR2, the point of measurement is TP3_eq. TP3_eq point is actually simulated from TP2. The TP2 data is first passed through Compliance Interconnect Channel (CIC) in software that emulates a “worst-case HBR cable” with an insertion loss that emulates the characteristics described by VESA. This point is called TP3. A CTLE filter is applied to the waveform to undo the channel effects. The resultant is called the TP3_eq point.

The algorithm runs as follows:

1. Collect data at the relevant point. (TP2 for RBR/HBR and TP3_eq for HBR2)
2. Call DPOJET. Run Mask Hits measurement on DPOJET. For HBR2 speed it will generate Plotstate.mat, which provides the edges and other information.
3. For RBR/HBR report result from DPOJET results. For HBR2 get the sample rate and symbol rate (UI) from the Plotstate.mat.
4. Find the number of samples per UI.
5. Find the fraction of UI points where the eye height needs to be measured. For example, if the eye height needs to be measured between 0.25 of UI and 0.75 UI. Get the UI points between 0.25 to 0.75 in steps (for example, 20 steps between 0.25 to 0.75).
6. Get the index (waveform sample number) of the UI point where the EYE height needs to be measured.
7. Get the sample value before and after the index point.
8. Find the time difference between the references clock (the exact time at which the sample needs to be found) and extracted clock (from the indexes calculated in steps 6 and 7 above).
9. Estimate the sample value for the exact time. A sample might not exist right at the exact point, so the sample is estimated by performing linear interpolation using the samples taken before and after the time at which the sample is needed.
10. From the estimated sample values, get the sample values that are above the waveform threshold (High level).
11. From the estimated sample values, get the sample values that are below the waveform threshold (High level).
12. Find the histogram of data for the high level sample values.
13. From the histogram, find the first value (high level amp) that is not empty in the histogram.
14. Find the histogram of data for the low level sample values.

15. From the histogram, find the last value (low level amp) that is not empty in the histogram.
16. The eye height is the difference between the high-level amplitude and the low-level amplitude.

The eye height is calculated at any passing location between 0.375 UI and 0.625 UI.

For HBR2, the test is a pass if the measurement passes at any of the allowed voltage and preemphasis levels.

3.2 Non Pre-Emphasis level verification testing (Normative)

This measurement finds the ratio between two Peak-to-Peak voltage readings, each taken at different Nominal level settings with the Nominal 0 dB pre-emphasis setting. The Peak-to-Peak voltage is calculated at all possible voltage levels.

Measurement Algorithm

Peak-to-Peak Voltage is defined as:

$Voltage_{Peak\text{-}Peak} = V_{H\ Npatterns} - V_{L\ Npatterns}$

Where:

- $V_{H\ Npatterns}$ is the composite average of High differential voltage.
- $V_{L\ Npatterns}$ is the composite average of Low differential voltage.

This algorithm runs as follows:

1. Acquire the qualifying patterns:
   - HBR and RBR: 1-0-1-0-1-1-1-1-1-1-1 for the High differential voltage, and 1-0-1-0-0-0-0-1 for the Low differential voltage
   - HBR2: 1-1-1-1-1-0-0-0-0-0 for both Low and High differential voltage
2. Calculate the High and Low differential voltage for each of the qualifying patterns:
   - For HBR and RBR, the High differential voltage is the average of the High or “one” voltage over three UIs ending at the 50% point of the 6th bit of the seven successive transmitted ones of the pattern. The Low differential voltage is the average of the voltage over two UIs ending at the 50% point of the 4th bit of the four successive transmitted 0s of the qualifying pattern.
   - For HBR2, the High differential voltage is the voltage over three UIs ending at the 50% point of the 5th bit of the five successive transmitted ones of the qualifying pattern. The Low differential voltage is the voltage over three UIs ending at the 50% point of the 5th bit of the five successive transmitted zeroes of the qualifying pattern.
3. Calculate the average of the High and Low differential voltages over all qualifying patterns.

4. Use the average High and Low differential voltage to calculate the Peak-to-Peak voltage.
5. Calculate the resultant using the Peak-to-Peak voltage at two different levels using the formula:

   Resultant = 20 × Log10[$Voltage_{Peak\text{-}Peak}$_Level A ÷ $Voltage_{Peak\text{-}Peak}$_Level B]

3.3 Pre-Emphasis level verification testing and Maximum peak-to-peak differential voltage testing (Normative)

This test evaluates the affect of pre-emphasis of the Source waveform by measuring the peak differential amplitude to ensure accuracy of the pre-emphasis setting. Comparisons are also made for the Level 0 transition state as well as non-transition levels. Maximum differential peak to peak voltage is also tested.

Measurement Algorithm

Pre-Emphasis level verificaiton testing

Each transition and non-transition swing is composed of High and Low voltage measurements that are combined to obtain the peak-to-peak voltage. The peak-to-peak voltage is calculated as follows:

Where:

- Pre-emphasis level = L_{vlX}
- $V_{T\_LvlX\_PP} = V_{T\_LvlX\_H} - V_{T\_LvlX\_L}$
- $V_{N\_LvlX\_PP} = V_{N\_LvlX\_H} - V_{N\_X\_LvlX\_\_L}$

Where:

- $V_{T\_LX\_PP}$ = the peak-to-peak voltage at the transition bit
- $V_{N\_LVIX\_PP}$ = the peak-to-peak voltage at the non-transition bits
- $V_{T\_LvlX\_H}$ and $V_{T\_LvlX\_L}$ are the high and low voltage at the transition bit
- $V_{N\_LvlX\_H}$ and $V_{N\_X\_LvlX\_L}$ are the high and low voltage at the non transition bits

The algorithm runs as follows:

1. Acquire the qualifying pattern:
   - HBR and RBR: 0-1-0-1-1-1-1-1-1-1-1 for the High Differential Voltage, 1-0-1-0-0-0-0-1 for the Low Differential Voltage
   - HRB2: 1-1-1-1-1-0-0-0-0-0 for both Low and High Differential Voltage
2. Calculate the high and low differential voltage for each of the qualifying patterns.

3. First calculate the transition voltage values.
   - For HBR and RBR, Transition measurements are preconditioned by the first 3 bits in these patterns. For a transition to high measurement, the precondition bits are 0-1-0. For a transition to low measurement, the precondition bits are 1-0-1. The transition voltage measurements, $V_{T\_LvlX\_H}$ and $V_{T\_LvlX\_L}$, are the average value over the 40% to 70% UI points in the transition bit.
   - For HBR2, the High level and Low level measurements require a 1-1-1-1-1-0-0-0-0-0 balanced pattern. The transition voltage measurements, $V_{T\_LvlX\_H}$ and $V_{T\_LvlX\_L}$, are the average value over the 40% to 70% UI points in the transition bit.
4. Calculate the non-transition voltage values for HBR, RBR, and HBR2 in the same manner as the non Pre-emphasis measurement.
5. Calculate the following three sets of measurements:
   - Pre-emphasis values for the Level 0 are calculated as: Resultant = $20\times\text{Log}\,[V_{T\_Lvl0\_PP} \div V_{N\_Lvl0\_PP}]$ for all supported levels.
   - Pre-emphasis Delta measurements are calculated as: Level 1 vs. Level 0: Resultant = $20\times\text{Log}\,[V_{T\_Lvl1\_PP} \div V_{N\_Lvl1\_PP}] - 20\times\text{Log}\,[V_{T\_Lvl0\_PP} \div V_{N\_Lvl0\_PP}]$, for Voltage Swing Levels 0, 1, and 2.

     Level 2 vs. Level 1: Resultant = $20\times\text{Log}\,[V_{T\_Lvl2\_PP} \div V_{N\_Lvl2\_PP}] - 20\times\text{Log}\,[V_{T\_Lvl1\_PP} \div V_{N\_Lvl1\_PP}]$, for Voltage Swing Levels 0 and 1.

     Level 3 vs. Level 2: Resultant = $20\times\text{Log}\,[V_{T\_Lvl3\_PP} \div V_{N\_Lvl3\_PP}] - 20\times\text{Log}\,[V_{T\_Lvl2\_PP} \div V_{N\_Lvl2\_PP}]$, for Voltage Swing Level 0 if supported.
   - Non Transition measurement is calculated as Resultant = $\text{MIN}(V_{N\_LvlX\_PP}) \div (V_{N\_Lvl0\_PP})$ at each supported level for all supported X (Lvl1, Lvl2, and Lvl3)

Maximum peak-to-peak differential voltage testing

A differential signal is defined by taking the voltage difference between two conductors. In this Standard, a differential signal or differential pair is comprised of a voltage on a positive conductor, V_{D+}, and a negative conductor, V_{D-}.

The differential voltage (VIDIFF) is defined as the difference of the positive and the negative conductor voltages.

$V_{DIFF} = V_{D+} - V_{D-}$

The algorithm runs as follows:

1. Acquire the qualifying differential signal.
2. Calculate the peak-to-peak voltage of acquired signal.
3. Find out the maximum value of peak-to-peak voltage.

3.3.2 Post cursor2 verification testing (Normative)

This test evaluates the effect of adding Post-Cursor2 in a Source waveform by measuring the peak differential amplitude to ensure accuracy of the Post-Cursor2 setting.

Measurement Algorithm

For each voltage swing and pre-emphasis level, transition bit voltages are measured at qualified sections of the pattern. Each transition swing is composed of High and Low voltage measurements that are combined to obtain the peak-to-peak voltage. Peak-to-Peak voltage is calculated as:

- $V_{T1010\_PC2\_LvlX\_PP} = V_{T1010\_PC2\_LvlX\_H} - V_{T1010\_PC2\_LvlX\_L}$

  Where: X corresponds to the PostCursor2 Level 0, 1, 2, or 3.

- $V_{T1100\_PC2\_LvlX\_PP} = V_{T1100\_LvlX\_H} - V_{T1100\_LvlX\_L}$

  Where: X corresponds to the PostCursor2 Level 0, 1, 2, or 3.

The algorithm runs as follows:

1. Acquire at least 1,000 qualifying PCTPAT patterns. Only HBR2 data rate is supported. The PCTPAT is an 80–bit custom pattern 1111100000 1111100000 1111100000 1100110011 0011001100 1100110011 0010101010 1010101010.
   - The $V_{T1010\_PC2\_LvlX}$ measurement is calculated as the average value over the 40% to 70% UI points in the fifth relevant bit in the 1010 part of the pattern.
   - The V_{T1010_H} measurement is made on the fifth 1 in this part of the waveform, and V_{T1010_L} measurements are made on the fifth 0 in this part of the waveform.
   - The $V_{T1100\_PC2\_LvlX}$ measurement is calculated as the average value over the 40% to 70% UI points in the fifth pair of relevant bits in the 1100 part of the pattern. V_{T1100_H} measurements are made on the fifth 11 in this part of the waveform, and V_{T1100_L} measurements are made on the fifth 00 in this part of the waveform.
2. Calculate the resultant value as Resultant = [$V_{T1100\_PC2\_LvlX\_PP} \div V_{T1010\_PC2\_LvlX\_PP}$] for all voltage swing and Pre-emphasis level combinations.

3.4 Inter-Pair skew testing (Normative)

This measurement finds the skew, or time delay, between respective differential data lanes in the DisplayPort interface.

Measurement Algorithm

The Inter-Pair Skew is obtained by finding the Inter-Lane skew for one acquisition and then averaging it over multiple acquisitions.

The algorithm runs as follows:

1. Simultaneously capture signals on two lanes on two single-ended measurement channels where each lane is composed of 2 single-ended signals, positive (D+) and negative (D–).
2. On both waveforms, locate the common point in PRBS7 sequence and measure the time difference between the corresponding edges at the transition point. Locate each edge by determining when the waveform crosses the transition amplitude where VTrn = 0 V.
3. Calculate Inter-Pair Lane skew as:

   Inter-Lane Skew = $\{1 \div \text{NumEdges}\}\Sigma \mid T_{\text{Transition\_LaneA}} - T_{\text{Transition\_LaneB}} \mid -$ Nominal Skew, where Nominal Skew is the expected offset between the two lanes.

3.5 Intra-Pair skew testing (Normative)

This measurement calculates the skew, or time delay, between differential data lanes in the DisplayPort interface.

Measurement Algorithm

The algorithm runs as follows:

1. Simultaneously capture the single-ended waveforms of both polarities (+ve and –ve) for one lane.
2. Compare the rising edge of the data true signal (D+) with the D– lane's falling edges. Compare the rising edge of the D– lane with the falling edge of the D+ signal.
3. Find the time of transition by determining when the waveform crosses the transition amplitude.
4. For each lane composed of 2 single-ended signals D+ and D–, find V_H and V_L for single-ended signals D+ and D– by finding the average value over the 0.60 to 0.75 UI region past the edge.
   a. For D+, measure V_{H+} and V_{L+}.
   b. For D–, measure V_{H-} and V_{L-}.
5. Calculate $V_{\text{Transition\_Data+}} = \{V_{H+} + V_{L+}\} \div 2$ and $V_{\text{Transition\_Data–}} = \{V_{H-} + V_{L-}\} \div 2$
6. The Intra Lane Skew is defined as:

$$\text{Intra-Lane Skew} = \{1 \div \text{NumEdges}\} \Sigma \left[\left(D+_{\text{Transition\_High}} - D-_{\text{Transition\_Low}}\right) + \left(D+_{\text{Transition\_Low}} - D-_{\text{Transition\_High}}\right) \right] \div 2$$

3.10 AC common mode noise measurements (Informative)

This measurement finds the AC common-mode noise (unfiltered rms), of the differential data line of the DisplayPort interface.

Measurement Algorithm

The measurement is calculated as follows:

1. Acquire the waveform using a differential probe.
2. Deskew the measurement channel.
3. Calculate the Vrms from DPOJET.

3.11 Non-ISI jitter measurements (Normative)

This measurement finds the amount of Non-ISI jitter in the data transmission. This measurement is only for RBR and HBR data rates.

Measurement Algorithm

This measurement is based on DPOJET and uses direct measurement provided in DPOJET for calculation of Non-ISI Jitter.

3.12.1 Total jitter (TJ) measurements (Normative)

This test evaluates the total jitter accompanying the data transmission as an explicit bit error rate of 10^{-9}. This measurement is a data time interval error (Data-TIE) jitter measurement.

Measurement Algorithm

This measurement is based on DPOJET and uses direct measurement provided in DPOJET for calculation of Total Jitter.

3.12.2 Random jitter (RJ) measurements (Normative)

This measurement evaluates the random jitter accompanying the data transmission to ensure that it does not exceed allowed amounts. This measurement is only for the HBR2 data rate.

Measurement Algorithm

This measurement is based on DPOJET and uses direct measurement provided in DPOJET for calculation of Random Jitter.

3.14 Main link frequency compliance (Normative)

This measurement finds the average data rate (average of UI values) of a waveform.

Measurement Algorithm

This measurement finds the mean of the UIs in the waveform. This measurement is calculated using the following procedure:

1. Acquire the waveform.
2. Find the zero crossing points. These zero crossing points are edges.
3. Calculate the time difference between adjacent edges.
4. Calculate the number of bits within the edge by dividing the time difference with the nominal UI (expected bit rate) and generate the actual UI.
5. Pass the UI values calculated for each bit through a second order 2 MHz low-pass filter (LPF) The LPF will remove all high frequency values (for example, it will remove values where there is an abrupt change in the UI).
6. Calculate the Mean of Filtered UI values.

The values of Nominal or Standard UI used in the algorithms are:

- RBR: 617.28 ps
- HBR: 370.37 ps
- HBR2: 185.89 ps

3.15 Spread spectrum modulation frequency (Normative)

This measurement finds the Spread Spectrum modulation frequency. If the waveform has SSC, the measurement will return the SSC frequency.

Measurement Algorithm

The algorithm runs as follows:

1. Acquire the waveform.
2. Find the zero crossing points. These zero crossing points are edges.
3. Calculate the time difference between adjacent edges.
4. Calculate the number of bits within the edge by dividing the time difference with the nominal UI (expected bit rate) and generate the actual UI.
5. Pass the UI value calculated for each bit through a second-order 2 MHz low-pass filter (LPF). The LPF will remove all high-frequency values (for example, it will remove values where there is an abrupt change in the UI). This will give the time trend (how the UI varies over time).
6. From the calculated UI values, remove the mean of all UIs (this will remove the DC from the UI trend) so that the UI values are center spread.
7. Find the zero crossing points in the UI values. Find the time difference between the adjacent alternate zero crossing points. The reciprocal of this value gives the modulation frequency.

$30\ \text{kHz} \leq f_{SSC} \leq 33\ \text{kHz}$

Where:

f_{SSC} is the Spread Spectrum Clock frequency.

3.16 Spread spectrum modulation deviation (Normative)

This measurement finds the maximum and minimum deviation of frequency from the standard (or nominal frequency).

Measurement Algorithm

The algorithm runs as follows:

1. Acquire the waveform.
2. Find the zero crossing points. These zero crossing points are edges.
3. Calculate the number of bits within the edge by dividing the time difference with the nominal UI (expected bit rate) and generate the actual UI.

4. Pass the UI values calculated for each bit through a second-order 2 MHz low-pass filter (LPF). The LPF removes all high-frequency values (values where there is an abrupt change in the UI).
5. From the calculated UI values, find the deviation of the UI values in PPM from the nominal/standard UI. The formula used to calculate the deviation is as follows:
   a. From the calculated deviation values, find the maximum deviation.
   b. From the calculated deviation values, find the minimum deviation.

The values of nominal UI used in the algorithm are:

- For RBR, Nominal UI is 617.28 ps
- For HBR, Nominal UI is 370.37 ps
- For HBR2, Nominal UI is 185.19 ps

3.17 dF-dt spread spectrum deviation HF variation (Informative)

This measurement finds the rate of change of frequency with time.

Measurement Algorithm

The algorithm runs as follows:

1. Acquire the waveform.
2. Find the zero crossing points. The zero crossing points are edges.
3. Calculate the number of bits within the edge by dividing the time difference with the Nominal UI (expected bit rate) and generate the actual UI.
4. Pass the UI values calculated for each bit through a second-order 2 MHz low pass filter (LPF). The LPF removes all high-frequency values (values where there is an abrupt change in the UI).
5. From the calculated UI values, find the deviation of the UI values in PPM from the nominal/standard UI.
6. Calculate the difference between the calculated UI deviation values to the left and right of the window.
7. From the calculated values, find the value that has maximum difference (it will give the maximum rate of change of frequency) and report.

SSCt dF/dt ≤ | 1250 | ppm/µSec variation

3.18 Dual mode TMDS clock (Normative if supported)

This measurement confirms that the TMDS clock waveform of a source operating in dual-mode does not exceed acceptable limits.

Measurement Algorithm

Confirm that the duty cycle of the TMDS clock waveform of a source operating in dual-mode does not exceed acceptable limits. Confirm that the wavefrom does not carry excessive jitter.

The algorithm runs as follows:

1. Put the device in the mode which transmits the maximum TMDS clock rate supported by the DUT.
2. Based on the frequency of clock, generate the mask.
3. Capture TMDS Clock with at least 400,000 clock periods.
4. Measure the duty cycle, Mask Hit and Jitter.

The measurement is a pass if:

- 40% < TMDS_CLOCK duty cycle < 60%
- Mask Hits = 0
- For 25 MHz ≤ TMDS Clock Frequency ≤ 165 MHz: Measured TMDS Clock Jitter ≤ 0.20 Tbit
- For 165 MHz < TMDS Clock Frequency ≤ 300 MHz: Measured TMDS Clock Jitter ≤ 120 ps

3.19 Dual mode eye diagram testing (Normative if supported)

This measurement evaluates the waveform and ensures that timing variables and amplitude trajectories meet the requirements for a dual-mode source device.

Measurement Algorithm

The algorithm runs as follows:

1. Put device in the mode which transmits the maximum TMDS clock rate supported by the DUT and capture clock as well as data on all the lanes simultaneously.
2. Select and run Mask Hits test and TIE measurement from DPOJET.
3. Evaluate results for over 400,000 UI.

The measurement is a pass if

- Mask Hits = 0
- 25 MHz ≤ TMDS Clock Frequency ≤ 165 MHz: Data Jitter ≤ 0.25 Tbit
- 165 MHz < TMDS Clock Frequency ≤ 300 MHz: Data Jitter ≤ 150 ps

8.1 AUX channel eye diagram testing

This measurement checks if AUX channel waveform timing variables and amplitude trajectories support the DisplayPort system.

Measurement Algorithm

This measurement checks if the Eye opening for the data transmitted by the Aux channel of the DUT is within limits. As the data comes in as a burst, cursors are used. The following are the steps for the measurement:

1. Initiate an AUX transaction from the DUT by using DP-AUX. Capture the burst of data transmitted by the DUT.
2. Find the edges from the burst. Edges are found such that at least 38 edges are available measurement (16 SYNC pulses, SYNC End,4 data bits).
3. Mask Hit measurement and eye height measurement is run using DPOJET.
4. Steps 1-3 are repeated 10 times and results are picked up from DPOJET.

The measurement is a pass if Eye height > 290 mV and there are no mask hits.

Procedure

1. Connect the VTM AUX test fixture to the DP interface port on the source DUT.
2. Power the VTM fixture by connecting the USB-Mini USB cable to one of the USB ports on the oscilloscope.
3. Bypass the AUX buffers on the VTM test fixture, using jumpers between pins 2 & 4 on both J14 & J15.
4. Connect the Tektronix DP Source test fixture to the female DP connector of the VTM test fixture.
5. Using a set of matched SMA cables, connect the SMA outputs for AUX from the Tektronix Source test fixture to the inputs of the Tektronix DP-AUX controller, being careful to connect the positive and negative sides of the AUX channel correctly.
6. Plug the differential probe into CH1 of the oscilloscope.
7. Using an adapter, connect the probe inputs to AUX-P and AUX-N pins on the VTM AUX test fixture, J2 pins 1 and 3.
8. Connect the HPD (Hot Plug Detect) ribbon cable between the Tektronix DPAUX controller and the Tektronix Source test fixture.
9. Connect the USB cable between the Tektronix DP-AUX controller and the USB port on the oscilloscope.

8.2 AUX channel eye sensitivity test

This measurement evaluates the sensitivity to the AUX CH EYE opening of the DUT.

Measurement Algorithm

The test verifies the EYE sensitivity of the AUX receiver in the DUT, by controlling the voltage swing of the AUX transmitter in the Reference device. The Reference device shall set the AUX differential voltage swing to 280 mV_diff_pp at its connector. Stimulus shall be applied to either the Reference device or DUT to cause AUX CH transactions to occur. This stimulus shall not be included in or affect the measurement.

The algorithm runs as follows:

1. Set Reference Source AUX channel voltage swing to a level between 240 mV and 280 mV_diff_pp as measured at its connector.
2. Induce an AUX transaction using DP-AUX to setup the DUT to transmit a specific pattern.
3. Determine whether AUX channel communication takes place successfully by checking if the requisite pattern was transmitted.
4. Repeat steps 2 and 3 twenty (20) times.

The measurement is a Pass if no errors are observed.

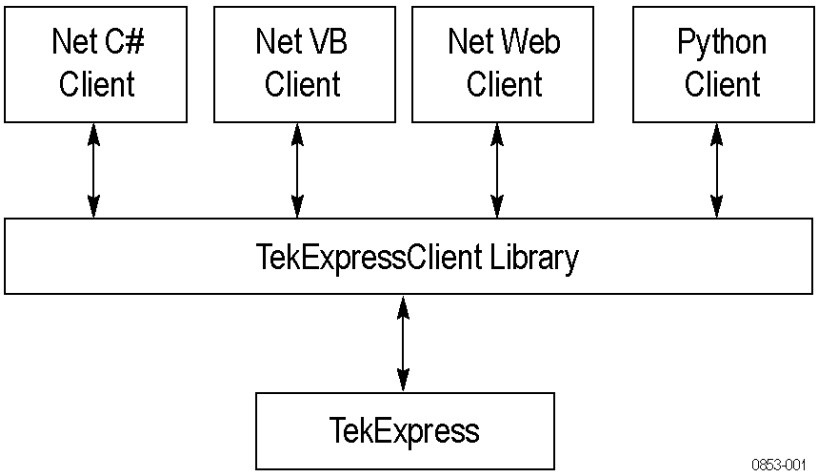
Procedure

1. Plug in Reference Source/Sink to DUT with a certified HBR 2 meter cable.
2. Set the Reference Source/Sink AUX Channel voltage swing to a level between 240 mV and 280 mV_diff_pp as measured at its connector.
3. Induce an AUX transaction using DP-Aux to setup the DUT to transmit a specific pattern. Determine whether AUX channel communication takes place successfully by checking if the requisite pattern was transmitted.

TekExpress programmatic interface

About the programmatic interface

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



The following terminology is used in this section to simplify description text:

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

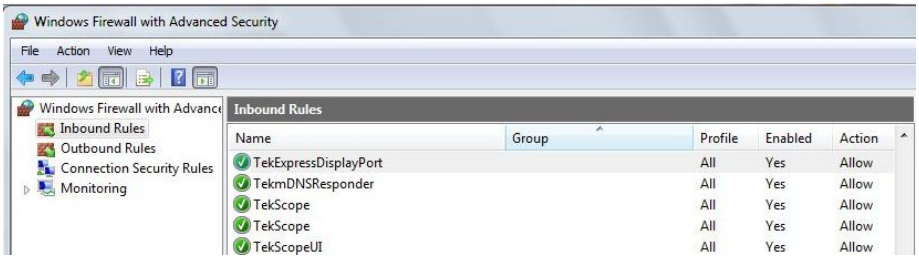
TekExpress leverages .Net Marshalling to enable the Programmatic Interface for TekExpress Client. TekExpress provides a client library for TekExpress clients to use the programmatic interface. The TekExpress client library is inherited from .Net MarshalByRef class to provide the proxy object for the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

See also *Requirements for developing TekExpress client*

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:

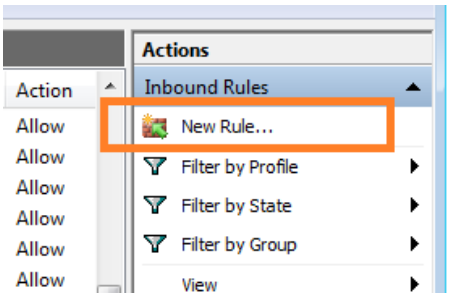
1. Access the Windows Control Panel and open the Windows Firewall tool (**Start > Control Panel > All Control Panel Items > Windows Firewall**).
2. Click **Advance Settings > Inbound Rules**.
3. Scroll through the **Inbound Rules** list to see if the following items (or with a similar name) are shown: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

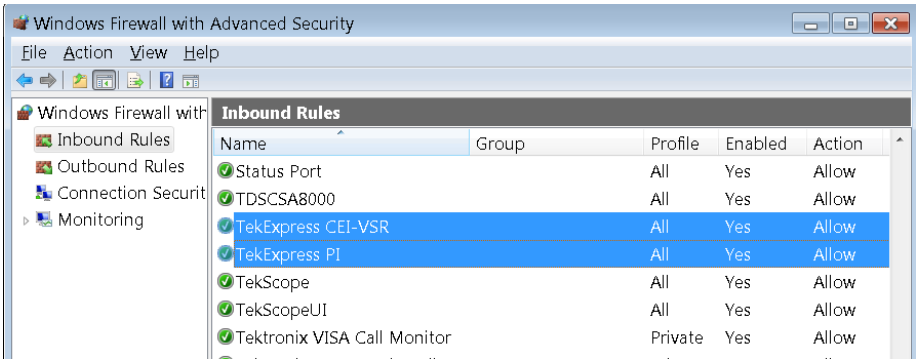


2. Verify that **Program** is selected in the Rule Type panel and click **Next**.
3. Click **Browse** in the Program panel and navigate to and select one of the following TekExpress applications (depending on the one for which you need to create a rule):
4. TekExpress DisplayPort.exe
5. TekExpress.exe

NOTE. *See 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 DisplayPort application could be **TekExpress DisplayPort Application**. 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 DisplayPort. 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 TekExpress DisplayPort PI code to access that instrument. For example:

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

Requirements for developing TekExpress client

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

References required

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

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

Required steps for a client

The client uses the following steps to use TekExpressClient.dll to programmatically control the server:

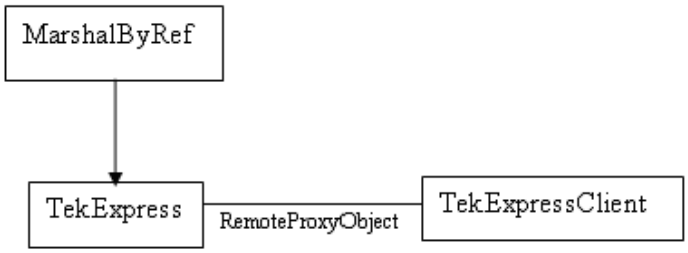
Develop a client UI to access the interfaces exposed through the server. This client loads TekExpressClient.dll to access the interfaces. After TekExpressClient.dll is loaded, the client UI can call the specific functions to run the operations requested by the client. When the client is up and running,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

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);

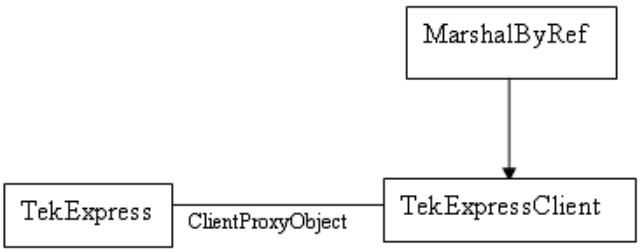
This object lets the remote client access the interfaces exposed at the server side. The client gets the reference to this object when the client gets connected to the server.

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:

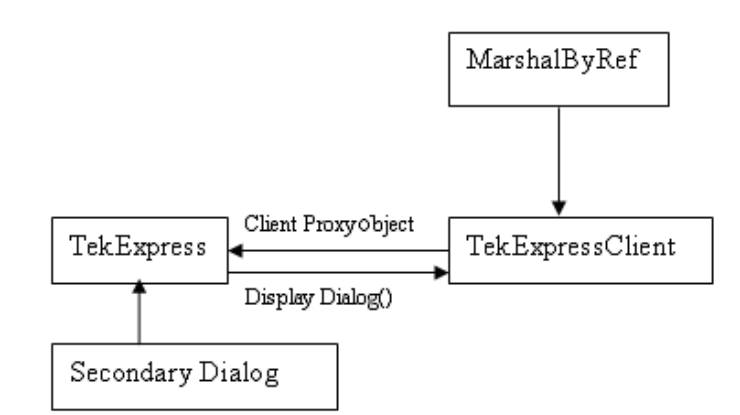
- To get the secondary dialog messages from the server.
- To get the file transfer commands from the server while transferring the report.

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.

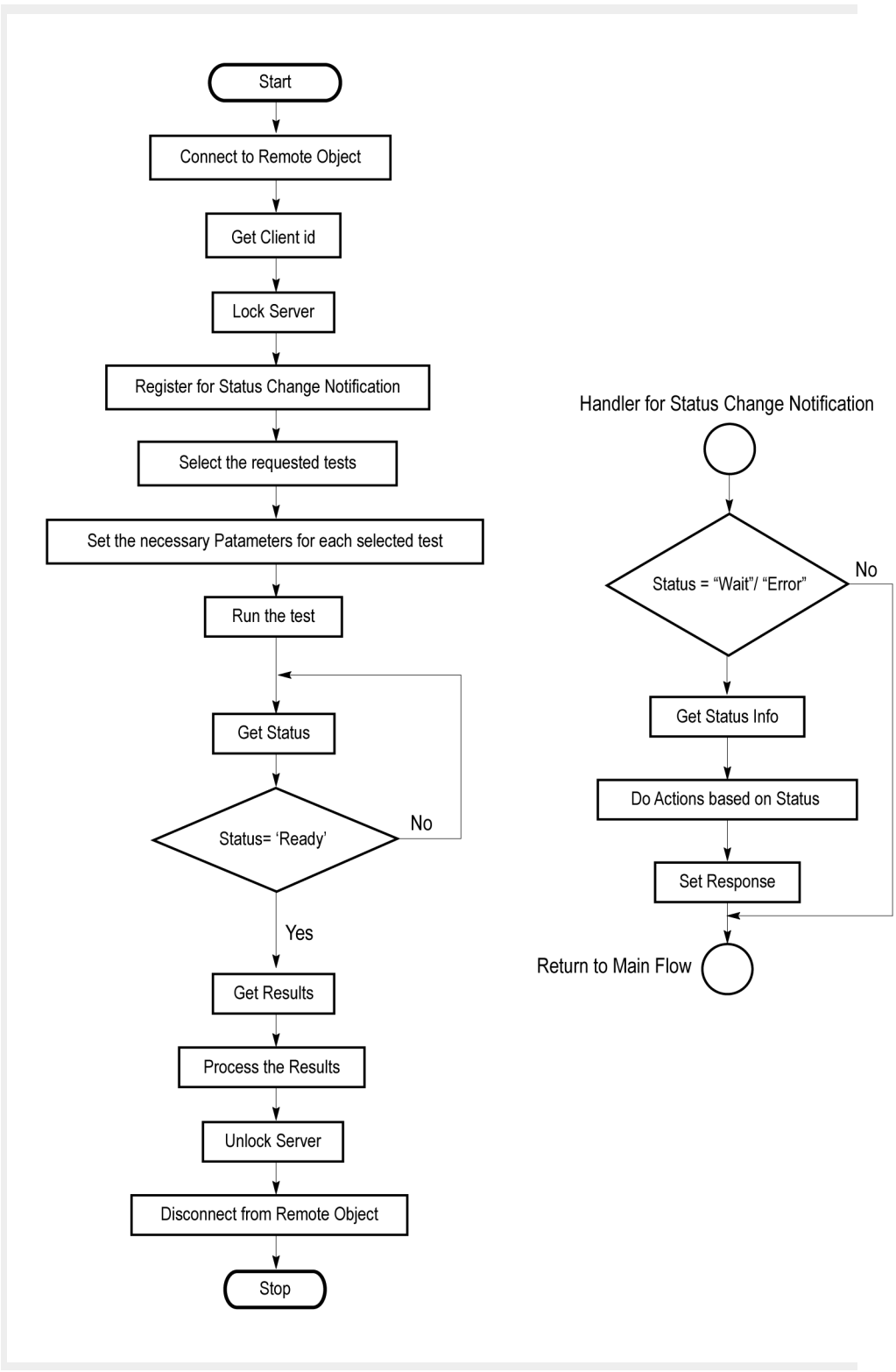
In the secondary dialog library, the assembly that is calling for the dialog box to be displayed is checked and if a remote connection is detected, the messages are directed to the remote end.

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 example

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


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

NOTE. *You can find programmatic interface examples under TekExpress DisplayPort Tx\Examples.*

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

   NOTE. *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.

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

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

   NOTE. *Skip step 8 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 you complete all the tasks.
12. Disconnect from the remote object.

Handler of status change notification

1. Get the status. If the status is Wait or Error, get the information 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 DisplayPort.

Table 14: Remote access code example

Task	Code
Start the application	
Connect through an IP address.	'assigns client IP address to variable clientID; address valid until connection or measurement session ends (Disconnect). See Connect() clientID = " " m_Client.Connect("localhost",out clientID)'True or False
Lock the server	m_Client.LockServer(clientID)
Disable the Popups	m_Client.SetVerboseMode(clientID, false)
Set the DUT ID	m_Client.SetDutId(clientID, "DUT_Name")
Run with set configurations	m_Client.Run(clientID)
Wait for the test to complete.	Do Thread.Sleep(500) m_Client.Application_Status(clientID) Select Case status Case "Wait"
Get the current state information	mClient.GetCurrentStateInfo(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxButtontexts)
Send the response	mClient.SendResponse(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxResponse) End Select Loop Until status = "Ready"
Save results	'Save all results values from folder for current run m_Client.TransferResult(clientID, logDirname)
Unlock the server	m_Client.UnlockServer(clientID)
Disconnect from server	m_Client.Disconnect(clientID)
Exit the application	

DisplayPort programmer interface commands

ApplicationStatus()

ApplicationStatus(clientId)

This method gets the status (ready, running, paused) of the server application.

Parameters

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

clientId variable

clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable:

clientID = “ ”

m_Client.Connect("localhost",out clientId)'True or False

The clientId variable is stored until you call the Disconnect command.

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

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

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

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

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

Return value

String value that gives the status of the server application.

Example

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)

Comments

The application is in the Running, Paused, Wait, or Error state at any given time.

Related command(s)

GetCurrentStateInfo

SendResponse

Status

Connect()

Connect(string HostIPAddress, out string clientID). This command connects the client to the server; address is the IP address of the server to which the client is trying to connect. This is required to establish the connection between the client and the server.

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.*

Parameters.

Parameter	Type	Direction	Description
HostIPAddress	string	IN	Obtains 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.
clientId	string	OUT	Identifier of the client that is performing the remote function. clientId variable

Return value. Value that indicates the connection status (connection was established or an error occurred). The return value can be a boolean value (true), or a string (returning the error message).

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

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

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

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

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

Example. try {

m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL

clientID = “ ”

m_Client.Connect("localhost",out clientID)'True or False

}

Comments. The server has to 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. Each client will get a unique id.

Related command(s). *Disconnect*

Disconnect()

Disconnect(clientId). This command disconnects the client from the server it is connected to.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is performing the remote function. clientId variable

Return value. Integer value that indicates the status of the operation upon completion.

1: Success

–1: Failure

Example. try

{

string returnVal = UnlockServer (clientId);

remoteObject.Disconnect (clientId);

return 1;

}

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

Related command(s). *Connect*

GetCurrentStateInfo()

GetCurrentStateInfo(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxButtontexts). This command 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.

NOTE. *This command is used when the application is running and is in the wait or error state.*

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is performing the remote function. clientId variable
WaitingMsbBxCaption	string	OUT	The wait state or error state message sent to you
WaitingMsbBxMessage	string	OUT	The wait state/error state message sent to you
WaitingMsbBxButtontexts	string array	OUT	An array of strings containing the possible response types that you can send

clientId variable

clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable:

clientID = “ ”

m_Client.Connect("localhost",out clientId)'True or False

The clientId variable is stored until you call the Disconnect command.

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

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

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

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

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

Return value. This command does not return any value.

This function populates the Out parameters that are passed when invoking this function.

Example. m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL

mClient.GetCurrentStateInfo(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxButtontexts)

Comments.

Related command(s). *ApplicationStatus*

SendResponse

GetDutId()

GetDutId(clientId, out dutId). This command returns the DUT id of the current set-up.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is performing the remote function. clientId variable
dutId	string	OUT	The DUT id of the setup.

Return value. String value that indicates the status of the operation upon completion.

Example. returnVal = remoteObject.GetDutId(clientId, out dutId);

if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)

{

return id;

}

else

return CommandFailed(returnVal);

Comments. The dutId is an OUT parameter whose value is set after the server processes the request.

Related command(s). *SetDutId*

GetPassFailStatus()

GetPassFailStatus(clientId, device, suite, test). This command gets the pass or fail status of the measurement after test completion.

NOTE. *Execute this command after completing the measurement.*

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is performing the remote function.
device	string	IN	Valid value is Source.
suite	string	IN	Valid value is Source
test	string	IN	Specifies the name of the test for which to obtain the pass or fail status.

Return value. String value that indicates the status of the operation upon completion.

Example. GetPassFailStatus(clientId, "Device", "Source", "Test 3.1_Eye diagram testing");

GetReportParameter()

GetReportParameter(clientId, device, suite, test, parameterString). This command gets the general report details such as oscilloscope model and TekExpress version.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is performing the remote function.
device	string	IN	Specifies the DUT type (Device).
suite	string	IN	Valid value is Source
test	string	IN	Specifies the name of the test for which to obtain the pass or fail status.
parameterString	string	IN	Specifies to return the measured value for the indicated test. Enter “Scope Model” , “TekExpress Version”, or “Application Version” for this argument

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

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

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

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

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

Return value. The return value is the connected oscilloscope model, TekExpress base software version, or DPTX application version.

Example. GetReportParameter(clientId, “Device”, “Source”, “”, "Application Version")

GetResultsValue()

GetResultsValue(clientId, device, suite, test, parameterString). This command gets the result values of the specified measurement after the run.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is performing the remote function.
device	string	IN	Specifies the DUT type (Device).
suite	string	IN	Valid value is Source

Parameter	Type	Direction	Description
test	string	IN	Specifies the name of the test for which to obtain the test result value. "Test 3.1_Eye Diagram Testing" "Test 3.1.1_Zero Cable Test" "Test 3.1.2_Max Differential Voltage" "Test 3.2_Non Pre-Emphasis Level Verification Testing" "Test 3.3_Pre-Emphasis Level Verification Testing And Maximum Peak-to-Peak Differential Voltage Testing" "Test 3.3.2_Post Cursor2 Verification Testing" "Test 3.4_Inter-Pair Skew Test" "Test 3.5_Intra-Pair Skew Test" "Test 3.10_AC Common Mode Noise Measurements" "Test 3.11_Non ISI Jitter Measurements" "Test 3.12.1_Total Jitter (TJ) Measurements" "Test 3.12.2_Random Jitter (RJ) Measurements" "Test 3.14_Main Link Frequency Compliance" "Test 3.15_Spread Spectrum Modulation Frequency" "Test 3.16_Spread Spectrum Modulation Deviation" "Test 3.17_dF/dt Spread Spectrum Deviation HF Variation" “Test 3.18_Dual-mode TMDS Clock” “Test 3.19_Dual-mode Eye Diagram Testing” “Test 8.1_AUX Manchester Channel EYE Test” “Test 8.2_AUX Manchester Channel

Parameter	Type	Direction	Description
parameterString	string	IN	Specifies to return the measured value for the indicated test. Enter “Value” for this argument or one of the Column names

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

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

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

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

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

Return value. String value that indicates the status of the operation upon completion. Returns the result value in the form of a string.

Example. GetResultsValue(clientId, “Device”, “Source”, "Test 3.1_Eye diagram testing”, "Value");

LockServer()

LockServer(clientId). This command locks the server. The client has to call this command before running any of the remote automations. The server is locked by only one client.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is performing the remote function. clientId variable

Return value. Returns the status of the operation upon completion.

Example. if (locked)

return “Server has already been locked!”;

returnVal = remoteObject.LockServer(clientId);

if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)

{

locked = true;

return “Server Locked...”;

}

Comments. When the client tries to lock a server that is locked by another client, the client gets a message that the server is already locked and it has to 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 automatically unlocked from that client.

Related command(s). *UnlockSession*

RecallSession()

RecallSession(clientId,sessionName). Recalls a saved session. The name of the session is provided by the client.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is performing the remote function. clientId variable
sessionName	string	IN	The name of the session being recalled.

Return value. String that indicates the status of the operation upon completion.

Example. returnVal = remoteObject.RecallSession(clientId,sessionName);

if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)

return “Session Recalled...”;

else

return CommandFailed(returnVal);

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

Related command(s). *SaveSession*

SaveSessionAs

Run()

Run(clientId). Runs the setup. Once the server is set up and configured, it can be run remotely using this function.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable

Return value. String that returns the status of the operation after completion.

Example. returnVal = remoteObject.Run(clientId);

if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)

return “Run started...”;

else

return CommandFailed(returnVal);

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

SaveSession()

SaveSession(clientId,sessionName). Saves the current session. The name of the session is provided by the client.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable
sessionName	string	IN	The name of the session being saved.

Return value. String that indicates the status of the operation upon completion.

Example. returnVal = remoteObject.SaveSession(clientId,sessionName);

if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)

return “Session Saved...”;

else

return CommandFailed(returnVal);

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

Once the session is saved under ‘name,’ you cannot use this command to save the session with a different name. Use SaveSessionAs to save the session to a new name.

Related command(s). *RecallSession*

SaveSessionAs

SaveSessionAs()

SaveSessionAs(clientId,sessionName). Saves the current session in a different name every time this command is called. The name of the session is provided by the client.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable
sessionName	string	IN	The name of the session being saved.

Return value. String that indicates the status of the operation upon completion.

Example. returnVal = remoteObject.SaveSessionAs(clientId,sessionName);

if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)

return “Session Saved...”;

else

return CommandFailed(returnVal);

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

Related command(s). *RecallSession*

SaveSession

SendResponse()

SendResponse(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxButtontexts). After receiving the additional information using the command 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.

NOTE. *This command is used when the application is running and is in the wait or error state.*

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable
WaitingMsbBxCaption	string	OUT	The wait state or error state message sent to you
WaitingMsbBxMessage	string	OUT	The wait state/error state message sent to you
WaitingMsbBxButtontexts	string array	OUT	An array of strings containing the possible response types that you can send

clientId variable

clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable:

clientID = “ ”

m_Client.Connect("localhost",out clientId)'True or False

The clientId variable is stored until you call the Disconnect command.

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

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

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

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

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

Return value. This command does not return any value.

Example. m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL

mClient.SendResponse(clientID, out WaitingMsbBxCaption, out WaitingMsbBxMessage, out WaitingMsbBxButtontexts)

Related command(s). *ApplicationStatus*

GetCurrentStateInfo

SelectDevice()

SelectDevice(clientId, device, true). This command selects the DUT type (Host or Device).

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable
device	string	IN	Specifies the DUT type (**Device**)

Return value. String value that indicates the status of the operation upon completion.

Example. SelectDevice(clientId,"Device")

)

SelectSuite()

SelectSuite(clientId, device, suite, true). This command selects one of the two suites: "Host to Module" or "Module to Host."

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable
device	string	IN	Specifies the DUT type (**Device**)
suite	string	IN	string with device connection type. Valid value is Source.

Return value. String value that indicates the status of the operation upon completion.

Example. SelectSuite(clientId,"Device","Source",true);

SelectTest() **SelectTest(clientId, device, suite, test, true).** This command selects a test.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable
device	string	IN	Specifies the DUT type (Device).
suite	string	IN	string with device connection type. Valid value is Source.

Parameter	Type	Direction	Description
test	string	IN	Name of the DisplayPort test. "Test 3.1_Eye Diagram Testing" "Test 3.1.1_Zero Cable Test" "Test 3.1.2_Max Differential Voltage" "Test 3.2_Non Pre-Emphasis Level Verification Testing" "Test 3.3_Pre-Emphasis Level Verification Testing And Maximum Peak-to-Peak Differential Voltage Testing" "Test 3.3.2_Post Cursor2 Verification Testing" "Test 3.4_Inter-Pair Skew Test" "Test 3.5_Intra-Pair Skew Test" "Test 3.10_AC Common Mode Noise Measurements" "Test 3.11_Non ISI Jitter Measurements" "Test 3.12.1_Total Jitter (TJ) Measurements" "Test 3.12.2_Random Jitter (RJ) Measurements" "Test 3.14_Main Link Frequency Compliance" "Test 3.15_Spread Spectrum Modulation Frequency" "Test 3.16_Spread Spectrum Modulation Deviation" "Test 3.17_dF/dt Spread Spectrum Deviation HF Variation" “Test 3.18_Dual-mode TMDS Clock” “Test 3.19_Dual-mode Eye Diagram Testing” “Test 8.1_AUX Manchester Channel EYE Test” “Test 8.2_AUX Manchester Channel Sensitivity Test”

Return value. String value that indicates the status of the operation upon completion.

Example. SelectTest(clientId, "Device", "Source", "Test 3.1_Eye diagram testing", true);

SetDutId()

SetDutId(clientID,newDutId). This command changes the DUT ID of the setup. The client must provide a valid DUT ID.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable
newDutId	string	IN	The new DUT ID of the setup.

Return value. String that gives the status of the operation after it was performed.

Return value is “DUT Id Changed” on success.

Example. 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)

Comments.

Related command(s). *GetDutId*

SetGeneralParameter()

SetGeneralParameter(clientId, device, suite, "", paramString). This command sets the general parameter and its value based on the "paramString" argument values as listed.

TekExpress programmatic interface SetGeneralParameter()

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server
device	string	IN	Specifies the DUT type (Device)
suite	string	IN	Valid value is Source
paramString	string	IN	Specifies to return the measured value for the indicated test. Enter "Value" for this argument or one of the Column names

Return value. String value that indicates the status of the operation upon completion.

List of available general parameters. Appviewmode$Compliance

Appviewmode$Advanced

RBR$Excluded

RBR$Included

HBR$Excluded

HBR$Included

HBR2$Excluded

HBR2$Included

HBR2$Excluded

PreEmphasis-0dB$Excluded

PreEmphasis-0dB$Included

PreEmphasis-3.5dB$Excluded

PreEmphasis-3.5dB$Included

PreEmphasis-6dB$Excluded

PreEmphasis-6dB$Included

PreEmphasis-9.5dB$Excluded

PreEmphasis-9.5dB$Included

Voltage swing-400mV$Excluded

Voltage swing-400mV$Included

Voltage swing-600mV$Excluded

Voltage swing-600mV$Included

Voltage swing-800mV$Excluded

Voltage swing-800mV$Included

Voltage swing-1200mV$Excluded

Voltage swing-1200mV$Included

PostCursor2-Level0$Excluded

PostCursor2-Level0$Included

PostCursor2-Level1$Excluded

PostCursor2-Level1$Included

PostCursor2-Level2$Excluded

PostCursor2-Level2$Included

PostCursor2-Level3$Excluded

PostCursor2-Level3$Included

SSC$Always On

SSC$Always Off

SSC$Both Supported

D10.2$Included

D10.2$

PRBS7$Included

PRBS7$Excluded

COMP-EYE$Included

COMP-EYE$Excluded

PLTPAT$Included

PLTPAT$Excluded

PCTPAT$Included

PCTPAT$Excluded

Link Widths$1 Lane

Link Widths$2 Lanes

Link Widths$4 Lanes

Example. SetGeneralParameter(clientId, "Device", "Source", "", "RBR $Included")

SetTimeOut()

SetTimeOut(clientId, time). Sets a timeout period specified by the client. After this timeout period expires, the server is unlocked automatically.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server
time	string	IN	The time in seconds that refers to the timeout period

Return Value. String value that indicates the status of the operation upon completion. On success the return value is “TimeOut Period Changed”.

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

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

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

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

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

Example. 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, time)

Related command(s). setTimeOut() **xxx add link**

TekExpress programmatic interface setVerboseMode()

SetVerboseMode()

setVerboseMode(clientId, verboseMode). This command sets the verbose mode to either true or false.

When the value is set to true, any message boxes that appear during the application are 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.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable
verboseMode	boolean	IN	Sets the verbose mode to be turned ON (true) or OFF (false).

Return value. String that gives the status of the operation after it was performed. Returnval as string

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”.

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

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

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

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

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

Example. m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.

Turn on verbose mode:

return=m_Client.SetVerboseMode(clientId, true)

Turn off verbose mode:

returnval=m_Client.SetVerboseMode(clientId, false)

Status()

Status(clientId, out statusMessages). This command gives the status of the run as messages. The status messages are generated once the run is started.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server
statusMessage	string	array OUT	The list of status messages generated during run.

Return value. String that indicates the status of the operation upon completion.

Example. returnVal = remoteObject.QueryStatus(clientId, out statusMessages);

if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)

return “Status updated...”;

else

return CommandFailed(returnVal);

Comments. The status messages are updated periodically after the run begins. The status is an out parameter which is set when the server processes the request.

Related command(s). ApplicationStatus (see page) TekExpress programmatic interface Stop()

Stop()

Stop(clientId). Stops the run operation.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable

Return value. String that indicates the status of the operation upon completion.

Example. returnVal = remoteObject.Stop(clientId);

if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)

return “Stopped...”;

else

return CommandFailed(returnVal);

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

UnlockServer()

UnlockServer(clientId). This command unlocks the server from the client. The client id of the client to be unlocked has to be provided.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable

clientId variable

clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable:

clientID = " "

m_Client.Connect("localhost",out clientId)'True or False

The clientId variable is stored until you call the Disconnect command.

Return value. String that indicates the status of the operation upon completion.

Example. returnVal = remoteObject.UnlockServer(clientId);

```
if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)
{
locked = false;
return "Server UnLocked...";
}
```

Comments. When the client is disconnected, it is automatically unlocked.

Related commands. *LockSession*

Handle error codes

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

Code	Value	Description
-1	FAIL	The operation failed
1	SUCCESS	The operation succeeded
2	NOT FOUND	Server not found
3	LOCKED	The server is locked by another client, so the operation cannot be performed

Code	Value	Description
4	UNLOCK	The server is not locked; lock the server before performing the operation
5	NULL	Nothing

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

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

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

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

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

SetComplianceMode()

SetComplianceMode(clientId, Mode). This command sets the mode to user defined mode.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server
Mode	string	IN	Sets the mode to be tuned 'User-defined' or 'compliance'

Example. Setting mode to user defined:

remoteClientObj.SetComplianceMode(id[1],"User-Defined")

Setting mode to Compliance:

remoteClientObj.SetComplianceMode(id[1],"Compliance")

After setting compliance mode, SetAcquireParameter() and SetAnalyzeParameter() commands can be used to set user defined values.

SetAquireParameter()

SetAcquireParameter(string clientID, string device, string suite, string test, string parameterString). This method sets the parameter values in the Scope Settings tabs in the test configuration section.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server.
device	string	IN	Specifies the name of the device.
suite	string	IN	Specifies the name of the suite.
test	string	IN	Specifies the name of the test to obtain the pass or fail status .
parameterString	string	IN	Selects or deselects a test.

Return value. String that displays the status of the operation after it has been performed.

The return value is " " (an empty string) on success.

Example. Here parameterString is "Acquire_Type$parName$Parvalue"

returnval = remoteClientObj.SetAcquireParameter(id[1], device, suite," Test 3.1_Eye diagram testing"," PRBS7$Sample Rate (GS/s)$25")

List of Acquire parameter. PRBS7$Horizontal scale RBR (us/div)$10

PRBS7$Resolution RBR (ps/pt)$10

PRBS7$Horizontal scale HBR (us/div)$10

PRBS7$Resolution HBR (ps/pt)$10

PRBS7$Sample rate (GS/s)$10

COMP-EYE$Sample rate (GS/s)$10

PRBS7$Acquisition$Included

PRBS7$Horizontal scale (us/div)$10

PRBS7$Resolution (ps/pt)$20

PRBS7$Sample rate (GS/s)$50

PRBS7-InterPair$Acquisition$Included

PRBS7-InterPair$Horizontal scale (us/div)$10

PRBS7-InterPair$Resolution (ps/pt)$10

PRBS7-InterPair$Sample rate (GS/s)$10

PRBS7-Single-ended$Acquisition$Included

PRBS7-Single-ended$Horizontal scale (us/div)$10

PRBS7-Single-ended$Resolution (ps/pt)$20
PRBS7-Single-ended$Sample rate (GS/s)$50
COMP-EYE$Acquisition$Included
COMP-EYE$Horizontal scale HBR2 (us/div)$10
COMP-EYE$Resolution HBR2 (ps/pt)$10
COMP-EYE$Horizontal scale (us/div)$10
COMP-EYE$Resolution (ps/pt)$20
PCTPAT$Acquisition$Included
PCTPAT$Horizontal scale (us/div)$10
PCTPAT$Resolution (ps/pt)$10
PCTPAT$Sample rate (GS/s)$10
PLTPAT$Acquisition$Included
PLTPAT$Horizontal scale (us/div)$10
PLTPAT$Resolution (ps/pt)$20
PLTPAT$Sample rate (GS/s)$50
D10.2$Acquisition$Included
D10.2$Horizontal scale (us/div)$10
D10.2$Resolution (ps/pt)$10
D10.2$Sample rate (GS/s)$10
D10.2-Single-ended$Acquisition$Included
D10.2-Single-ended$Horizontal scale (us/div)$10
D10.2-Single-ended$Resolution (ps/pt)$10
D10.2-Single-ended$Sample rate (GS/s)$10
TMDS Clock$Record Length (M)$20
TMDS Clock$Sample rate (GS/s)$25
TMDS Clock$Edge trigger level (mV)$25
TMDS Clock$Edge trigger slope$Negative
TMDS Clock$Edge trigger slope$Positive
TMDS EyeDiagram$Record Length (M)$20
TMDS EyeDiagram$Sample rate (GS/s)$25
TMDS EyeDiagram$Edge trigger level (mV)$25
TMDS EyeDiagram$Edge trigger slope$Negative
TMDS EyeDiagram$Edge trigger slope$Positive
Aux EyeDiagram$Sample rate (GS/s)$12.5
Aux EyeDiagram$Edge trigger level (mV)$25

Aux EyeDiagram$Edge trigger slope$Negative

Aux EyeDiagram$Edge trigger slope$Positive

Aux Sensitivity$Sample rate (GS/s)$25

Aux Sensitivity$Edge trigger level (mV)$25

Aux Sensitivity$Edge trigger slope$Negative

Aux Sensitivity$Edge trigger slope$Positive

SetAnalyzeParameter()

SetAnalyzeParameter(string clientID, string device, string suite, string test, string parameterString). This method sets the parameter values in the Ref Levels and Clock Settings tabs in the test configuration section.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server.
device	string	IN	Specifies the name of the device.
suite	string	IN	Specifies the name of the suite.
test	string	IN	Specifies the name of the test to obtain the pass or fail status .
parameterString	string	IN	Selects or deselects a test.

Return value. String that displays the status of the operation after it has been performed.

The return value is " " (an empty string) on success.

Example. Here parameterString is "parName$Parvalue"

returnval = remoteClientObj.SetAnalyzeParameter(id[1],device,suite," Test 3.1_Eye diagram testing", "Clock recovery method$PLL-Custom BW")

List of Analyze parameter. Clock recovery method$Constant Clock-Mean

Clock recovery method$PLL-Custom BW

PLL Model Type$1

PLL Model Type$2

Ref Levels Autoset Basetop Method$MINMAX

Ref Levels Autoset Basetop Method$FULLHISTOGRAM

Ref levels$Absolute

Mid level$10

Hysteresis$2
Ref levels$Percentage
Mid level$60
Hysteresis$15
Analysis point-HBR$TP2
Analysis point-HBR$TP3_EQ
BIT Type$TRANsition
BIT Type$ALLBits
BIT Type$NONTRANsition
Low UI for eye height calculation$0.35
High UI for eye height calculation$0.65
RBR Damping (m)$1452
RBR Loop bandwidth (MHz)$7.7
Ref Levels Autoset Basetop Method$EYEHISTOGRAM
Ref Levels Autoset Basetop Method$AUTO
Clock recovery method$Edge
Signal Type$DATA
Signal Type$CLOCK
Signal Type$AUTO
Clock Edge$FALL
Clock Edge$RISE
Clock Edge$BOTH
Clock recovery method$Explicit Clock-PLL
HBR Damping (m)$1452
HBR Loop bandwidth (MHz)$7.7
HBR2 Damping (m)$1452
HBR2 Loop bandwidth (MHz)$7.7

TransferImages()

TransferImages(clientId, filePath). This command transfers all the images (screen shots) to the specified client and folder (directory from the current run.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server. clientId variable
filePath	string	IN	Enclose the path in quotes

Return Value. String value that indicates the status of the operation upon completion.

Transfers all the images in the form of a string.

Example. Images = remoteClientObj.TransferImages(id[1],"C:\\Images")

GetResultsValueForSubMeasurements()

GetResultsValueForSubMeasurements(string clientID, string device, string suite, string test, string parameterString, int rowNr). This method gets the result values for individual submeasurements after the run.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server.
device	string	IN	Specifies the name of the device
suite	string	IN	Specifies the name of the suite.
test	string	IN	Specifies the name of the test to obtain the pass or fail status.
parameterString	string	IN	ParameterString string IN Specifies to return the measured value for the indicated test.
int rowNr	int	IN	The last parameter accepts the row number of table for that particular test.

Return Value. String that gives the status of the operation after it has been performed .

Returns the result value in the form of a string.

Example. resultsValue_sub = remoteClientObj.GetResultsValueForSubMeasurements(id[1],device,suite,measurement,"Value",2)

TransferResult()

TransferResult(clientId, logDirname). This command transfers the report generated after the run to the specified folder (directory). The report contains the summary of the run. The client has to use this command before the saved session command. The client has to provide the location wherre the report is to be saved at the client-end.

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable
logDirname	string	IN	Path to the target folder to which to transfer the report file. Enclose the path in quotes.

Return Value. String that gives the status of the operation after it has been performed

Transfers all the result values in the form of a string.

Example. reports = remoteClientObj.TransferResult(id[1],"C:\\reports ")

Reference

Deskew channels

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

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

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

See Also *Pre-Run checklist*

Map the my TekExpress folder

Follow these steps to map the My TekExpress folder on the instrument:

1. Open Windows Explorer.
2. From the Windows Explorer menu, click **Computer**.
3. In the menu bar, select **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).

To determine the IP address of the instrument where 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 then press **Enter**.
3. At the command prompt, type “ipconfig” and then press **Enter**.

See Also *Before you click start*

Install the software

Measurement parameter descriptions

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

Table 15: Test measurement parameters

Parameter type	Parameter	Description	Applies to ...
Acquire	Pattern	The type of pattern used to acquire data	All tests except test 3.18, 3.19, 8.1, 8.2
	Include this Pattern	When selected, the pattern displayed in the Pattern field is acquired for the test selected. If this option is not selected, the pattern will not be acquired for the test. In the Acquisitions table, the pattern will be marked “Excluded” in the Acquisition column of the Acquisitions table (if the Show Acquire Parameters check box is selected).	Test 3.1, 3.2, 3.3, 3.12.1
	Data Rates	The data rate for the DUT (Gbps) HBR and RBR are displayed for CTS 1.1a. HBR, RBR, and HBR2 are displayed for CTS 1.2 All data rates are selected by default.	Test 3.1, 3.2, 3.3, 3.3.2, 3.4, 3.11, 3.12.1, 3.12.2, 3.14, 3.15, 3.16, 3.17
	SSC	Select the setting that matches how the DUT handles Spread Spectrum Clocking signals.	Test 3.1, 3.2, 3.3, 3.3.2, 3.4, 3.11, 3.12.1, 3.12.2, 3.14, 3.15, 3.16, 3.17
	Pre-Emphasis Levels	Select the signal levels generated by the DUT to make these available in the test.	Test 3.1, 3.2, 3.3, 3.3.2, 3.4, 3.11, 3.12.1, 3.12.2, 3.14, 3.15, 3.16, 3.17
	Voltage Swing Levels	400 mV, 600 mV, 800 mV, 1200 mV. All voltage levels are selected by default.	Test 3.1, 3.2, 3.3, 3.3.2, 3.4, 3.11, 3.12.1, 3.12.2, 3.14, 3.15, 3.16, 3.17
	Post Cursor Levels	Four levels: 0, 1, 2, and 3	Test 3.1, 3.2, 3.3, 3.3.2, 3.4, 3.11, 3.12.1, 3.12.2, 3.14, 3.15, 3.16, 3.17
Scope Settings	Horizontal Scale	Horizontal scale used to capture the signals	All tests except test 3.18, 3.19, 3.10, 8.1, 8.2
	Resolution	Resolution used on the measurement	All tests except test 3.18, 3.19
	Sample Rate	How often the digital oscilloscope takes a snapshot or sample of the signal	All tests except test 3.18, 3.19, 8.1, 8.2
	Edge Trigger Level (mV)		Test 8.2
	Edge Trigger Slope		All tests except test 3.18, 3.19, 3.10, 8.1, 8.2
	Record Length (M)		Test 8.1 and 3.19
Signal Type	Signal Type		Test 3.18
	Clock Edge		
	Clock Recovery Method		
Bit Type	Bit Type		
Mask File	Mask File		Test 3.19

Parameter type	Parameter	Description	Applies to ...
Analyze	Mask File Path	Specifies the mask file to use for the different data rates	Test 3.1
	Analysis Path	This group of controls is displayed only if CTS 1.1a is selected in the Version field. TP2 is selected by default. If CTS 1.2 is selected, these controls will not be visible and the analysis point is internally set to TP2 by default.	Test 3.3
	Bit Type	■ All Bits: Eye analysis includes both transition and nontransition bits ■ Transition: Eye analysis is only on transition bits ■ Nontransition: Eye analysis is only on nontransition bits	Test 3.1 and 3.12.1
	Unit Interval Selection		Test 3.1
Filters	List of equipment for which filters are available	For each piece of equipment listed, specify the path to the filter file to use.	Test 3.1, 3.12.1
Clock Recovery	View clock recovery for	Select the bit rate: RBR, HBR, or HBR2	Test 3.1, 3.3.2, 3.11, 3.12.1, and 3.12.2
	Method	Clock recovery method: ■ Constant Clock-Mean ■ PLL-Custom BW: Phase Locked Loop Custom Bandwidth	
	PLL Model	Selects between Type 1 and Type 2 phase-locked loop	
	Damping	Specifies the damping ratio of the PLL	
	Loop BW	Sets the bandwidth of the clock recovery PLL	

Parameter type	Parameter	Description	Applies to ...
Ref Levels	Reference Level	■ Absolute: Use to manually set the reference levels. ■ Percentage: Use to set the reference levels as a percentage.	Test 3.1
	Ref Levels Autoset Basetop Method	Method used for calculating the Base and Top of the waveform. (Eye Histogram, Auto) ■ Min-Max: Uses the minimum and maximum values in the waveform to determine the base and top amplitude. Useful for waveforms with low noise and that are free from excessive overshoot ■ Full Histogram: Uses histogram to determine the base and top amplitude. Creates a histogram of the amplitudes of the entire waveform ■ Eye Histogram: Uses histogram to determine the base and top amplitude. Creates a histogram of the amplitudes in the center of each bit (unit interval) while ignoring the waveform during bit transitions ■ Auto: Automatically determines the best Base Top method to use	
	Mid Level	A reference voltage level that defines when the waveform state transition occurs at a given threshold	
	Hysteresis	Used to prevent small amounts of noise in a waveform from producing multiple threshold crossings. Use when the rising and falling thresholds for a given reference voltage level are set to the same value	

See Also *Common test parameters and values*

Configure tests

Index

L

M

N

O

P

R

S

T

U

V

W