

TekExpress®
SATA Compliance Automated Solution
Printable Online Help



077-0984-00

TekExpress®
SATA Compliance Automated Solution
Printable Online Help

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

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

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

TekExpress is a registered trademark of Tektronix, Inc.

TekExpress SATA Compliance Automated Solution, 076-0362-00.

Contacting Tektronix

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

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

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

Table of Contents

Getting help and support

Related documentation	1
Conventions used in help	1
Technical support	2

Getting started

Installing the SATA software	
Install the SATA software	3
Activate the license	3
View software version and license information	4
Introduction to the application	
SATA key features	5
Supported oscilloscopes	5
Supported arbitrary waveform generators	5
Minimum system requirements	6
Application directories and files	7
File name extensions	8

Operating basics

Run the application	9
Exit the application	9
Application controls and menus	
Application controls	10
Options menu	
Options menu overview	11
Instrument control settings	12
Email settings	
Email Settings dialog box overview	12
Configure email settings	13
Deskew	14
Application panels	
Application panel overview	16
Setup panel	
Setup panel overview	16
Set DUT parameters	17
Select tests	18
Set acquisition parameters source	21
Set test notification preferences	22

Configure test parameters	
About configuring test parameters	23
Configure test parameters	23
Common test parameters	23
Status panel overview	25
Results panel	
Results panel overview	26
View test results	27
View test-related files	28
Plots panel	
Plots panel overview	28
Reports panel	
Reports panel overview	29
Select report options	30
View a report	31
Report contents	32

Setting up and configuring tests

About setting up tests	35
Equipment connection setup	36
View connected instruments	39
Test setup overview	41

Running tests

About running tests	43
Before you click Start	43
Pre-run checklist	45

Saving and recalling test setups

Test setups overview	47
Save a test setup	47
Recall a saved test setup	48
Create a new test setup based on an existing one	48

TekExpress programmatic interface

About the programmatic interface	49
Requirements for developing TekExpress client	50
Remote proxy object	51
Client proxy object	51
Client programmatic interface overview	53
Program remote access code example	56
SATA application commands	

SATA application commands flow	58
Connect through an IP address	63
Lock the server	64
Disable the popups	65
Set or get the DUT ID	66
Set the configuration parameters for a suite or measurement	67
Query the configuration parameters for a suite or measurement	69
Select a suite	70
Select a test	71
Select a channel	72
Configure the selected measurement	73
Run with set configurations or stop the run operation	74
Handle error codes	75
Get or set the timeout value	76
Wait for the test to complete	77
After the test is complete	79
Save, recall, or query a saved session	83
Unlock the server	84
Disconnect from the server	84

SATA tests

SATA test definitions	85
TSG01 - Differential Output Voltage - Option 1	85
TSG01 - Differential Output Voltage - Option 2	85
TSG02 - Rise/Fall time	85
TSG03 - Differential Skew	86
TSG04 - AC Common Mode Voltage	87
TSG05 - Rise/Fall Imbalance	88
TSG06 - Amplitude Imbalance	89
TSG09 - Gen1 TJ at connector Clock to Data fBAUD-500	90
TSG10 - Gen1 DJ at connector Clock to Data fBAUD-500	90
TSG11 - Gen2 TJ at connector Clock to Data fBAUD-500	90
TSG12 - Gen2 DJ at connector Clock to Data fBAUD-500	91
TSG13 - Gen3 Transmit Jitter	91
TSG14 - Gen3 TX Maximum Differential Voltage Amplitude	92
TSG15 - Gen3 TX Minimum Differential Voltage Amplitude	92
Eye Diagram	92
PHY01 - Unit Interval	93
PHY02 - Long term Frequency Stability	93
PHY03 - Spread Spectrum Modulation Frequency	93
PHY04 - Spread-Spectrum Modulation Deviation	94
dF/dt Measurement	94

OOB01 - Signal Detection Threshold	95
OOB02 - UI during OOB Signaling	97
OOB03 - COMINIT/RESET and COMWAKE Transmit Burst Length	98
OOB04 - COMINIT/RESET Transmit Gap Length	99
OOB05 - COMWAKE Transmit Gap Length	100
OOB06 - COMWAKE Gap Detection Windows	101
OOB07 - COMINIT Gap Detection Windows	102
RSG01 - GEN1 (1.5 Gb/S) Receiver Jitter Test	105
RSG02 - GEN2 (3 Gb/S) Receiver Jitter Test	105
RSG03 - GEN3 (6 Gb/S) Receiver Jitter Test	105
RSG05 - Asynchronous Receiver Stress Test	106
RSG06 - Asynchronous Receiver Stress Test with SSC	106
RMT01 - Gen1 (1.5 Gb/S) Receiver Margin Test	106
RMT02 - Gen2 (3.0 Gb/S) Receiver Margin Test	107
RMT03 - Gen3 (6.0 Gb/S) Receiver Margin Test	107

Index

Related documentation

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

Table 1: Product documentation

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

See also

[Technical Support \(see page 2\)](#)

Conventions used in help

Online Help uses the following conventions:

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

Technical support

Tektronix values your feedback on our products. To help us serve you better, please send us your suggestions, ideas, or comments on your application or oscilloscope. Contact Tektronix through mail, telephone, or the Web site. See [Contacting Tektronix](#) for more information.

When you contact Tektronix Technical Support, please include the following information (be as specific as possible):

General information

- All instrument model numbers
- Hardware options, if any
- Probes used
- Your name, company, mailing address, phone number, FAX number
- Please indicate if you would like to be contacted by Tektronix about your suggestion or comments.

Application specific information

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

Install the SATA software

The TekExpress SATA web installer page provides a link to the software package TekExpress_SATA_Deployment_Package_v1.0.0.xx.exe used to install the application on a supported oscilloscope. For a list of compatible instruments, see [Supported oscilloscopes and probes \(see page 5\)](#).

To download and install the files:

1. Close the TekScope application.
2. Go to the www.tek.com Web site and search for TekExpress SATA to locate the installation file. Download the file TekExpress_SATA_Deployment_Package_v1.0.0.xx.exe.
3. If you downloaded the file to a PC, copy the file to the oscilloscope.
4. Double-click the executable file to extract the installation files.

After extraction, the installer launches and the software automatically installs in the following Windows 7 location:

C:\Program Files\Tektronix\TekExpress\TekExpress SATA

5. To run the application, open the TekScope application and select **Analyze > TekExpress SATA**.

See also

[Minimum system requirements \(see page 6\)](#)

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

Activate the license

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

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

See also

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

View software version and license information

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

To view version information:

1. From the Options menu, select **About TekExpress**. The following screen appears:



To view license information:

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

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

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

See also

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

[Options menu \(see page 11\)](#)

SATA key features

Welcome to the TekExpress® SATA Compliance Automated Solution application (Option SATA-TSG and SATA-RSG). TekExpress SATA is based on TekExpress version 3.0 Framework, the Tektronix Compliance Test Automation Framework, developed to support current and future test automation needs. It is a highly modular architecture that enables deploying automated test solutions for various serial standards in a relatively short time. TekExpress provides a compliance test automation solution for the Serial ATA Gen 1, Gen 2, and Gen 3 standards.

Key SATA compliance testing features include:

- Automated testing reduces the complexity of executing compliance tests and enables you to test devices faster
- Complete SATA Gen1, Gen2, and Gen3 Test Coverage for Hosts, Drives for i, m and u interface types
- PHY Receiver Signal Requirements and Receiver Margin Test (RMT) for SATA Gen1, Gen2, and Gen3

Supported oscilloscopes

The TekExpress SATA application runs on the following Tektronix oscilloscopes:

- DPO/DSA/MSO71254C
- DPO/DSA/MSO71604C
- DPO/DSA/MSO72004C
- DPO/MSO72304DX
- DPO/DSA/MSO72504D/DX
- DPO/DSA/MSO73304D/DX

See also

[Minimum system requirements \(see page 6\)](#)

Supported arbitrary waveform generators

The TekExpress SATA application supports the following Tektronix Arbitrary Waveform Generators:

- AWG7122B/C with Options 01, 06, and 08

Minimum system requirements

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

Table 2: System requirements

Oscilloscope	■ Supported oscilloscopes (see page 5)
Arbitrary Waveform Generator	■ Supported arbitrary waveform generators (see page 5)
Processor	Same as the oscilloscope
Operating System	Windows 7 64-bit
Memory	Same as the oscilloscope
Hard Disk	Same as the oscilloscope
Display	Same as the oscilloscope
Firmware	TekScope v7.1.1 and above (Windows 7)
Software	<ul style="list-style-type: none"> ■ Matlab Runtime engine v8.0 (Windows 7) ■ IronPython 2.7.3 ■ PyVisa 1.0.0.25 ■ On AWG, SerialXpress application Ver. 3.0 and above ■ Microsoft .NET 4.0 Framework ■ Microsoft Internet Explorer 6.0 SP1 or later ■ Adobe Reader 7.0 or equivalent software for viewing portable document format (PDF) files
DPOJET	v6.1.0 and above
Other Devices	<ul style="list-style-type: none"> ■ Matched pair of SMA cables, one-set minimum for single lane ■ Microsoft compatible mouse or compatible pointing device

See also

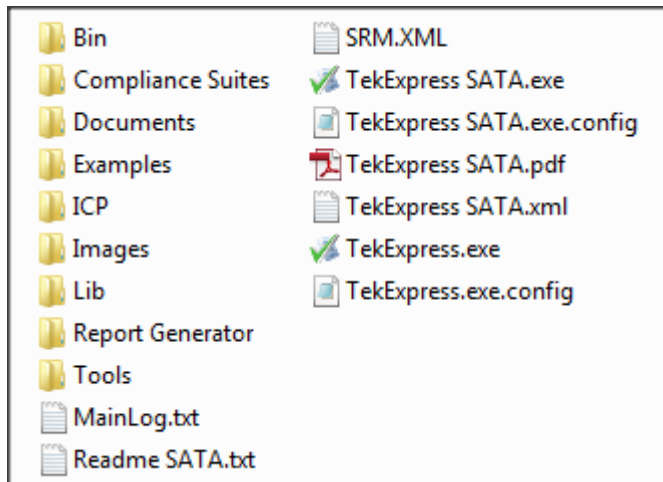
[Supported oscilloscopes \(see page 5\)](#)

[Supported arbitrary waveform generators \(see page 5\)](#)

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

Application directories and files

The application directory and associated files are organized as follows:



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

Table 3: Application directories

Directory names	Contains
InstallDir\TekExpress\TekExpress SATA	Application and associated files
TekExpress SATA\Bin	Miscellaneous SATA application libraries
TekExpress SATA\Compliance Suites	Compliance-specific files
TekExpress SATA\Examples	Various support files
TekExpress SATA\ICP	Instrument and SATA application-specific interface libraries
TekExpress SATA\Lib	Contains utility files specific to the SATA application
TekExpress SATA\Report Generator	Excel Active X interface Library for Report Generation
TekExpress SATA\Tools	Instrument and SATA application-specific files
TekExpress SATA\Documents	Various support files
TekExpress SATA\Images	Application specific images

See also

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

[File name extensions \(see page 8\)](#)

File name extensions

The SATA application uses the following file name extensions:

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

See also

[Select report options \(see page 30\)](#)

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

[Application directories and files \(see page 7\)](#)

Run the application

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

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

When you open the application after installation, the application checks for a file called `Resources.xml` located in the `C:\Users\<username>\My Documents` folder. The `Resources.xml` file gets mapped to the X: drive when the application launches. Session files are then stored inside the `X:\SATA` folder. The `Resources.xml` file contains information about available network-connected instruments. If this file is not found, the application runs an instrument discovery program before launching SATA to locate available instruments.

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


See also

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

Exit the application








Use the following method to exit the application:

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

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

Application controls

Table 4: Application control descriptions

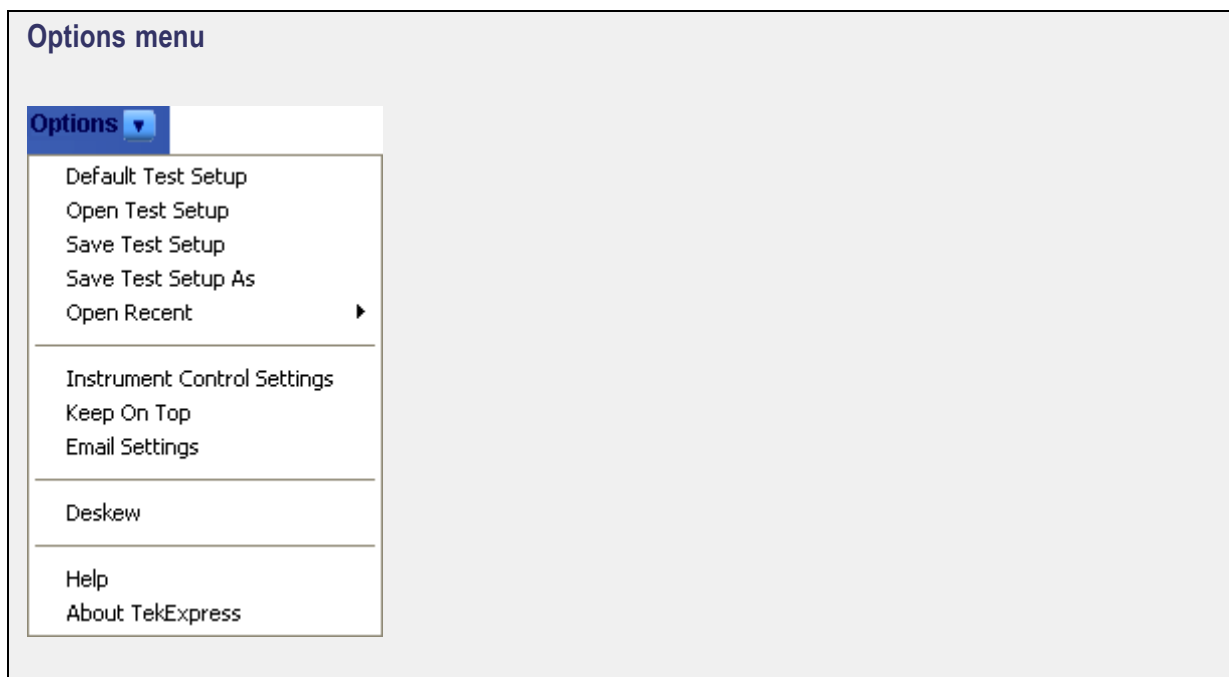
Item	Description
Options menu (see page 11)	Opens the Options menu for access to global controls
Panels (see page 16)	Visual frames with sets of related options
Command buttons	Buttons that initiate an immediate action such as the Start, Stop, Pause, Continue, and Clear command buttons
Start button	<p>Start</p>  <p>Starts continuous measurement acquisition and accumulation. If prior acquired measurements have not been cleared, the new measurements are added to the existing set.</p>
Stop button	<p>Stop</p>  <p>Stops (aborts) the current measurement acquisition.</p>
Pause \ Continue button	<p>Pause Continue</p>  <p>Use the Pause button to temporarily interrupt the current acquisition. When a test is paused, the button name changes to Continue.</p>
Clear button	<p>Clear</p>  <p>Clears all existing measurement results. Adding or deleting a measurement, or changing a configuration parameter of an existing measurement, also clears measurements. This prevents the accumulation of measurement statistics or sets of statistics that are not coherent. This button is available only on the Results panel (see page 26).</p>
Clear Log	<p>Clear Log</p>  <p>This button is available only on the Status panel (see page 25).</p>
Save	<p>Save...</p>  <p>This button is available only on the Status panel (see page 25).</p>
Application window move icon	 <p>Place the cursor over the three-dot pattern in the upper left corner of the application window. When the cursor changes to a hand, drag the window to the desired location.</p>

Options menu overview

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

The [Options menu \(see page 11\)](#) has the following selections:

Menu	Function
Default Test Setup	Opens an untitled test setup with defaults selected
Open Test Setup	Opens a saved test setup
Save Test Setup	Saves the current test setup selections
Save Test Setup As	Creates a new test setup based on an existing one
Open Recent	Displays a menu of recently opened test setups from which to select
Instrument Control Settings (see page 12)	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 SATA utility on top of other open windows on the desktop
Email settings dialog box (see page 12)	Use to configure email options for test run and results notifications
Deskew (see page 14)	Loads oscilloscope channel deskew settings into the application.
Help	Displays TekExpress Help
About TekExpress	<ul style="list-style-type: none"> ■ Displays application details such as software name, version number, and copyright ■ Provides access to software version and license information (see page 4) for your SATA installation ■ Provides a link to the Tektronix Web site

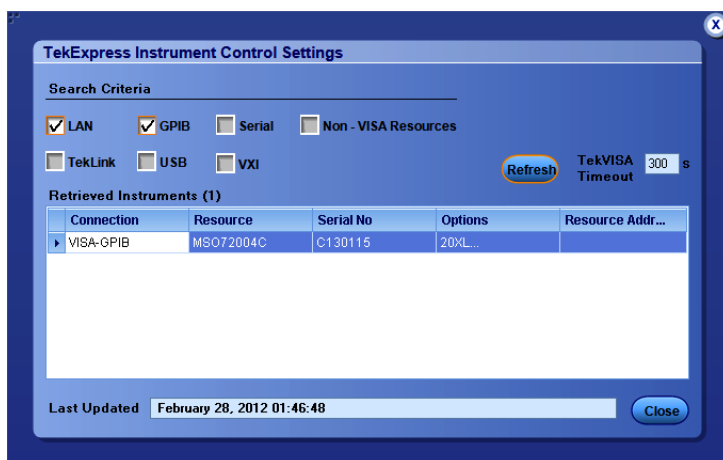


See also

[Application controls \(see page 10\)](#)

Instrument control settings

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



Use the Instrument Control Settings feature to and view instrument connection details. Connected instruments displayed here can be selected for use in the Global Settings tab in the configuration section. See step 1 of [Configure Tests \(see page 23\)](#) for details.

See also

[View connected instruments \(see page 39\)](#)

[Options menu overview \(see page 11\)](#)

Email Settings dialog box overview

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

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

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

Email Settings

Recipient e-mail Address(es)

Note: Separate Email addresses with a comma

Sender's Address

Email Attachments

Reports

Status Log Last 20 Lines Full Log

Server Configuration

SMTP Server SMTP Port

Login

Password

Host Name

Email Configuration

Email Format HTML Plain Text

Number of Attempts to Send

Max Email Size (MB) Timeout (Sec)

Email Test Results When complete or on error

Test Email Apply Close

See also

[Configure email settings \(see page 13\)](#)

[Options menu \(see page 11\)](#)

[Select test notification preferences \(see page 22\)](#)

Configure email settings

To be notified by email when a test completes, fails, or produces an error, configure the email settings.

1. Select **Options > Email Settings** to open the Email settings dialog box.
2. (Required) For Recipient email Address(es), enter one or more email addresses to which to send the test notification. To include multiple addresses, separate the addresses with commas.
3. (Required) For Sender's Address, enter the email address used by the instrument. This address consists of the instrument name, followed by an underscore, followed by the instrument serial number, the @ symbol and then the email server used. For example: DPO72004C_B130099@yourcompany.com.
4. (Required) In the Server Configuration section, type the SMTP Server address of the Mail server configured at the client location, and the SMTP Port number, in the corresponding fields.

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

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

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

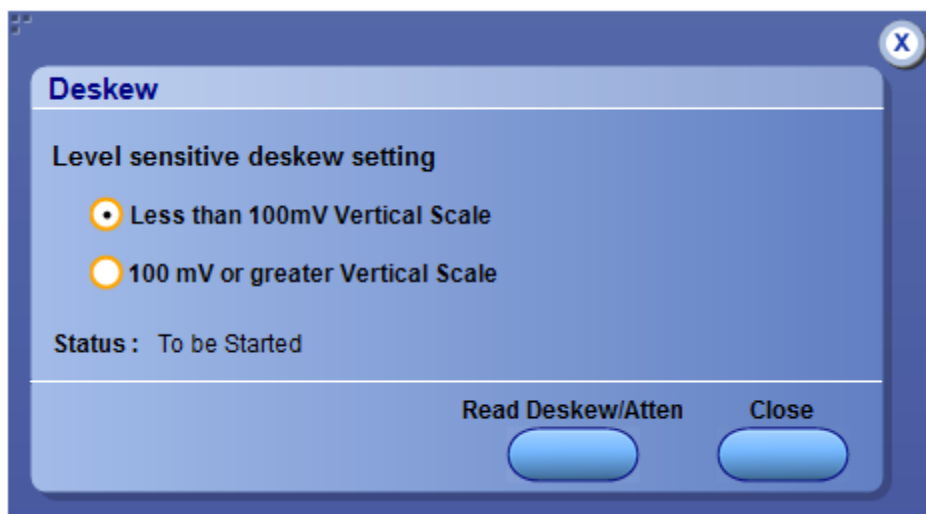
Deskew

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

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

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

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



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

Application panel overview

Panels group related configuration, test and results settings.

The TekExpress SATA panels are:

Table 5: Application panels

Panel name	Description
Setup (see page 16)	Set the DUT, test, acquisition, and report parameters.
Status (see page 25)	View the progress and analysis status of the selected tests, and view test logs.
Results (see page 26)	View a summary of test results and select results viewing preferences.
Plots (see page 28)	View plots related to RSG and RMT tests.
Reports (see page 29)	Browse for reports, save reports as specific file types, specify report naming conventions, select report content to include (such as summary information, detailed information, user comments, setup configuration, application configuration), and select report viewing options.

See also

[Application controls](#) (see page 10)

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

Setup panel overview

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

[Set the DUT parameters](#) (see page 17).

[Select test\(s\)](#) (see page 18).

[Set lane acquisition source](#) (see page 21).

[Configure the selected tests](#) (see page 23).

[Select test notification preferences](#) (see page 22).

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

Set DUT parameters

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


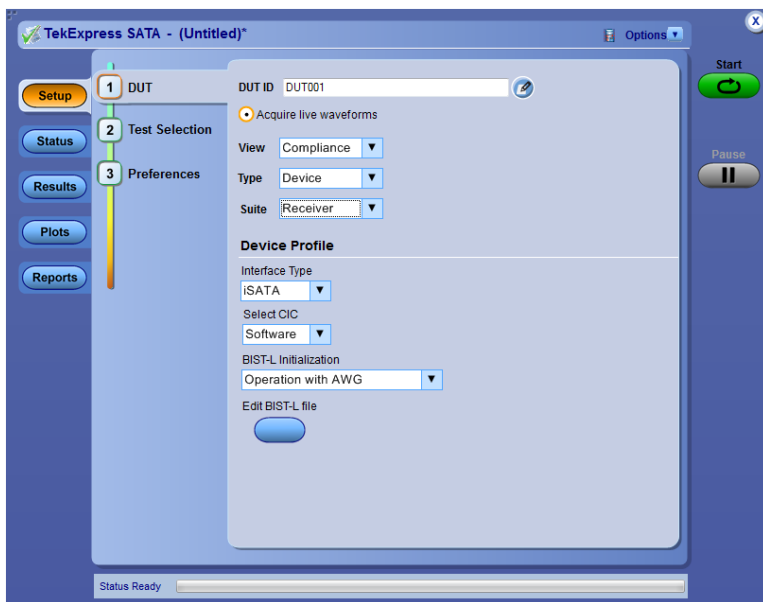
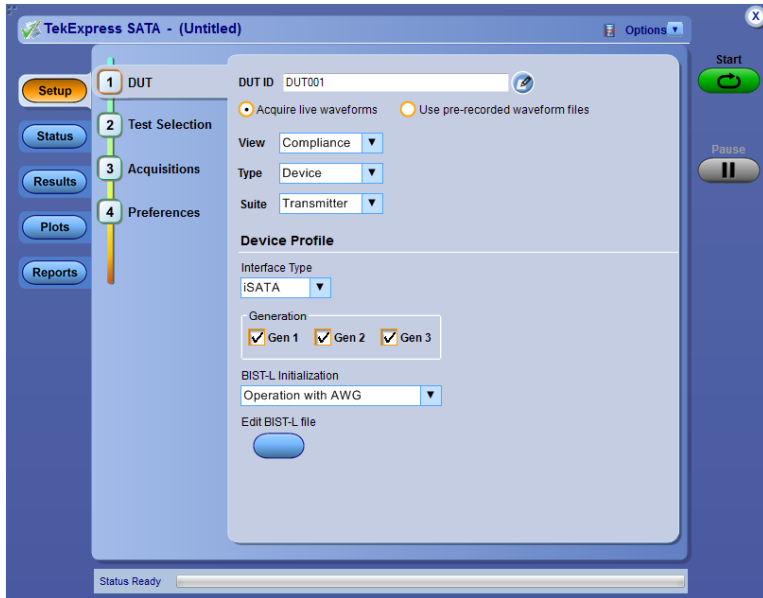
1. Click **Setup > DUT**.
2. (Optional) Enter the ID for the device. The default value is DUT001. The DUT ID parameter is added to reports.
3. (Optional) To add comments to the test report, click the note pad icon () to the right of the DUT ID field. Enter comment text up to 256 characters. To enable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

Table 6: DUT tab settings

Setting	Description
Acquire live waveforms	Acquire active signals from the oscilloscope for testing.
Use pre-recorded waveform files	Run tests on a saved run session file. Select a run session file from the list.
View	Determines where to access the test configuration settings: <ul style="list-style-type: none"> ■ Compliance: View configuration settings by clicking Setup > Test Selection > Configure ■ Advanced: Enables the Setup > Configuration tab in which to view configuration settings.
Type	Select the device type of either Device or Host. The test suite options are automatically updated corresponding to the device selected
Suite	The application allows you to select tests of transmitter or RSG-RMT to execute.
Interface Type	Choose between iSATA, mSATA and uSATA. uSATA is applicable only for device type Host.
Generation	Select the data rates to include in the tests
BIST-L Initialization	Choose between Operation with AWG, User Defined BIST-L or Custom Utility. This choice determines how the DUT will be put into loopback.

See also

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



Select tests

Use the [Test Selection tab](#) to select the tests to run on the connected DUT.

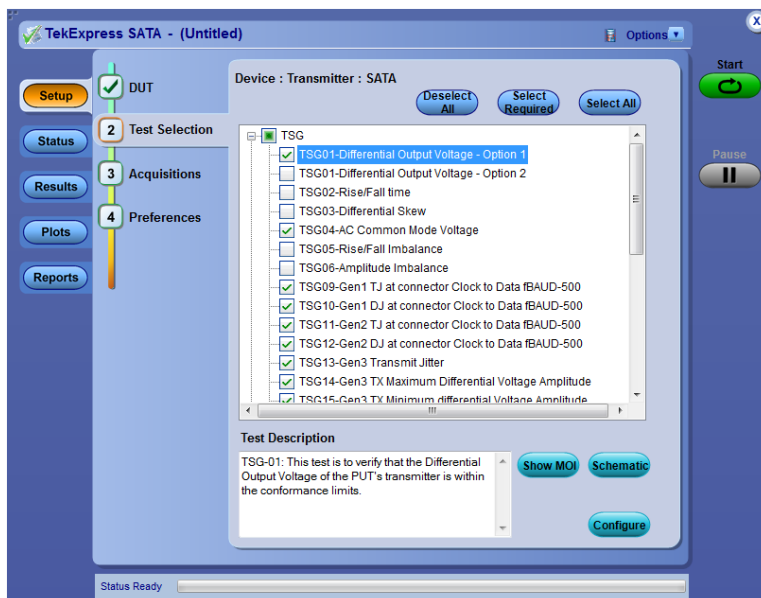
1. Click **Setup > Test Selection**.
2. Select the test(s) to run:
 - Click one or more check boxes adjacent to each test.
 - Click **Deselect All** to deselect all tests. All tests are selected by default.
 - Click **Select Required** to select all tests that are required for compliance.
 - Click **Select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

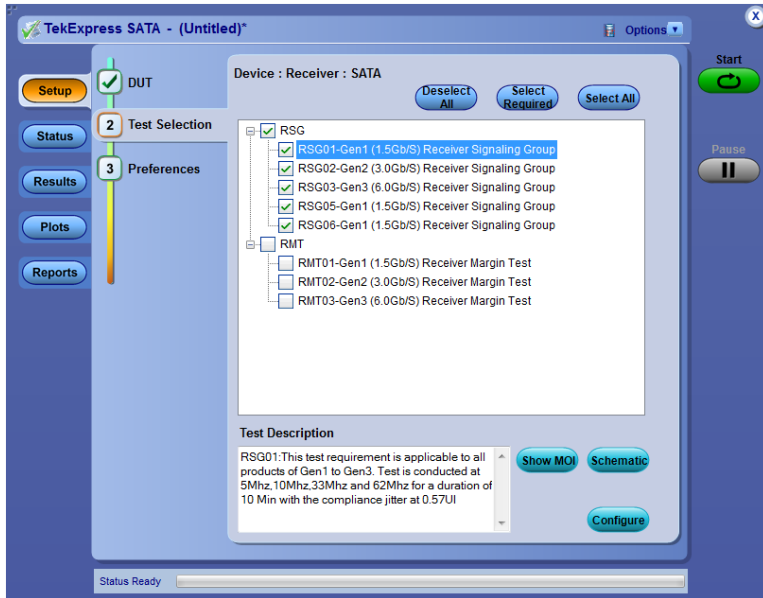
NOTE. *The Configure button is not displayed if the View in the DUT tab is set to Advanced.*

- Click **Schematic** to display a schematic diagram that shows the DUT test setup. Use the diagram to verify the test setup before running the test.
- Click **Show MOI** to display MOI PDF for TSG related tests.

See also

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

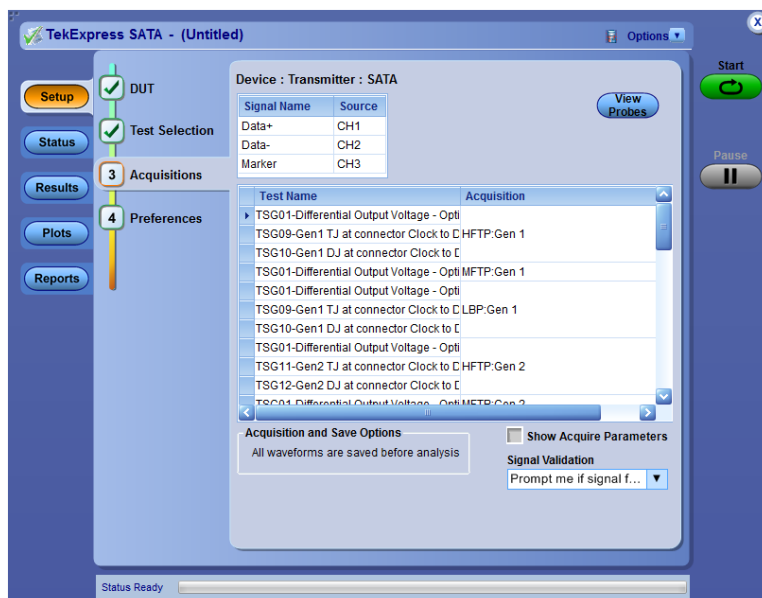




Set acquisition parameters source

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

1. Click **Setup > Acquisitions**.
2. Click the **Data + (+)**, **Data - (-)** and **Marker** fields to select the oscilloscope channels to use for those lanes.
3. Click on the **View Probes** button to view the probe configuration of each oscilloscope channel.
4. Select **Show Acquire Parameters** to show the acquisition parameters in the test list.
5. Select a **Signal Validation** parameter:
 - Select **Prompt me if signal fails** to open a dialog box when the application fails to acquire a valid signal after a specified number of retries (as set in the Configuration tab or fields). Select one of the three options in the dialog box:
 - **Re-Acquire**: The application attempts to re-acquire the signal.
 - **Use Anyway**: Use the acquired signal for all applicable tests.
 - **Skip Test**: Skip (ignore) any test(s) that depend on this acquisition. Skipped tests are listed in the status panel and in the report.
 - Select **Use signal as is - Don't Check** to skip signal validation and use the signal as-is for testing. The test results may not be as expected.



Set test notification preferences

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

1. Click **Setup > Preferences**.
2. Select the **Email Test Results when complete** check box in the Email Settings dialog box. Click **Email Settings** to [configure the email settings \(see page 13\)](#).

See also

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

[Select report options \(see page 30\)](#)

About configuring test parameters

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

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

See also

[Configure test parameters](#)

[About running tests \(see page 43\)](#)

Configure test parameters

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

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

1. Modify [Global settings \(see page 23\)](#) as desired:
 - To select the instruments for testing, click **Global Settings**. In the Instruments Detected section, click in the shaded areas to activate the drop-down lists and select an instrument. If you do not see the desired instrument in the list, [refresh the list \(see page 39\)](#).
2. To modify any individual test measurement settings, click **Measurements**, select the test in the tree view, and change the settings.

See also

[About Setting Up Tests \(see page 35\)](#)

Common test parameters

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

Table 7: Common parameters and values

Parameter type	Parameter
Mode	<p>Determines whether test parameters are in compliance mode or can be edited (User Defined Mode).</p> <ul style="list-style-type: none"> ■ Compliance: Most test parameter values cannot be edited. ■ User Defined: Most test parameter values can be edited.
Global Settings	<p>These settings apply to all tests selected for the current session. You can change only some of these settings.</p> <ul style="list-style-type: none"> ■ Scope Settings: Specifies the instrument channel to use as the source for the trigger. This can be changed. ■ Channel Amplitude: The value of the amplitude to be set for the selected channel.
Measurements	<p>These settings apply to the test selected in the tree view of the configuration section.</p>
Limits Editor	<p>Shows the upper and lower limits for the applicable measurement using different types of comparisons.</p> <p>In Compliance Mode, you are able to view the measurement high and low limits used for the test selected in the tree view of the Measurements tab.</p> <p>When running tests in User Defined Mode, you can edit the limit settings (see page 24) in the Limits Editor. When running tests in User Defined Mode, the cells in the Limits Editor table are active for editing so you can change parameters.</p>

The screenshot shows a window titled "Limits Editor" with the following table:

Test Name	Details	Compare String	Low Limit	Compare String	High Limit
O0B02-UI During OOB Signaling	UI During OOB Sign...	>= Greater Than O...	646.67	<= Less Than Or E...	686.67
O0B03-COMINIT/RESET and COM...	Transmit Burst Leng...	>= Greater Than O...	103.5	<= Less Than Or E...	109.9
O0B04-COMINIT/RESET Transmit...	COMINIT/RESET Tr...	>= Greater Than O...	310.4	<= Less Than Or E...	329.6
O0B05-COMWAKE Transmit Gap ...	Transmit Gap Lengt...	>= Greater Than O...	103.5	<= Less Than Or E...	109.9
PHY01-Unit Interval	Gen1s:Unit Interval ...	>= Greater Than O...	666.43	<= Less Than Or E...	670.23
	Gen2s:Unit Interval ...	>= Greater Than O...	333.22	<= Less Than Or E...	335.12
	Gen3s:Unit Interval ...	>= Greater Than O...	166.61	<= Less Than Or E...	167.56
	Gen1m:Unit Interva...	>= Greater Than O...	666.43	<= Less Than Or E...	670.23
	Gen2m:Unit Interva...	>= Greater Than O...	333.22	<= Less Than Or E...	335.12

When running tests in User Defined Mode, the cells in the Limits Editor table are active for editing so you can change parameters.

See also

[Configure tests \(see page 23\)](#)

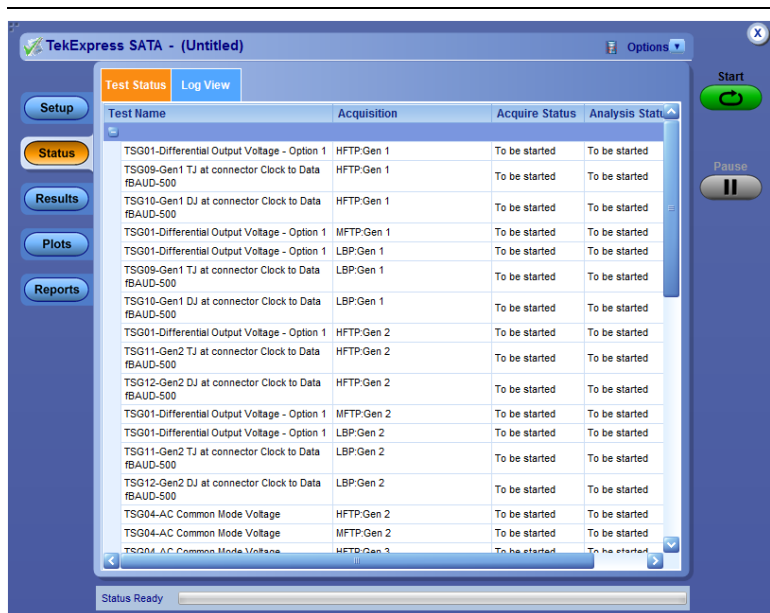
Status panel overview

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

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

Table 8: Test Status tab

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



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

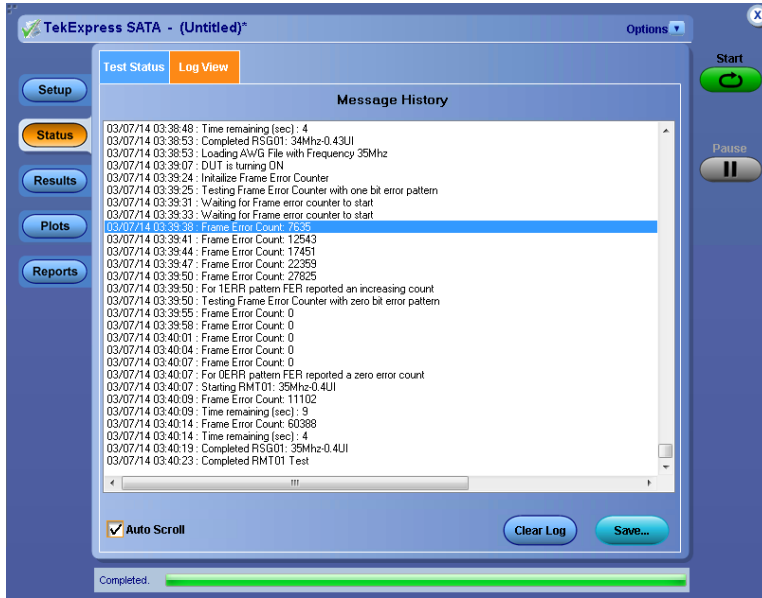


Table 9: Log View fields

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

See also

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

Results panel overview

When a test finishes, the application switches to the [Results panel](#) to display a summary of test results. Set viewing preferences for this panel from the Preferences menu in the upper right corner. Viewing preferences include showing whether a test passed or failed, summary results or detailed results, and enabling wordwrap. For information on using this panel, see [View test results \(see page 27\)](#).

- To expand the width of a column, place the cursor over the vertical line that separates the column from the column to the right. When the cursor changes to a double-ended arrow, hold down the mouse button and drag the column to the desired width.
- To sort the test information by column, click the column head. When sorted in ascending order, a small up arrow is displayed. When sorted in descending order, a small down arrow is displayed.
- To clear all test results displayed, click **Clear**.

See also

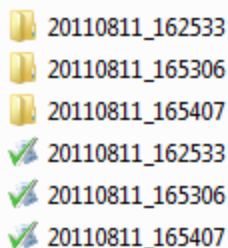
[View a report \(see page 31\)](#)

View test-related files

Files related to tests are stored in the My `TekExpress\SATA` folder. In the `SATA` 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 `SATA` icon and usually has no visible file extension.

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

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



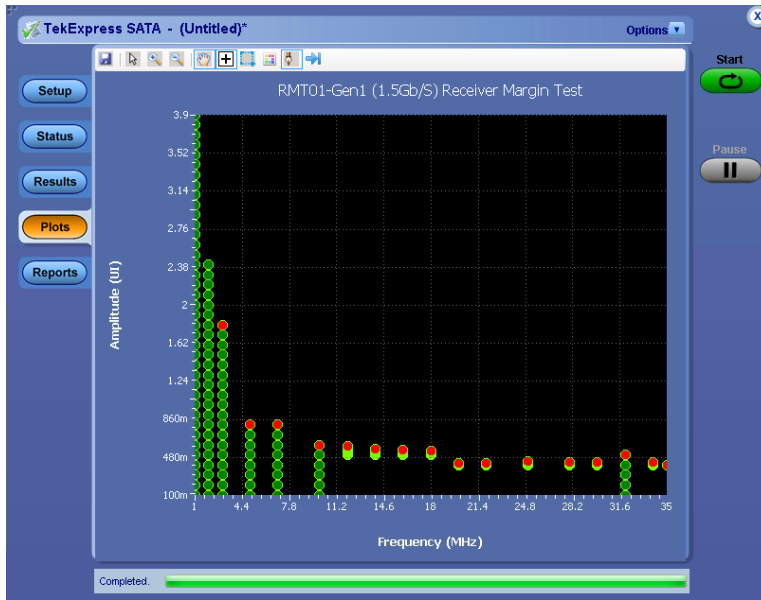
The first time you run a new, unsaved session, the session files are stored in the `Untitled Session` folder located at `.\My TekExpress\SATA`. When you name and save the session, the files are placed in a folder with the name that you specify.

See also

[File name extensions \(see page 8\)](#)

Plots panel overview

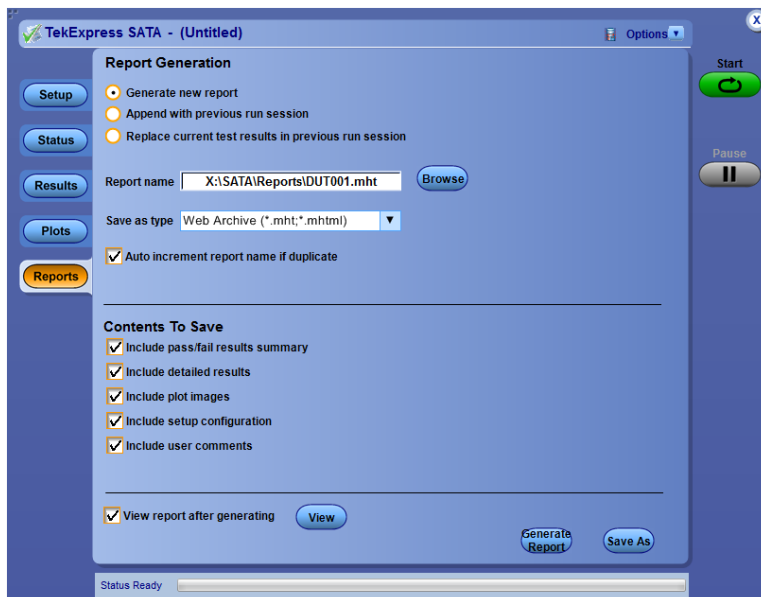
Use the Plots panel to display plots of RSG and RMT tests.



Reports panel overview

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

For information on setting up reports, see [Select report options \(see page 30\)](#). For information on viewing reports, see [View the report \(see page 31\)](#).



Select report options

Use the [Reports panel \(see page 29\)](#) to select which test information to include in the report, and the naming conventions to use for the report. For example, always give the report a unique name or select to have the same name increment each time you run a particular test. Generally, you would select report options before running a test or when creating and saving test setups. Report settings are included in saved test setups.

In the Reports panel, select from the following options:

Table 10: Report options

Setting	Description
Generate New Report	Creates a new report.
Replace current test values with the new results	Replaces the previous test results with the latest test results. Newly added tests results are appended to the end of the report.
Append to Previous Report	Appends the latest test results to the end of the current test results report.
Report Name	<p>Displays the name and location from which to open a report. The default location is at My TekExpress\SATA\Untitled Session. The report file in this folder gets overwritten each time you run a test unless you specify a unique name or select to auto increment the report name.</p> <p>Change the report name or location.</p> <p>Do one of the following:</p> <ul style="list-style-type: none"> ■ In the Report Name field, type over the current folder path and name. ■ Double-click in the Report Name field. Make selections from the popup keyboard and then click the Enter button. <p>Be sure to include the entire folder path, the file name, and the file type. For example: C:\Documents and Settings\your user name\My Documents\My TekExpress\SATA\DUT001_Test_211.mht.</p> <p>NOTE. You cannot change the file location using the Browse button.</p> <p>Open an existing report.</p> <p>Click Browse, locate and select the report file and then click View at the bottom of the panel.</p>
Save As Type	<p>Saves a report in the specified file type. Lists supported file types to choose from.</p> <p>NOTE. If you select a file type different from the default, be sure to change the report file name extension in the Report Name field to match.</p>
Auto increment report name if duplicate	Sets the application to automatically increment the name of the report file if the application finds a file with the same name as the one being generated. For example: DUT001, DUT002, DUT003. This option is enabled by default.
Include Pass/Fail Results Summary	Sets the application to include the color block labeled Test Result (indicating whether the test passed or failed) in the report. For details, see Report Contents in View a Report (see page 31) .

Table 10: Report options (cont.)

Setting	Description
Include Detailed Results	Sets the application to include parameter limits, execution time, and test-specific comments generated during the test.
Include Plot Images	Sets the application to include plotted diagrams.
Include Setup Configuration	Sets the application to include information about hardware and software used in the test in the summary box at the top of the report. Information includes: the oscilloscope model and serial number, probe model and serial number, the oscilloscope firmware version, SPC and factory calibration status, and software versions for applications used in the measurements.
Include Complete Application Configuration	Sets the application to include a table listing general, common, and acquired parameters used in the test. This option is disabled by default.
Include User Comments	Select to include any comments about the test that you or another user added in the DUT tab of the Setup panel. Comments appear in the Comments section under the summary box at the beginning of each report.
View Report After Generating	Automatically opens the report in your Web browser when the test completes. This option is selected by default.

See also

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

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

View a report

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

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

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

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

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

Report contents

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

Setup Configuration Information

Setup configuration information is listed in the summary box at the top of the report. This information includes the oscilloscope model and serial number, probe model and serial number, and software versions. To exclude this information from the report, clear the **Include Setup Configuration** check box in the Reports panel before running the test.

Tektronix TekExpress Automated Test Solution TekExpress SATA			
Setup Information			
DUT ID	DUT001	Scope Model	DPO73304D
Date/Time	2014-01-14 19:02:46	Scope Serial Number	B241041
Device Type	Device	SPC, FactoryCalibration	PASS;PASS
Interface Type	ISATA	Scope F/W Version	7.1.1 devBuild 1
TekExpress Version	SATA:0.0.0.14 Framework:3.0.1.38	DPOJET Version	6.1.0.543
Overall Compliance Mode	Yes	Probe1 Model	1X
Execution Mode	Live	Probe1 Serial Number	N/A
Overall Test Result	Pass	Probe2 Model	1X
Overall Execution Time	0:03:05	Probe2 Serial Number	N/A
DUT COMMENT:	SATA Transmitter		

User Comments

If you selected to include comments in the test report, any comments you added in the DUT tab of the Setup panel appear in the Comments section directly below the summary box.

DUT COMMENT:	SATA Transmitter
--------------	------------------

Test results

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

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

Test Name Summary Table							
OOB01-Signal Detection Threshold				Pass			
OOB02-UI During OOB Signaling				Pass			
OOB03-COMINIT/RESET and COMWAKE Transmit Burst Length				Pass			
OOB04-COMINIT/RESET Transmit Gap Length				Pass			
OOB05-COMWAKE Transmit Gap Length				Pass			
OOB06-COMWAKE Gap Detection Windows				Pass			
OOB07-COMINIT Gap Detection Windows				Pass			
OOB01-Signal Detection Threshold							
Measurement Details	Data Rate	Measured Value	Units	Test Result	Margin	Low Limit	High Limit
OOB Signal Detection Threshold-Response	N.A	Yes	N.A	Pass	N.A	N.A	N.A
OOB Signal Detection Threshold-No Response	N.A	Yes	N.A	Pass	N.A	N.A	N.A
COMMENTS		Back to Summary Table					
OOB02-UI During OOB Signaling							
Measurement Details	Data Rate	Measured Value	Units	Test Result	Margin	Low Limit	High Limit
OOB UI during OOB signaling	N.A	667.2314	ps	Pass	20.5614,19.4386	646.67	686.67
COMMENTS		Back to Summary Table					

See also

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

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

About setting up tests

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

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

See also

[About test setups \(see page 47\)](#)

[Before you click Start \(see page 43\)](#)

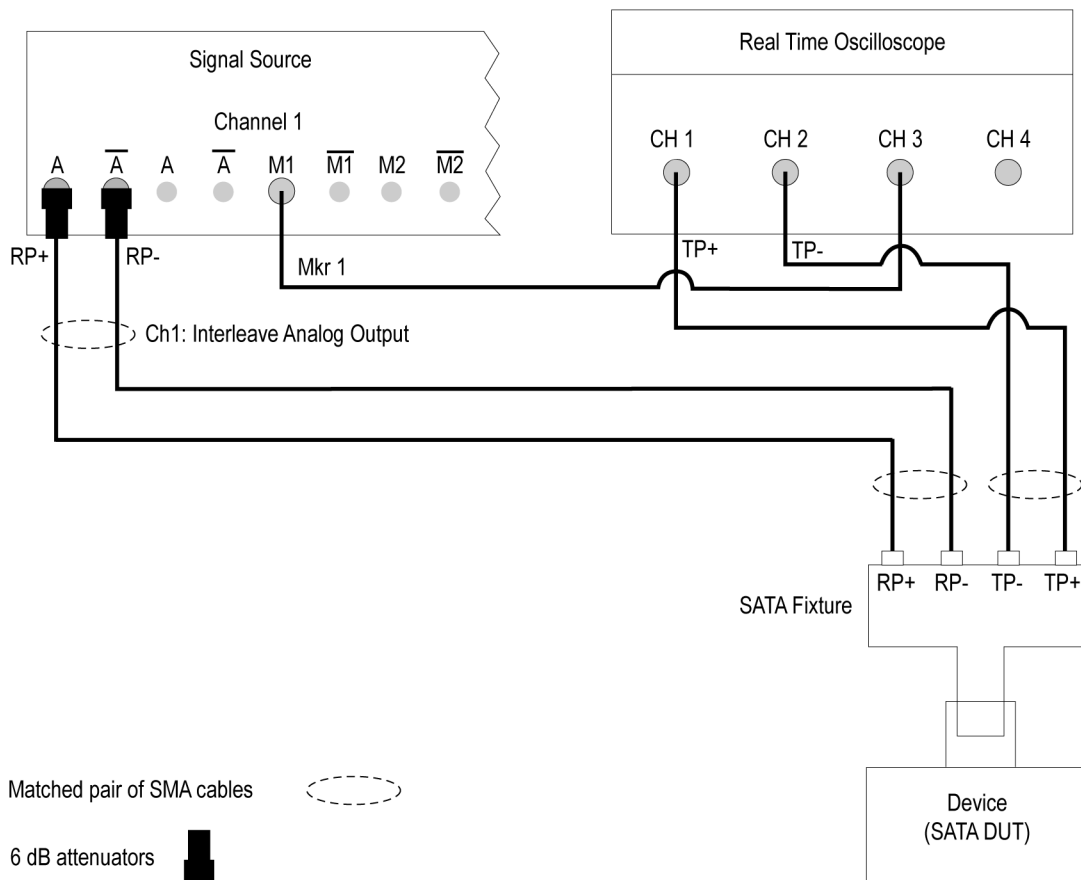
[About running tests \(see page 43\)](#)

Equipment connection setup

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

Device connection diagrams

Setup diagram for Device PHY-TSG-00B

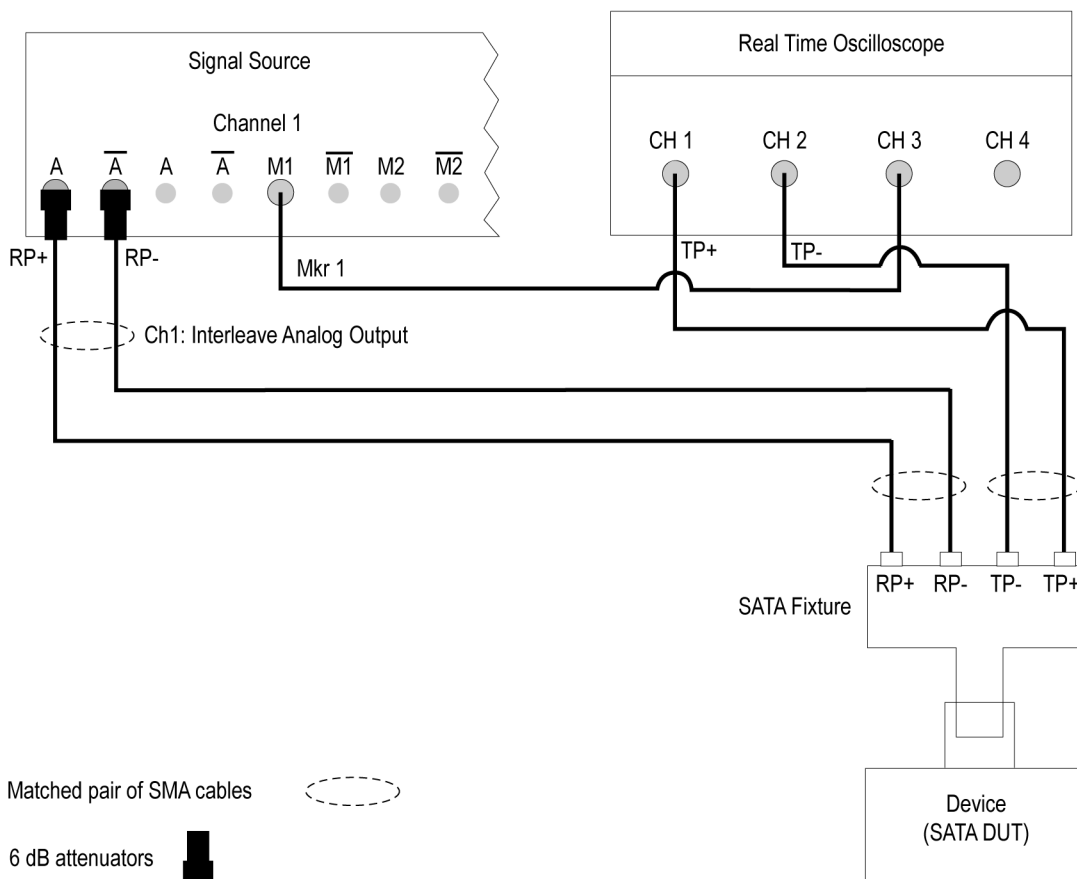


If DUT Power Cycle sequence is automated using DC output of Signal Source, do the following:

- Connect Channel-1 of DC output to pin #14 (and pin #15 to GND) in case of 20 pin Molex connector of AT/ATX power supply.
- Connect Channel-1 of DC output to pin #16 (and pin #17 to GND) in case of 24 pin Molex connector of AT/ATX power supply.

0362-005

Setup diagram for Device RSG-RMT

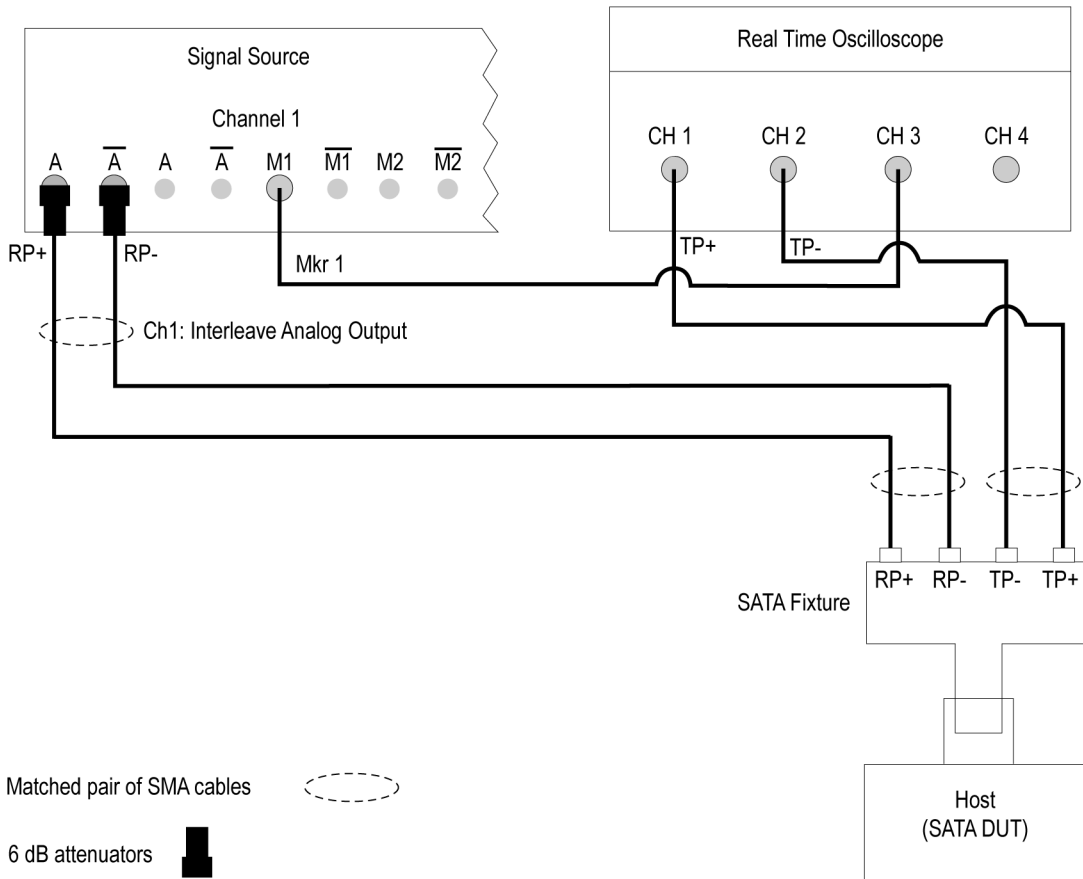



If DUT Power Cycle sequence is automated using DC output of Signal Source, do the following:


- Connect Channel-1 of DC output to pin #14 (and pin #15 to GND) in case of 20 pin Molex connector of AT/ATX power supply.
- Connect Channel-1 of DC output to pin #16 (and pin #17 to GND) in case of 24 pin Molex connector of AT/ATX power supply.

0362-006

Setup diagram for Host PHY-TSG-00B

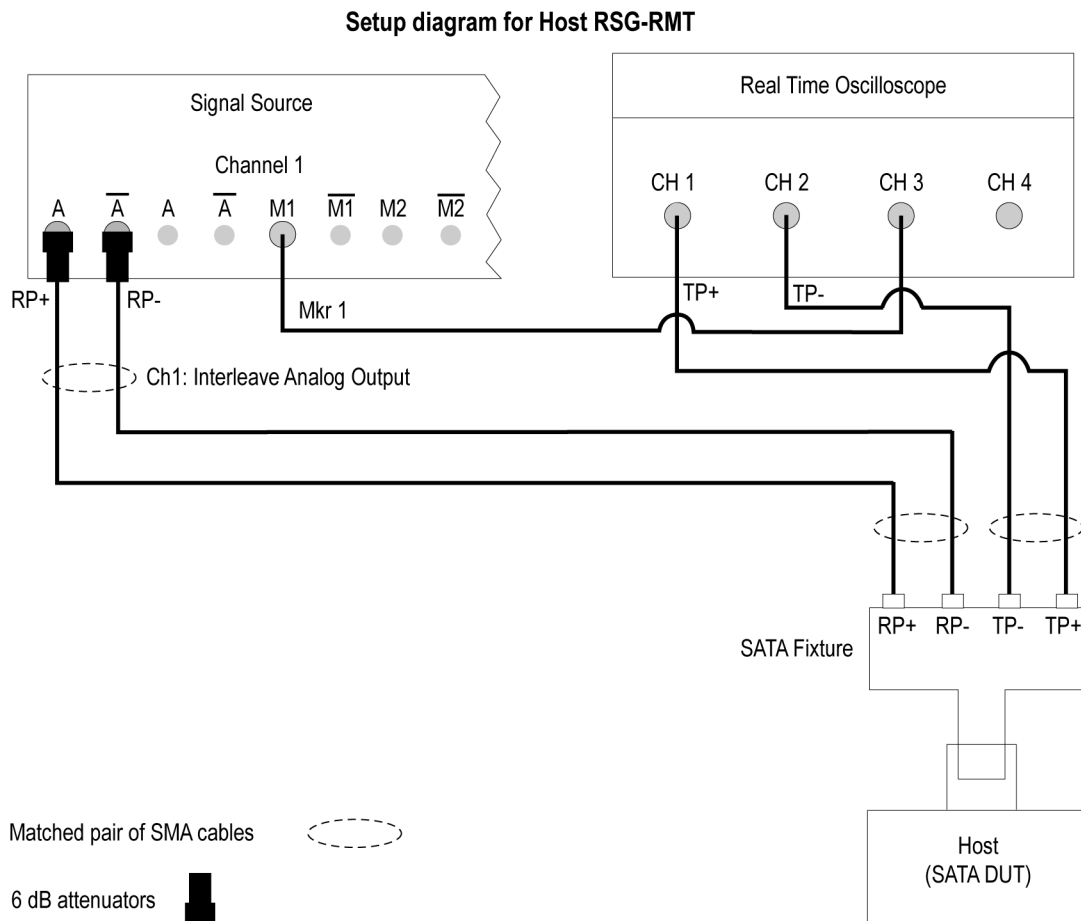


Matched pair of SMA cables 

6 dB attenuators 

If DUT Power Cycle sequence is automated using DC output of Signal Source, do the following:
 -Connect Channel-1 of DC output to pin #14 (and pin #15 to GND) in case of 20 pin Molex connector of AT/ATX power supply.
 -Connect Channel-1 of DC output to pin #16 (and pin #17 to GND) in case of 24 pin Molex connector of AT/ATX power supply.

0362-003



If DUT Power Cycle sequence is automated using DC output of Signal Source, do the following:
 -Connect Channel-1 of DC output to pin #14 (and pin #15 to GND) in case of 20 pin Molex connector of AT/ATX power supply.
 -Connect Channel-1 of DC output to pin #16 (and pin #17 to GND) in case of 24 pin Molex connector of AT/ATX power supply.

0362-004

See also

- [Minimum system requirements \(see page 6\)](#)
- [View connected instruments \(see page 39\)](#)
- [About setting up tests \(see page 35\)](#)

View connected instruments

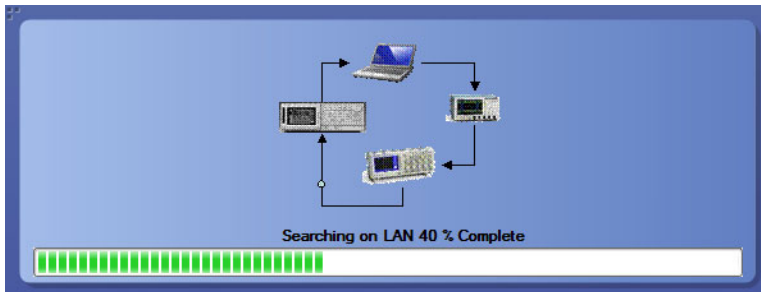
Use the Instrument Control Settings dialog box to view or search for connected instruments required for the tests. The application uses TekVISA to discover the connected instruments.

To refresh the list of connected instruments:

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

Instrument search is based on the VISA layer but different connected cables determine the resource type, such as LAN, GPIB, and USB. For example, if you choose LAN, the search will include all the instruments supported by TekExpress that are communicating over the LAN. If the search does not find any instruments that match a selected resource type, a message appears telling you that no such instruments were found.

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



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



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

See also

[Configure Tests \(see page 23\)](#)

[Equipment Setup \(see page 36\)](#)

Test setup overview

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

1. [Set the DUT parameters \(see page 17\)](#).
2. [Select one or more tests \(see page 18\)](#).
3. [Configure tests \(see page 23\)](#).
4. [Select test notification preferences \(see page 22\)](#).
5. [Select report options \(see page 30\)](#).

See also

[About test Setups \(see page 47\)](#)

[Pre-run checklist \(see page 45\)](#)

[Before you click Start \(see page 43\)](#)

[About running tests \(see page 43\)](#)

About running tests

After selecting and configuring the test, [review the pre-run checklist \(see page 45\)](#) and then click **Start** to run the tests. While tests are running, you cannot access the Setup or Reports panels. To monitor the test progress, switch back and forth between the Status panel and the Results panel.

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

See also

[About Configuring Tests \(see page 23\)](#)

[About Setting Up Tests \(see page 35\)](#)

[Before You Click Start \(see page 43\)](#)

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 SATA, it creates the following folders on the oscilloscope:

- \My Documents\My TekExpress\SATA
- \My Documents\My TekExpress\SATA\Untitled Session

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



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

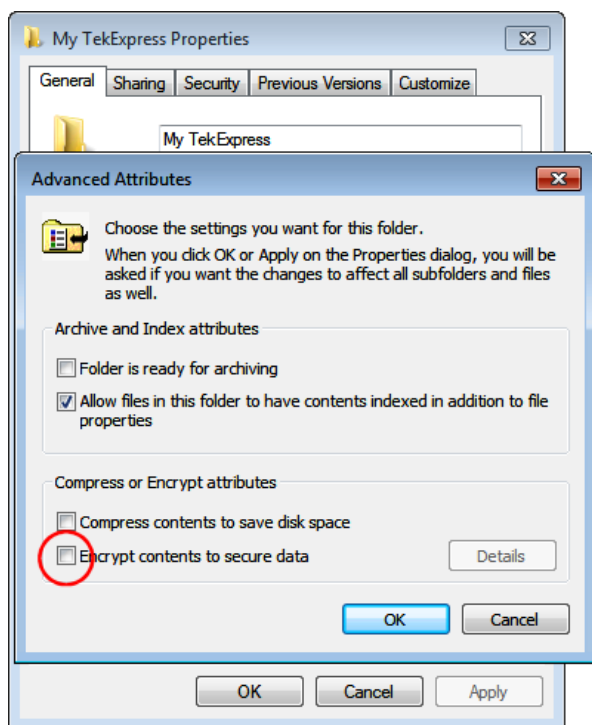
2. Map the shared My TekExpress folder as X: (X drive) on all instruments used in test setup running Microsoft Windows Operating System.

The My TekExpress folder has the shared name format <domain><user ID>My TekExpress. Or, if the instrument is not connected to a domain, then the shared name format is <instrument

name><user ID>My TekExpress. This shared folder is used to save the test session files and is used during any other file transfer operations.

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

3. Make sure that the My TekExpress folder has read and write access and that the contents are not set to be encrypted:
 - a. Right-click the folder and select **Properties**.
 - b. Select the **General** tab and then click **Advanced**.
 - c. In the Advanced Attributes dialog box, make sure that the option **Encrypt contents to secure data** is NOT selected. Example



4. Review the [pre-run checklist \(see page 45\)](#) before you run a test.

See also

[View Test-Related Files \(see page 28\)](#)

[Application Directories and Usage \(see page 7\)](#)

[File Name Extensions \(see page 8\)](#)

Pre-run checklist

Do the following before you click Start to run a test. If this is the first time you are running a test for a setup, refer to the information in [Before You Click Start \(see page 43\)](#).

On the oscilloscope:

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

In the SATA application:

1. Verify that the application is able to find the instrument. If it cannot, [perform a search for connected instruments \(see page 39\)](#).
 - a. Select **Setup > Test Selection**. Select any test and then click **Configure**.
 - b. In the Configuration section, click **Global Settings**.
 - c. In the Instruments Detected section, click the drop-down arrow to the right of the listed instruments and make sure that the instrument is listed.
2. Run the Deskew utility (**Options > Deskew**).

See also

[Equipment Connection Setup \(see page 36\)](#)

Test setups overview

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

See also

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

[Save a test setup \(see page 47\)](#)

[Recall a saved test setup \(see page 48\)](#)

[Create a new test setup based on an existing one \(see page 48\)](#)

Save a test setup

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

The following instructions start from the default test setup:

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

See also

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

[About configuring tests \(see page 23\)](#)

[Test setup overview \(see page 41\)](#)

Recall a saved test setup

To recall a saved test setup:

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

See also

[Test setups overview \(see page 47\)](#)

[Create a new test setup based on an existing one \(see page 48\)](#)

Create a new test setup based on an existing one

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

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

See also

[Test setups overview \(see page 47\)](#)

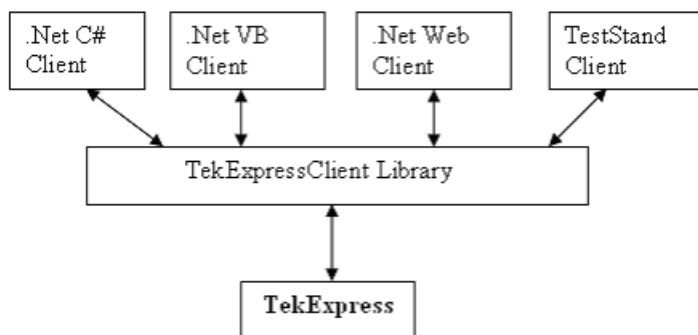
About the programmatic interface

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

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

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

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



See also

[Requirements for Developing TekExpress Client \(see page 50\)](#)

[Remote Proxy Object \(see page 51\)](#)

[Client Proxy Object \(see page 51\)](#)

Requirements for developing TekExpress client

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

References required

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

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

Required steps for a client

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

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

1. To connect to the server, the client provides the IP address of the PC where the server is running.
2. The client locks the server application to avoid conflict with any other Client that may try to control the server simultaneously. “Lock” would also disable all user controls on the server so that server state cannot be changed by manual operation.

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

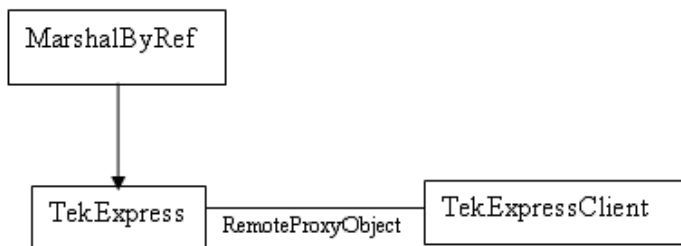
3. When the client has connected to and locked the server, the client can access any of the programmatic controls needed to run the remote automations.
4. After the client operations finish, the client unlocks the server.

See also

[SATA application commands flow \(see page 58\)](#)

Remote proxy object

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



The following is an example:

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

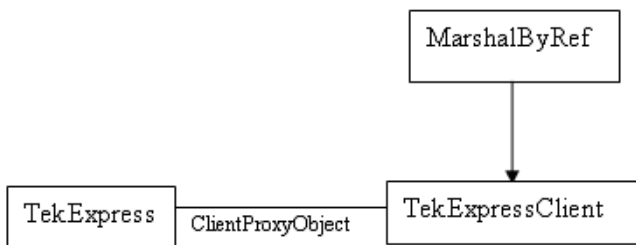
This object lets the remote client access the interfaces exposed at the server side. The client gets the reference to this object when the client gets connected to the server.

For example,

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

Client proxy object

Client exposes a proxy object to receive certain information.



For example,

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

clientInterface = new ClientInterface();

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

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

The client proxy object is used for the following:

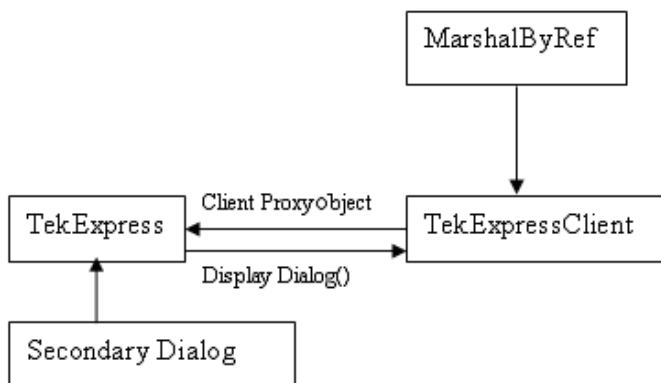
- To get the secondary dialog messages from the server.
- To get the file transfer commands from the server while transferring the report.

Examples

```
clientObject.clientIntf.DisplayDialog(caption, msg, iconType, btnType);
clientObject.clientIntf.TransferBytes(buffer, read, fileLength);
```

For more information, click the following links:

[Secondary Dialog Message Handling](#)



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

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

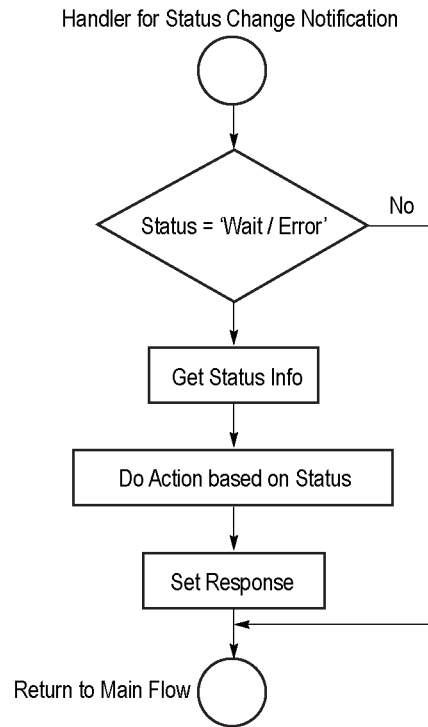
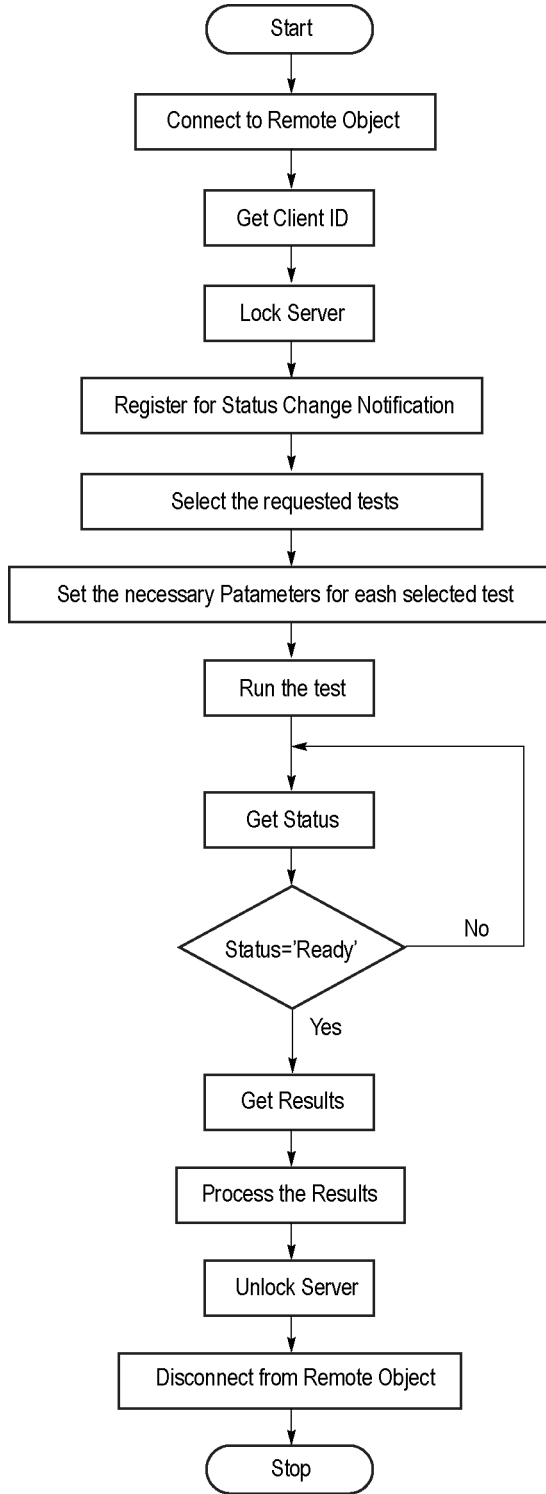
[File Transfer Events](#)

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

Client programmatic interface overview

The following is an overview of the client programmatic interface:

Process flowchart



0643-001

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

NOTE. *The server identifies the client with this ID only and rejects any request if the ID is invalid.*

3. Lock the server for further operations. This disables the application interface.

NOTE. *You can get values from the server or set values from the server to the client only if the application is locked.*

4. Register for receiving notifications on status change events on the server. To register you need to give a handler as a parameter. For details, see [Handler of Status Change Notification \(see page 55\)](#).

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

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

NOTE. *Skip this step if you are registered for the status change notification and the status is Ready.*

9. After completing the tests, get the results.
10. Create a report or display the results and verify or process the results.
11. Unlock the server after completing all the tasks.
12. Disconnect from the remote object.

Handler of Status Change Notification

1. Get the status. If the status is Wait or Error, get the information that contains the title, message description, and the expected responses for the status.
2. Perform the actions based on the status information.
3. Set the response as expected.

See also[SATA application commands flow \(see page 58\)](#)[Program remote access code example \(see page 56\)](#)

Program remote access code example

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

Table 11: Remote access code example

Task	Code
Start the application	
Connect through an IP address.	<code>m_Client.Connect("localhost") 'True or False</code> <code>clientID = m_Client.getClientID</code>
Lock the server	<code>m_Client.LockServer(clientID)</code>
Disable the Popups	<code>m_Client.SetVerboseMode(clientID, false)</code>
Set the DUT ID	<code>m_Client.SetDutId(clientID, "DUT_Name")</code>
Select a test	<code>returnval=mClient.SelectTest(clientID, "Device", "Transmitter", "TSG13-Gen3 Transmit Jitter", True)</code>
Set General Parameter	<code>returnval=mClient.SetGeneralParameter(clientID,"Device","Transmitter","", "BIST-L Initialization\$User Defined BIST-L pattern")</code>
Set Acquire parameters	<code>returnval=mClient.SetAcquireParameter(clientID, "Device", "Transmitter", "TSG13-Gen3 Transmit Jitter", "Record Length (M)\$10")</code>
Set Analyze parameter	<code>returnval=mClient.SetAnalyzeParameter(clientID, "Device", "Transmitter", "TSG13-Gen3 Transmit Jitter", "PLL model type\$TWO")</code>
Run with set configurations	<code>m_Client.Run(clientID)</code>
Wait for the test to complete.	Do <code>Thread.Sleep(500)</code> <code>m_Client.Application_Status(clientID)</code> Select Case status Case "wait" Case "Error"
Get the current state information	<code>mClient.GetCurrentStateInfo(clientID, waitingMsBxBxCaption, waitingMsBxBxMessage, waitingMsBxBxButtonTexts)</code>
Send the response	<code>mClient.SendResponse(clientID, waitingMsBxBxCaption, waitingMsBxBxMessage, waitingMsBxBxResponse)</code> End Select Loop Until status = "Ready"

Table 11: Remote access code example (cont.)

Task	Code
Save results	'Save all results values from folder for current run <code>m_Client.TransferResult(clientID, logDirname)</code>
Unlock the server	<code>m_Client.UnlockServer(clientID)</code>
Disconnect from server	<code>m_Client.Disconnect()</code>
Exit the application	

SATA application commands flow

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

[Connect through an IP address \(see page 63\)](#)

[Lock the server \(see page 64\)](#)

[Disable the popups \(see page 65\)](#)

[Set or get the DUT ID \(see page 66\)](#)

[Set the configuration parameters for a suite or measurement \(see page 67\)](#)

[Query the configuration parameters for a suite or measurement \(see page 69\)](#)

[Select a test \(see page 71\)](#)

[Select a suite \(see page 70\)](#)

[Select a channel \(see page 72\)](#)

[Configure the selected measurement \(see page 73\)](#)

[Run with set configurations or stop the run operation \(see page 74\)](#)

[Handle error codes \(see page 75\)](#)

[Get or set the timeout value \(see page 76\)](#)

[Wait for the test to complete \(see page 77\)](#)

[After the test is complete \(see page 79\)](#)

[Save, recall, or query a saved session \(see page 83\)](#)

[Unlock the server \(see page 84\)](#)

[Disconnect from the server \(see page 84\)](#)

string id			
Name	Type	Direction	Description
id	string	IN	Identifier of the client performing the remote function
Ready: Test configured and ready to start			
Running: Test running			
Paused: Test paused			
Wait: A popup that needs your inputs			
Error: An error is occurred			

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

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

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

string ipAddress			
Name	Type	Direction	Description
ipAddress	string	IN	The ip address of the server to which the client is trying to connect. This is required to establish the connection between the server and the client.

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

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

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

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

string dutId

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

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

string device

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

string suite

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

string test

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

string parameterString

Name	Type	Direction	Description
parameterString	string	IN	Specifies the string value of the parameter that is being set

int rowNr

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

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

out string[] status

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

string name

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

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

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

string name

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

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

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

string name

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

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

bool isSelected

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

string time

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

The time parameter gives the timeout period, which is the time the client is allowed to be locked and idle. After the timeout period if the client is still idle, it gets unlocked.

The time parameter should be a positive integer; otherwise, the client is prompted to provide a valid timeout period.

bool_verbose

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

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

string filePath

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

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

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

out string WaitingMsbBxCaption

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

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

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

string WaitingMsbBxResponse			
Name	Type	Direction	Description
response	string	IN	A string containing the response type that you can select (it must be one of the strings in the string array buttonTexts)

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

Connect through an IP address

Command name	Parameters	Description	Return value	Example
Connect()	string ipAddress (see page 59) out string clientID (see page 59)	This method connects the client to the server. Note (see page 59) The client provides the IP address to connect to the server. The server provides a unique client identification number when connected to it.	Return value is either True or False	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as boolean returnval = m_Client.Connect(ipaddress,m_clientID)</pre>

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

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

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

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

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

Lock the server

Command name	Parameters	Description	Return value	Example
LockSession()	string clientID (see page 63)	This method locks the server. Note (see page 60) The client must call this method before running any of the remote automations. The server can be locked by only one client.	String value that gives the status of the operation after it has been performed The return value is "Session Locked..." on success.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval = m_Client.LockServer(clientID)

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

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

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

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

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

Disable the popups

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

Command name	Parameters	Description	Return value	Example
SetVerboseMode()	string clientID (see page 63) bool _verbose (see page 62)	This method sets the verbose mode to either true or false. When the value is set to true, any message boxes that appear during the application will be routed to the client machine that is controlling TekExpress. When the value is set to false, all the message boxes are shown on the server machine.	String that gives the status of the operation after it has been performed When Verbose mode is set to true, the return value is "Verbose mode turned on. All dialog boxes will be shown to client". When Verbose mode is set to false, the return value is "Verbose mode turned off. All dialog boxes will be shown to server".	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string Verbose mode is turned on return=m_Client.SetVerbose- Mode(clientID, true) Verbose mode is turned off returnval=m_Client.SetVer- boseMode(clientID, false)</pre>

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

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

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

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

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

Set or get the DUT ID

Command name	Parameters	Description	Return value	Example
SetDutId()	string clientID (see page 63) string dutName (see page 59)	This method changes the DUT ID of the setup. The client must provide a valid DUT ID.	String that gives the status of the operation after it has been performed Return value is "DUT Id Changed" on success	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string return=m_Client.SetDutId(clientID,desiredDutId) Note (see page 59)</pre>
GetDutId()	string clientID (see page 63) string dutId (see page 60)	This method gets the DUT ID of the current setup.	String that gives the status of the operation after it has been performed	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string return=m_Client.GetDutId(clientID, out DutId)</pre>

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

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

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

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

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

Set the configuration parameters for a suite or measurement

Command name	Parameters	Description	Return value	Example
SetGeneralParameter	string clientID (see page 63) string device (see page 60) string suite (see page 60) string test (see page 60) string parameterString (see page 60)	This method sets the number of video lanes for the selected measurement. NOTE. Use this command to select a lane, channel, or source type.	String that gives the status of the operation after it has been performed The return value is "" (an empty String) on success.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string Select channel example</pre>
SetAnalyzeParameter()	string clientID (see page 63) string device (see page 60) string suite (see page 60) string test (see page 60) string parameterString (see page 60)	This method sets the configuration parameters in the Analyze panel of the Configuration Panel dialog box for a suite or measurement.	String that gives the status of the operation after it has been performed The return value is "" (an empty String) on success.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string</pre>
SetAcquireParameter()	string clientID (see page 63) string device (see page 60) string suite (see page 60) string test (see page 60) string parameterString (see page 60)	This method sets the configuration parameters in the Acquire panel of the Configuration Panel dialog box for a suite or measurement.	String that gives the status of the operation after it has been performed The return value is "" (an empty String) on success.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnVal = remoteObject.SetAcquireParameter(id, device, suite, test, parameterString) if ((OP_STATUS) returnVal != OP_STATUS.SUCCESS) return CommandFailed(returnVal)</pre>

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

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

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

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

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

Set clock edge example

```
returnval=mClient.SetAnalyzeParameter(clientID, "Device", "Transmitter", "5.3.9 - Random Jitter (RJ)", "Clock Edge$RISE")
```

Query the configuration parameters for a suite or measurement

Command name	Parameters	Description	Return value	Example
GetGeneralParameter()	string clientID (see page 63) string device (see page 60) string suite (see page 60) string test (see page 60) string parameterString (see page 60)	This method gets the general configuration parameters for a suite or measurement.	The return value is the general configuration parameter for a specified suite or measurement that is set.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string</pre> Query status example (see page 79)
GetAnalyzeParameter()	string clientID (see page 63) string device (see page 60) string suite (see page 60) string test (see page 60) string parameterString (see page 60)	This method gets the configuration parameters set in the Analyze panel of the Configuration Panel dialog box for a specified suite or measurement.	The return value is the configuration parameter set in the Analyze panel of the Configuration Panel dialog box for a specified suite or measurement.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string</pre>
GetAcquireParameter()	string clientID (see page 63) string device (see page 60) string suite (see page 60) string test (see page 60) string parameterString (see page 60)	This method gets the configuration parameters set in the Acquire panel for a specified suite or measurement.	The return value is the configuration parameter set in the Acquire panel for a specified suite or measurement.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string</pre>

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

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

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

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

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

Query parameter example

```
returnval=mClient.GetAnalyzeParameter(clientID, " Device", "Transmitter", "TSG13-Gen3 Transmitter Jitter", "Clock Edge")
```

Select a suite

Command name	Parameters	Description	Return value	Example
SelectSuite()	string clientID (see page 63) string device (see page 60) string suite (see page 60) bool isSelected (see page 61)	This method selects or deselects a specified suite. When this parameter is set to true, you can select a suite. When this parameter is set to false, you can deselect a suite.	String that gives the status of the operation after it has been performed The return value is "" (an empty String) on success.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string</pre> <p>Select Suite (Default):</p> <pre>returnval=mClient.SelectSuite(clientID, "Device","Transmitter", true)</pre>

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

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

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

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

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

Select a test

Command name	Parameters	Description	Return value	Example
SelectTest()	string clientId (see page 63) string device (see page 60) string suite (see page 60) string test (see page 60) bool isSelected (see page 61)	This method selects or deselects a specified test. If this Setting parameter is set to true, you can select a measurement. If this Setting parameter is set to false, you can deselect a measurement.	String that displays the status of the operation after it has been performed The return value is "" (an empty String) on success.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string Select test example (see page 71)</pre>

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

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

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

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

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

Select test example

To select measurement test TSG13-Gen3 Transmitter Jitter:

```
mClient.SelectTest(clientId, "Device", "Transmitter", "TSG13-Gen3 Transmitter Jitter", true)
```

Select a channel

Command name	Parameters	Description	Return value	Example
SetGeneralParameter()	string clientID (see page 63) string device (see page 60) string suite (see page 60) string test (see page 60) string parameterString (see page 60)	This method sets the parameters that are not specific to any one test. NOTE. Using this command we can select a lane, channel, or source type.	String that gives the status of the operation after it has been performed The return value is "" (an empty String) on success.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string</pre>
SetAnalyzeParameter()	string clientID (see page 63) string device (see page 60) string suite (see page 60) string test (see page 60) string parameterString (see page 60)	This method sets the configuration parameters in the Analyze panel of the Configuration Panel dialog box for a specified suite or measurement.	The return value is "" (an empty String) on success.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string</pre>
SetAcquireParameter()	string clientID (see page 63) string device (see page 60) string suite (see page 60) string test (see page 60) string parameterString (see page 60)	This method sets the configuration parameters in the Acquire panel of the Configuration Panel dialog box for a specified suite or measurement.	The return value is "" (an empty String) on success.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnVal = remoteObject.SetAcquireParameter(id, device, suite, test, parameterString) if ((OP_STATUS) returnVal != OP_STATUS.SUCCESS) return CommandFailed(returnVal)</pre>

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

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

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

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

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

Configure the selected measurement

Command name	Parameters	Description	Return value	Example
SetAnalyzeParameter()	string clientID (see page 63) string device (see page 60) string suite (see page 60) string test (see page 60) string parameterString (see page 60)	This method sets the Analyze parameters (Configuration parameters) for a specified test.	The return value is "" (an empty String) on success.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string

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

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

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

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

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

Run with set configurations or stop the run operation

Command name	Parameters	Description	Return value	Example
Run()	string clientID (see page 63)	Runs the selected tests Note (see page 61) After the server is set up and configured, run it remotely using this function.	String that gives the status of the operation after it has been performed. The return value is "Run started..." on success.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.Run(clientID)
Stop()	string clientID (see page 63)	Stops the currently running tests Note (see page 62)	String that gives the status of the operation after it has been performed The return value is "Stopped..." on success.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.Stop(clientID)

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

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

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

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

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

Handle error codes

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

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

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

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

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

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

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

Get or set the timeout value

Command name	Parameters	Description	Return value	Example
GetTimeout()	string clientID (see page 63)	Returns the current timeout period set by the client	String that gives the status of the operation after it has been performed The default return value is 1800000.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.GetTimeout()
SetTimeout()	string clientID (see page 63) string time (see page 62)	Sets a timeout period specified by the client. After this timeout period expires, the server is unlocked automatically.	String that gives the status of the operation after it has been performed On success the return value is "TimeOut Period Changed".	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.SetTimeout(clientID, desiredTimeout)

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

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

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

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

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

Wait for the test to complete

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

Command name	Parameters	Description	Return value	Example
<code>ApplicationStatus()</code>	string clientID (see page 63)	This method gets the status of the server application. The states are Ready , Running , Paused , Wait , or Error . (see page 58)	String value that gives the status of the server application	<code>m_Client = new Client()</code> <code>//m_Client is a reference to the Client class in the Client DLL.</code> returnval as string returnval=m_Client.ApplicationStatus(clientID)
<code>QueryStatus()</code>	string clientID (see page 63) out string[] status (see page 61)	An interface for the user to transfer Analyze panel status messages from the server to the client	String that gives the status of the operation after it has been performed On success the return value is "Transferred...".	<code>m_Client = new Client()</code> <code>//m_Client is a reference to the Client class in the Client DLL.</code> returnval as string Query status example (see page 79)

Command name	Parameters	Description	Return value	Example
GetCurrentState-Info() NOTE. This command is used when the application is running and is in the wait or error state.	string clientID (see page 63) out string WaitingMsbBxCaption (see page 62) out string WaitingMsbBxMessage (see page 63) out string[] WaitingMsbBxButtontexts (see page 63)	This method gets the additional information of the states when the application is in Wait or Error state. Except client ID, all the others are Out parameters.	This command does not return any value. This function populates the Out parameters that are passed when invoking this function.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL m_Client.GetCurrentState-Info(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxButtontexts)</pre>
SendResponse() NOTE. This command is used when the application is running and is in the wait or error state.	string clientID (see page 63) out string WaitingMsbBxCaption (see page 62) out string WaitingMsbBxMessage (see page 63) string WaitingMsbBxResponse (see page 63)	After receiving the additional information using the method GetCurrentStateInfo(), the client can decide which response to send and then send the response to the application using this function. The response should be one of the strings that was received earlier as a string array in the GetCurrentStateInfo function. The _caption and _message should match the information received earlier in the GetCurrentStateInfo function.	This command does not return any value.	<pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL m_Client.SendResponse(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxResponse)</pre>

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

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

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

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

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

```

Query status example
returnVal=m_Client.QueryStatus(clientID, out statusMessages)
if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)
    return "Status updated..."
else
    return CommandFailed(returnVal)
    
```

After the test is complete

Command name	Parameters	Description	Return value	Example
GetPassFailStatus()	string clientID (see page 63) string device (see page 60) string suite (see page 60) string test (see page 60)	This method gets the pass or fail status of the measurement after test completion. NOTE. <i>Execute this command after completing the measurement.</i>	String that gives the status of the operation after it has been performed Returns the pass or fail status in the form of a string	<pre> m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.GetPassFailStatus(clientID, device,suite,"TSG13-Gen3 Transmitter Test") </pre>
GetResultsValue()	string clientID (see page 63) string device (see page 60) string suite (see page 60) string test (see page 60) string parameterString (see page 60)	This method gets the result values of the measurement after the run.	String that gives the status of the operation after it has been performed Returns the result value in the form of a string	<pre> m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as srting returnval=m_Client.GetResultsValue(clientID,"Device","Transmitter","TSG13-Gen3 Transmitter Test","True") returnval=m_Client.GetResultsValue(clientID,"Device","Transmitter","TSG13-Gen3 TransmitterTest","False") </pre>

Command name	Parameters	Description	Return value	Example
GetResultsValue- ForSubMeasure- ments()	string clientID (see page 63) string device (see page 60) string suite (see page 60) string test (see page 60) string parame- terString (see page 60) int rowNr (see page 60)	This method gets the result values for individual sub-measurements after the run.	String that gives the status of the operation after it has been performed Returns the result value in the form of a string	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string Get results for a submeasurement example (see page 82)
GetReportParame- ter()	string clientID (see page 63) string device (see page 60) string suite (see page 60) string test (see page 60)	This method gets the general report details such as oscilloscope model, TekExpress version, and SATA version.	The return value is the oscilloscope model, TekExpress version, and SATA version.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string Oscilloscope Model returnval=m_Client.GetReportParameter(clientID,"Scope Model") TekExpress Version returnval=m_Client.GetReportParameter(clientID,"TekExpress Version") SATA Version returnval=m_Client.GetReportParameter(clientID,"Application Version")

Command name	Parameters	Description	Return value	Example
TransferReport()	string clientID (see page 63) string filePath (see page 62)	This method transfers the report generated after the run. The report contains the summary of the run. The client must provide the location where the report is to be saved at the client-end.	String that gives the status of the operation after it has been performed Transfers all the result values in the form of a string	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.TransferReport(clientID,"C:\Report")
TransferImages()	string clientID (see page 63) string filePath (see page 62)	This method transfers all the images (screen shots) from the specified client and folder for the current run (for a suite or measurement). NOTE. Every time you click Start, a folder is created in the X: drive. Transfer the waveforms before clicking Start.	String that gives the status of the operation after it has been performed Transfers all the images in the form of a string	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.TransferImages(clientID, "C:\Waveforms")

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

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

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

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

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

string parameterString

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

Get results for a submeasurement example

This example returns the specified submeasurement results for test TSG13-Gen3 Transmitter Test.

```
returnval=m_Client.GetResultsValueForSubMeasurements(clientID,"Device","Transmitter","TSG13-Gen3 Transmitter Test ","Value",0)
```

```
returnval=m_Client.GetResultsValueForSubMeasurements(clientID,"Device","Transmitter","TSG13-Gen3 Transmitter Test ","Value",1)
```

Save, recall, or query a saved session

Command name	Parameters	Description	Return value	Example
CheckSession-Saved()	string clientID (see page 63) out bool saved (see page 59)	This method checks whether the current session is saved.	Return value is either True or False	<code>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.CheckSessionSaved(m_clientID, out savedStatus)</code>
RecallSession()	string clientID (see page 63) string name (see page 61)	Recalls a saved session. The client provides the session name.	String that gives the status of the operation after it has been performed The return value is "Session Recalled..."	<code>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.RecallSession(clientID, savedSessionName)</code>
SaveSession()	string clientID (see page 63) string name (see page 61)	Saves the current session. The client provides the session name.	String that gives the status of the operation after it has been performed The return value is "Session Saved..."/"Failed..."	<code>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.SaveSession(clientID, desiredSessionName)</code>
SaveSessionAs()	string clientID (see page 63) string name (see page 61)	Saves the current session under a different name every time this method is called. The client provides the session name.	String that gives the status of the operation after it has been performed The return value is "Session Saved..."	<code>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.SaveSessionAs(clientID, desiredSessionName)</code>

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

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

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

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

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

Unlock the server

Command name	Parameters	Description	Return value	Example
UnlockSession()	string clientID (see page 63)	This method unlocks the server from the client. The ID of the client to be unlocked must be provided. Note (see page 62)	String that gives the status of the operation after it has been performed. The return value is "Session Un-Locked..."	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.Unlock-Server(clientID)

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

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

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

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

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

Disconnect from the server

Command name	Parameters	Description	Return value	Example
Disconnect()	string clientID (see page 63)	This method disconnects the client from the server. Note (see page 59)	Integer value that gives the status of the operation after it has been performed 1 for Success -1 for Failure	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.Discon-nect(m_clientID)

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

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

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

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

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

SATA test definitions

The SATA tests listed in this section are available with the application. Some tests may require specific options be installed.

TSG01 - Differential Output Voltage - Option 1

This test is to verify that the Differential Output Voltage of the PUT's transmitter is within the conformance limits. LBP pattern at specified speed is used for the conformance test.

The differential voltage $[(Tx+) - (Tx-)]$ measured at the Transmitter shall comply with the respective electrical specifications. This is measured at mated Serial ATA connector on transmit side including any pre-emphasis.

TSG01 - Differential Output Voltage - Option 2

This test is to verify that the Differential Output Voltage of the PUT's transmitter is within the conformance limits. LFTP pattern at specified speed is used for the conformance test.

The differential voltage $[(Tx+) - (Tx-)]$ measured at the Transmitter shall comply with the respective electrical specifications. This is measured at mated Serial ATA connector on transmit side including any pre-emphasis.

TSG02 - Rise/Fall time

This test is to verify that the Rise/Fall time of the PUT's transmitter is within the conformance limits.

Rise time is the time taken by the DUT output to reach 20% to 80% of the high voltage during transition and vice versa. Transition happens whenever a bit change happens i.e. from 0 to 1 or from 1 to 0.

This is a DPOJET based measurement. DPOJET is set in such a way that rise time/fall time is calculated as the time to rise from 20% to 80% of the waveform and vice versa. A LFTP signal is used to make sure that the reference levels can be calculated without errors. This will lead to a correct estimation of rise time.

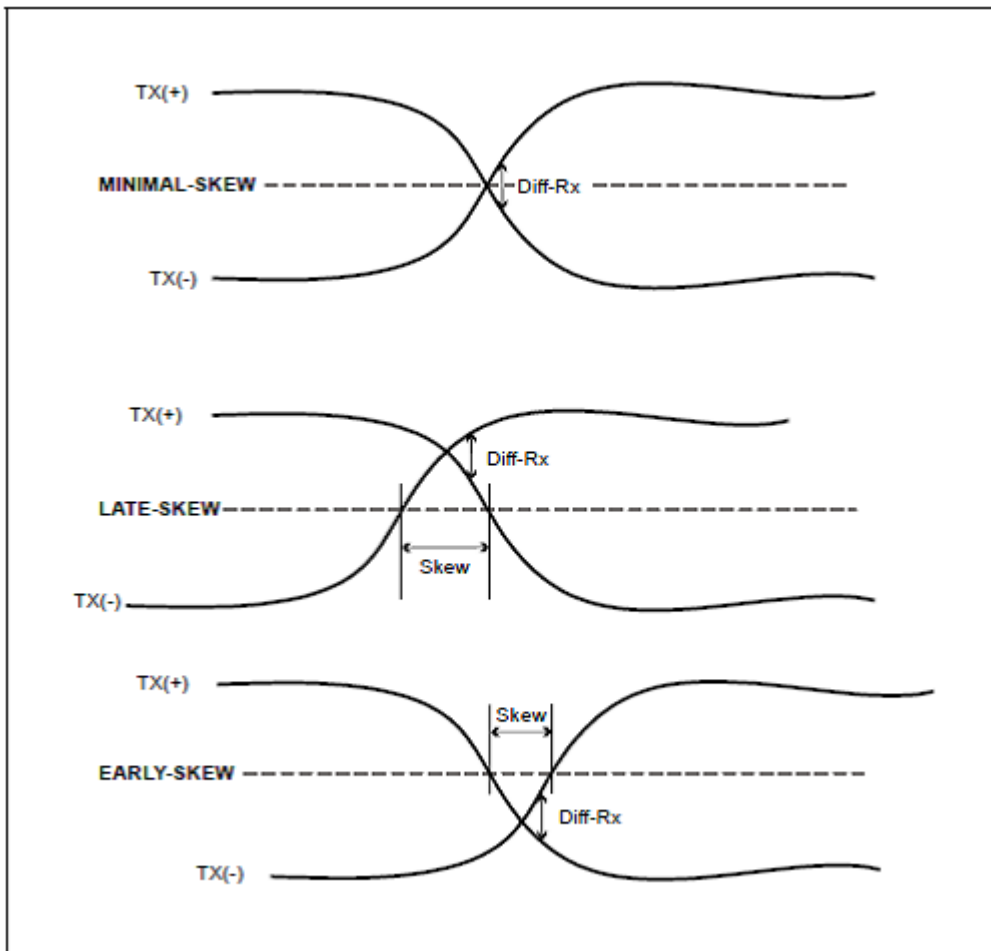
Fall time is the time taken by the DUT output to reach 80% to 20% of the high voltage during transition and vice versa. Transition happens whenever a bit change happens i.e. from 0 to 1 or from 1 to 0.

This is a DPOJET based measurement. DPOJET is set in such a way that rise time/fall time is calculated as the time to rise from 20% to 80% of the waveform and vice versa. A LFTP signal is used to make sure that the reference levels can be calculated without errors. This will lead to a correct estimation of rise time.

TSG03 - Differential Skew

This test is to verify that the Differential Skew of the PUT's transmitter is within the conformance limits. HFTP and MFTP patterns at specified speeds are used for the conformance test.

Tx Differential Skew is the time difference between the single-ended mid-point of the Tx+ signal rising/falling edge, and the single-ended mid-point of the Tx- signal falling/rising edge. It is an important parameter to control as excessive skew may result in increased high frequency jitter and common mode noise levels seen at the far end of the interconnect. Excessive Tx Differential Skew also increases electromagnetic interference (EMI) emissions.

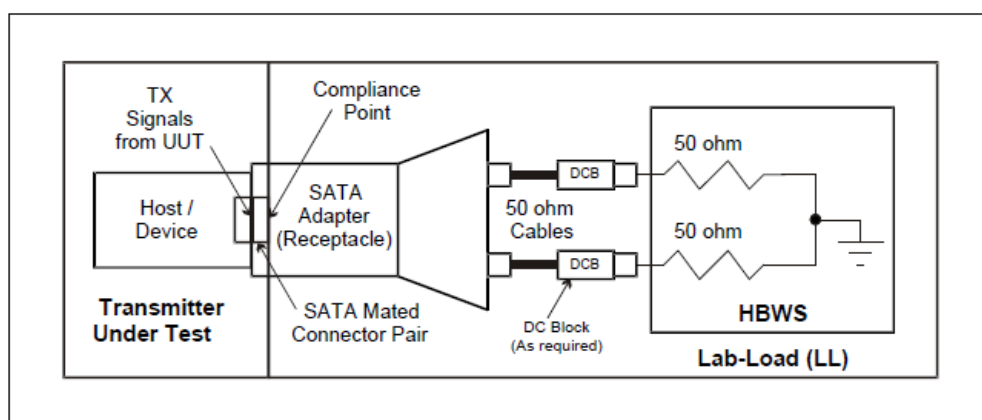


TSG04 - AC Common Mode Voltage

This test is to verify that the AC Common Mode Voltage of the PUT's transmitter is within the conformance limits. HFTP patterns at Gen 3 speed is used for the conformance test.

A common-mode signal is one that appears in phase and with equal amplitudes on both lines of a two-wire cable with respect to the local common or ground in phase and with equal amplitudes.

This parameter is a measure of common mode noise other than the common mode (CM) spikes during transitions due to Tx+/Tx- mismatch and skews that are limited by the rise/fall mismatch and other requirements. Measurement of this parameter is achieved by transmitting through a mated connector into a lab-load (see following Figure).



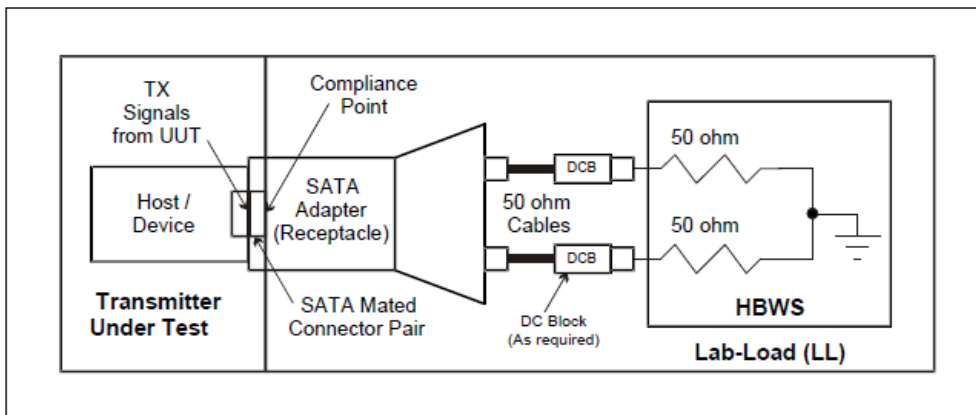
The transmitter shall use an MFTP during a data transfer only, not involving OOB transmissions. The measurement shall be done with a HBWS having a measurement bandwidth limited on the low end at 200 MHz and on the high end at $\text{bitrate}/2$ (fundamental), using first order filtering.

Separate channels shall be used for Tx+ and Tx-, with the common mode being $(\text{Tx}+ + \text{Tx}-) / 2$. The raw common mode is filtered to remove the noise contribution from the edge mismatches. The peak-to-peak voltage of the filter output is the AC Common Mode Voltage, and shall remain below the specified limit.

TSG05 - Rise/Fall Imbalance

This test is to verify that the Rise/Fall Imbalance of the PUT's transmitter is within the conformance limits.

This parameter is a measure of the match in the simultaneous single-ended rise/fall or fall/rise times of the Transmitter. The test setup shown in the following figure shall be used for this measurement.



This parameter shall be measured and met with both the HFTP and MFTP patterns. In order to determine the imbalance, the single ended 20% to 80% rise and fall times of both Tx+ and Tx- shall be determined for a given pattern. Two imbalance values for that pattern are then determined by the two equations:

$$\frac{[TX+, rise - TX-, fall]}{average}$$

Where,

$$average = \frac{TX+, rise - TX-, fall}{2}$$

And,

$$\frac{[TX+, fall - TX-, rise]}{average}$$

Where,

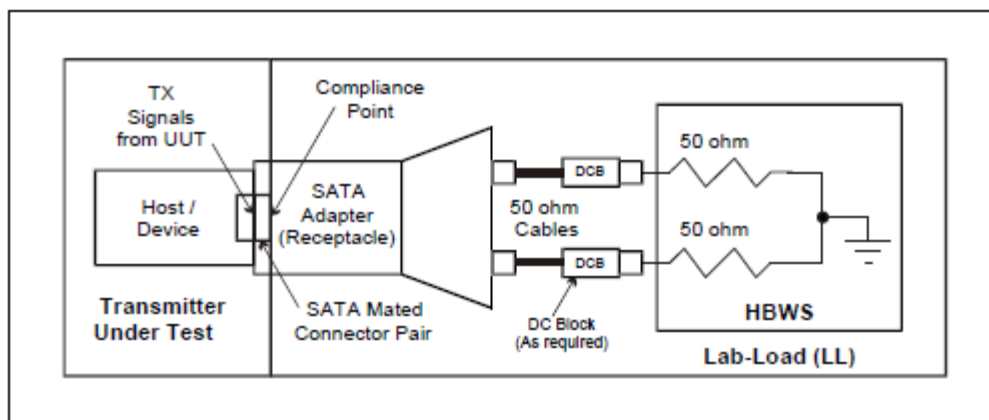
$$average = \frac{TX+, fall + TX-, rise}{2}$$

Both values for each pattern shall follow the limit as per specification.

TSG06 - Amplitude Imbalance

This test is to verify that the Amplitude Imbalance of the PUT's transmitter is within the conformance limits.

This parameter is a measure of the match in the single-ended amplitudes of the Tx+ and Tx- signals. The test setup shown following figure shall be used for this measurement.



This parameter shall be measured and met with both the HFTP and MFTP patterns. Clock-like patterns are used here to enable the use of standard mode-based amplitude measurements for the sole purpose of determining imbalance. Due to characteristics of the MFTP, it is required that the measurement points be taken between 0.45 UI to 0.55 UI of the second bit within the pattern. All amplitude values for this measurement shall be the statistical mode measured at 0.5 UI, nominal, over a minimum of 10 000 UI.

In order to determine the amplitude imbalance, single ended mode high and mode low based amplitudes of both Tx+ and Tx- over 10 cycles to 20 cycles of the clock-like pattern being used shall be determined. The amplitude imbalance value for that pattern is then determined by the equation:

$$\frac{[TX + amplitude - TX - amplitude]}{average}$$

Where,

$$average = \frac{TX + amplitude + TX - amplitude}{2}$$

Both values for each pattern shall follow the limit as per specification.

TSG09 - Gen1 TJ at connector Clock to Data fBAUD-500

This test is to verify that the TJ at Connector of the PUT's transmitter is within the conformance limits.

Total Jitter is a measure of how erroneous the signal is. Lower jitter means less error and higher jitter means more errors. This is a DPOJET based measurement. This is a DPOJET based measurement. Measurements are made on the differential signal.

This test will be performed while the DUT is transmitting HFTP and LBP patterns. The specification requires that the measurement include the effects of the JTF (Jitter Transfer Function), which is a Standard-defined weighting function that is intended to separate the low-frequency timing variations due to SSC from the actual jitter. The TJ measurement requires that the Jitter Measurement Device (JMD) be configured to use the proper JTF characteristics required by the Standard.

TSG10 - Gen1 DJ at connector Clock to Data fBAUD-500

This test is to verify that the DJ at Connector of the PUT's transmitter is within the conformance limits.

Deterministic Jitter is a measure of how the deterministic event like periodic jitter affects the signal. Lower jitter means less errors and higher jitter means more errors.

This test will be performed while the DUT is transmitting HFTP and LBP patterns. The specification requires that the measurement include the effects of the JTF (Jitter Transfer Function), which is a Standard-defined weighting function that is intended to separate the low-frequency timing variations due to SSC from the actual jitter. The RJ measurement requires that the Jitter Measurement Device (JMD) be configured to use the proper JTF characteristics required by the Standard.

TSG11 - Gen2 TJ at connector Clock to Data fBAUD-500

This test is to verify that the TJ at Connector of the PUT's transmitter is within the conformance limits. HFTP and LBP patterns at Gen2 Speed are used for conformance test.

Total Jitter is a measure of how erroneous the signal is. Lower jitter means less error and higher jitter means more errors. This is a DPOJET based measurement. This is a DPOJET based measurement. Measurements are made on the differential signal.

This test will be performed while the DUT is transmitting HFTP and LBP patterns. The specification requires that the measurement include the effects of the JTF (Jitter Transfer Function), which is a Standard-defined weighting function that is intended to separate the low-frequency timing variations due to SSC from the actual jitter. The TJ measurement requires that the Jitter Measurement Device (JMD) be configured to use the proper JTF characteristics required by the Standard.

TSG12 - Gen2 DJ at connector Clock to Data fBAUD-500

This test is to verify that the DJ at Connector of the PUT's transmitter is within the conformance limits. HFTP and LBP patterns at Gen2 Speed are used for conformance test.

Deterministic Jitter is a measure of how the deterministic event like periodic jitter affects the signal. Lower jitter means less errors and higher jitter means more errors.

This test will be performed while the DUT is transmitting HFTP and LBP patterns. The specification requires that the measurement include the effects of the JTF (Jitter Transfer Function), which is a Standard-defined weighting function that is intended to separate the low-frequency timing variations due to SSC from the actual jitter. The RJ measurement requires that the Jitter Measurement Device (JMD) be configured to use the proper JTF characteristics required by the Standard.

TSG13 - Gen3 Transmit Jitter

This test is to verify the maximum amount of jitter that a transmitter may generate and still be SATA compliant. LBP and MFTP patterns at Gen 3 speed are used for the conformance test.

Transmit Jitter is a measure of how erroneous the signal is. Lower jitter means less error and higher jitter means more errors. This is a DPOJET based measurement. This is a DPOJET based measurement. Measurements are made on the differential signal. This test will be performed while the DUT is transmitting LBP pattern. The specification requires that the measurement include the effects of the JTF (Jitter Transfer Function), which is a Standard-defined weighting function that is intended to separate the low-frequency timing variations due to SSC from the actual jitter. The RJ measurement requires that the Jitter Measurement Device (JMD) be configured to use the proper JTF characteristics required by the Standard.

The Transmit Jitter values TJ(10-12) and TJ(10-6) specified in specification refer to the output signal from the DUT at the mated connector into a lab-load. The signals are not specified while attached to a system cable or backplane. All the interconnect characteristics of the transmitter, package, printed circuit board traces, and mated connector pair are included in the measured transmitter jitter. Since the SATA adapter is also included as part of the measurement, good matching and low loss in the adapter are desirable to minimize its contributions to the measured transmitter jitter.

TSG14 - Gen3 TX Maximum Differential Voltage Amplitude

The differential voltage measured at the Transmitter shall comply with the respective electrical specifications of section 7.2. MFTP pattern at Gen 3 speed is used for the conformance test.

The differential voltage $[(Tx+) - (Tx-)]$ measured at the Transmitter shall comply with the respective electrical specifications.

This is measured at mated Serial ATA connector on transmit side including any pre-emphasis. For Gen3i and Gen3u the maximum differential output voltage is likewise measured at the TX compliance point.

TSG15 - Gen3 TX Minimum Differential Voltage Amplitude

This test is to verify that the AC Common Mode Voltage of the PUT's transmitter is within the conformance limits for SATA Gen3i. LBP pattern at Gen 3 speed is used for the conformance test.

The differential voltage $[(Tx+) - (Tx-)]$ measured at the Transmitter shall comply with the respective electrical specifications.

This is measured at mated Serial ATA connector on transmit side including any pre-emphasis. The minimum differential output voltage is measured after the Gen3i CIC. The minimum voltage for Gen3u is not measured after the Gen3i CIC.

Eye Diagram

This test is an informative test.

An eye diagram is a common indicator of the quality of signals in high-speed digital transmissions. An oscilloscope generates an eye diagram by overlaying sweeps of different segments of a long data stream driven by a master clock. The triggering edge may be positive or negative, but the displayed pulse that appears after a delay period may go either way; there is no way of knowing beforehand the value of an arbitrary bit. Therefore, when many such transitions have been overlaid, positive and negative pulses are superimposed on each other. Overlaying many bits produces an eye diagram, so called because the resulting image looks like the opening of an eye.

In an ideal world, eye diagrams would look like rectangular boxes. In reality, communications are imperfect, so the transitions do not line perfectly on top of each other, and an eye-shaped pattern results. On an oscilloscope, the shape of an eye diagram will depend upon various types of triggering signals, such as clock triggers, divided clock triggers, and pattern triggers. Differences in timing and amplitude from bit to bit cause the eye opening to shrink.

PHY01 - Unit Interval

This measurement is for measuring the unit interval or bit rate of a signal. It verifies that the unit interval is within the conformation limits.

PHY02 - Long term Frequency Stability

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

For the purpose of this test, first the time points will be found where the differential signal crosses zero volts (such as, zero crossings). A diff (difference) operation will be performed on this array of time values to produce an array of Unit Interval (UI) widths. The high frequency changes in UI are errors and will be removed by filtering the data using a low-pass filter, which will reveal the underlying lower-frequency stability of the link rate.

The inverse of the UI values will be taken to produce an array of instantaneous frequency values. The low pass filter will then be applied to the inverse of the UI values. The resulting waveform that is produced at the output of the test filter will then be converted to ppm. This result will represent the instantaneous bitrate of the DUT. All of the values must be between specified limit as per specification.

PHY03 - Spread Spectrum Modulation Frequency

This measurement is for finding the Spread spectrum modulation frequency. If the waveform has SSC this measurement will return SSC frequency.

In this test, the SSC modulation frequency of the DUT's transmitted output signaling will be measured while the DUT is transmitting SSC. A sample of the DUT's data signaling will be captured using a real-time DSO, and will be post processed to recover the transmitter's SSC modulation profile. The frequency of the modulation will be observed by measuring the average period over a minimum of 10 SSC cycles, and the inverse of this result will be computed to produce the modulation frequency result.

PHY04 - Spread-Spectrum Modulation Deviation

This measurement verifies that the max SSC Spread is less than 0.23% or less than 2300 ppm. The deviation of the modulation will be determined by measuring maximum and minimum profile peak values per period, over at least 10 complete SSC cycles. From these values, the average maximum and average minimum peak values will be computed, and the results compared against the requirements listed above for the appropriate SSC modulation types.

dF/dt Measurement

This is an informative test.

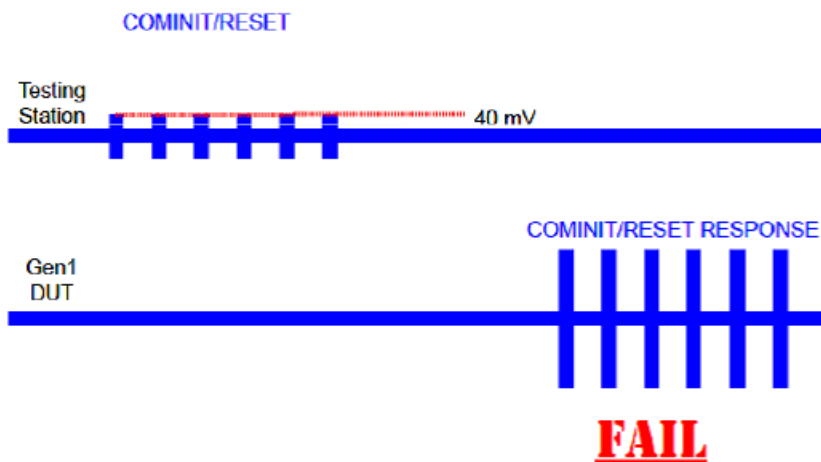
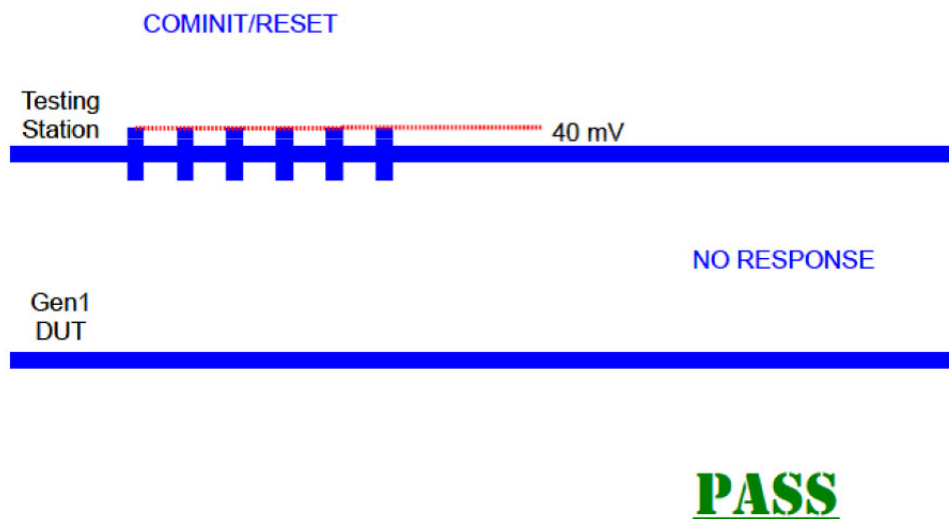
This measurement finds the rate of change of frequency with time. It calculates the deviation of UI value from the Nominal/standard UI. It calculates the first order derivative of unit interval with respect to time and reports the maximum.

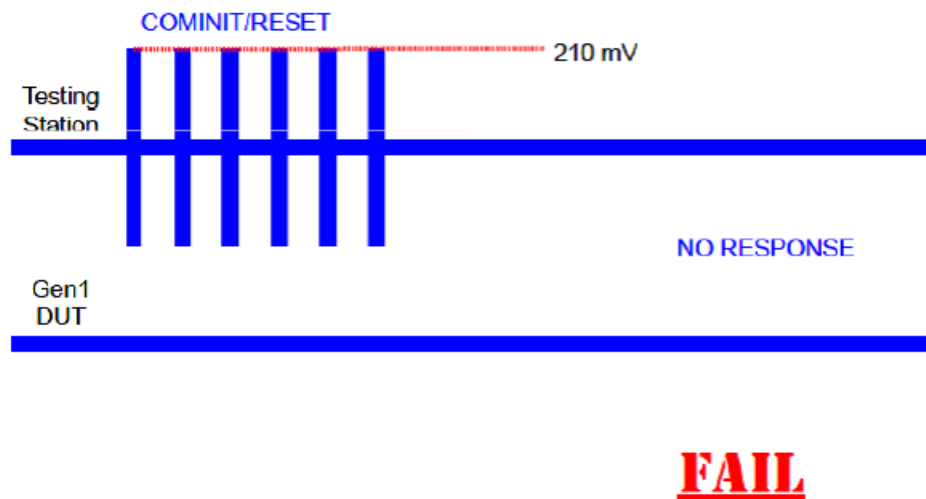
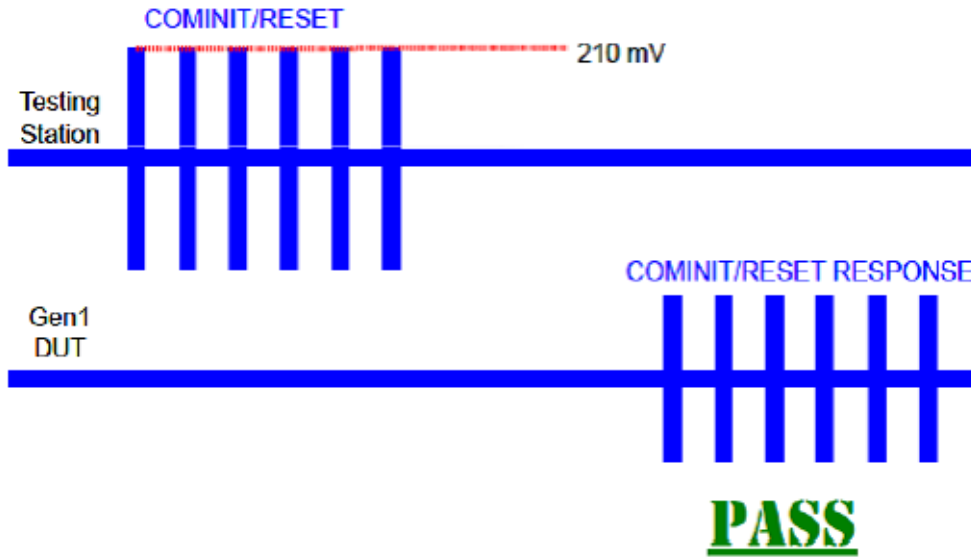
In this test, the slope of the SSC profile will be computed using post-processing techniques. The profile used will be the profile that was measured in the Modulation deviation and Balance test. An additional processing step will be performed on the profile, where a “sliding window” will be moved across the profile values to compute the slope. This window will have a width of 0.27 μ s, and the slope value for each horizontal point will be calculated.

The DFDT result must be no greater than conformant limit as per specification.

OOB01 - Signal Detection Threshold

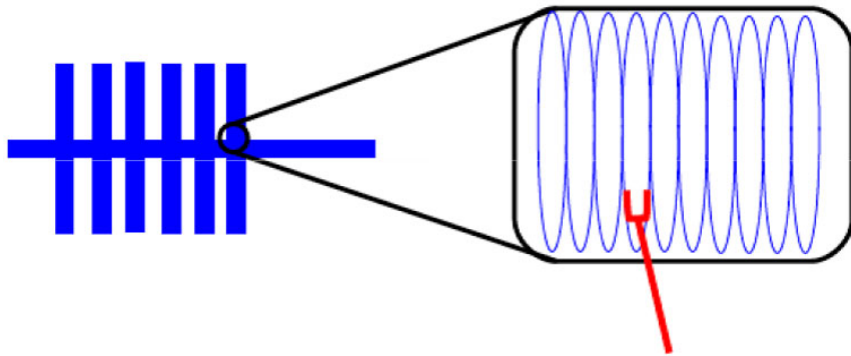
Determines whether a product properly ignores OOB signals that are too small, and that a product properly detect OOB signals that are properly sized.





OOB02 - UI during OOB Signaling

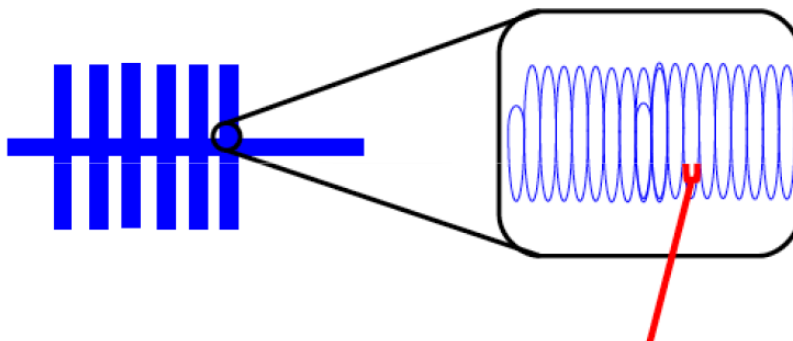
Determines that a product transmits OOB bursts using the proper UI (unit interval / bit time) of: 646.7 ps < t < 686.67 ps.



646.7 ps < t < 686.67 ps

Same as Gen1 SATA UI (1.5 Gbps)

PASS



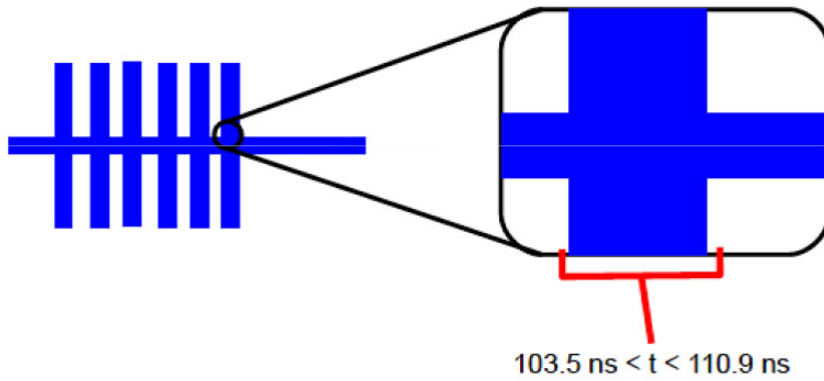
t < 646.7 ps or t > 686.67 ps

Same as Gen1 SATA UI (1.5 Gbps)

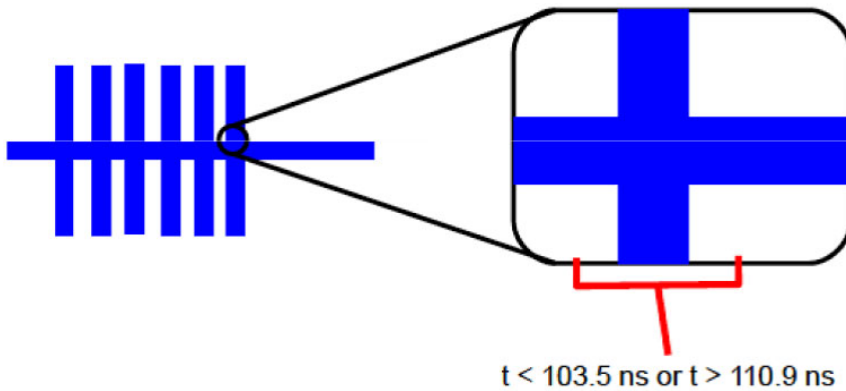
FAIL

OOB03 - COMINIT/RESET and COMWAKE Transmit Burst Length

Determines that a burst within any OOB signal is the proper length (160 OOB): $103.5 \text{ ns} < t < 110.9 \text{ ns}$.



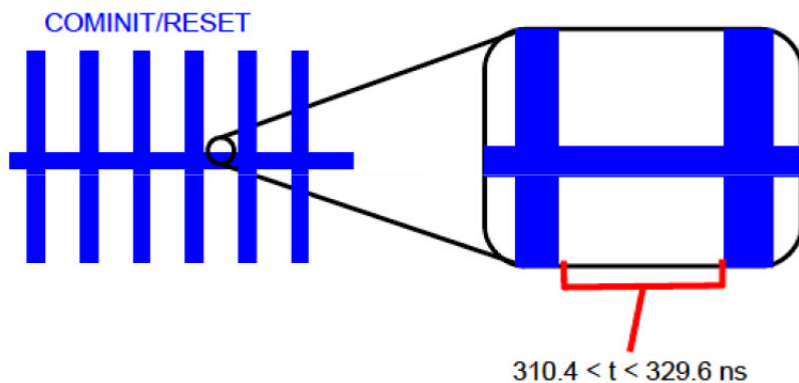
PASS



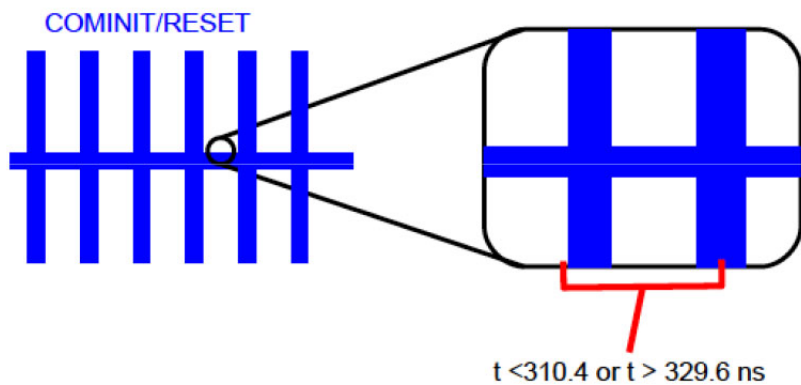
FAIL

OOB04 - COMINIT/RESET Transmit Gap Length

Determines that a Gap within a COMINIT/RESET signal is the proper length (480 OOB1) : $310.4 \text{ ns} < t < 329.6 \text{ ns}$



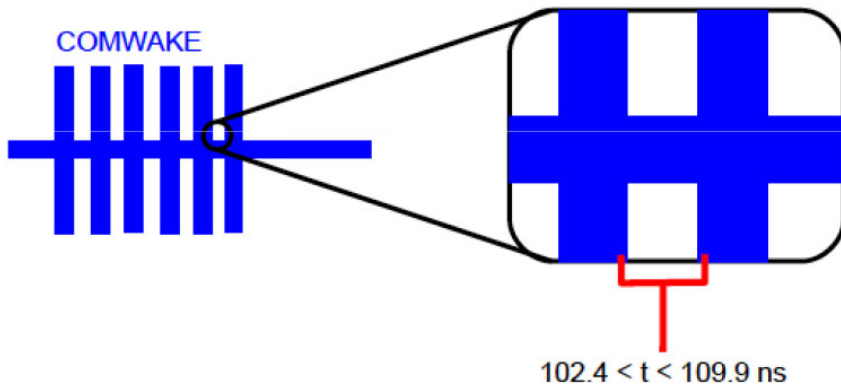
PASS



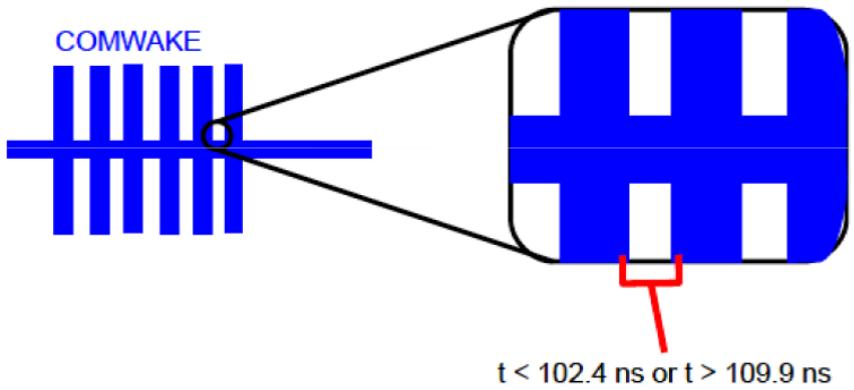
FAIL

OOB05 - COMWAKE Transmit Gap Length

Determines that a Gap within a COWAKE signal is the proper length (160 OOB1): $102.4 \text{ ns} < t < 109.9 \text{ ns}$.



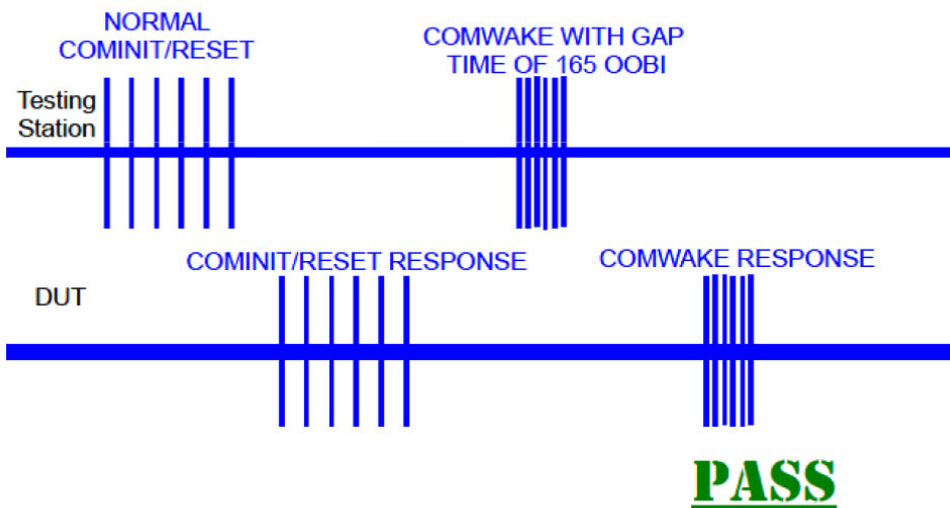
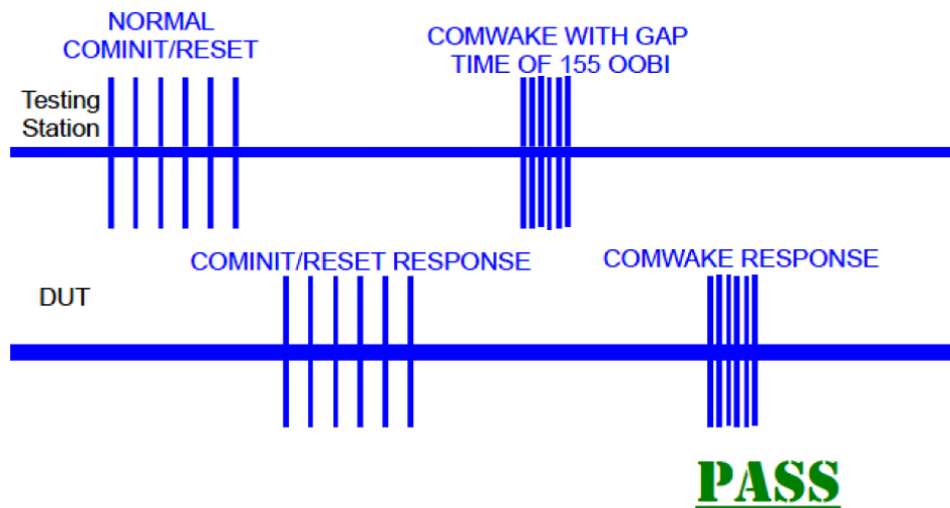
PASS



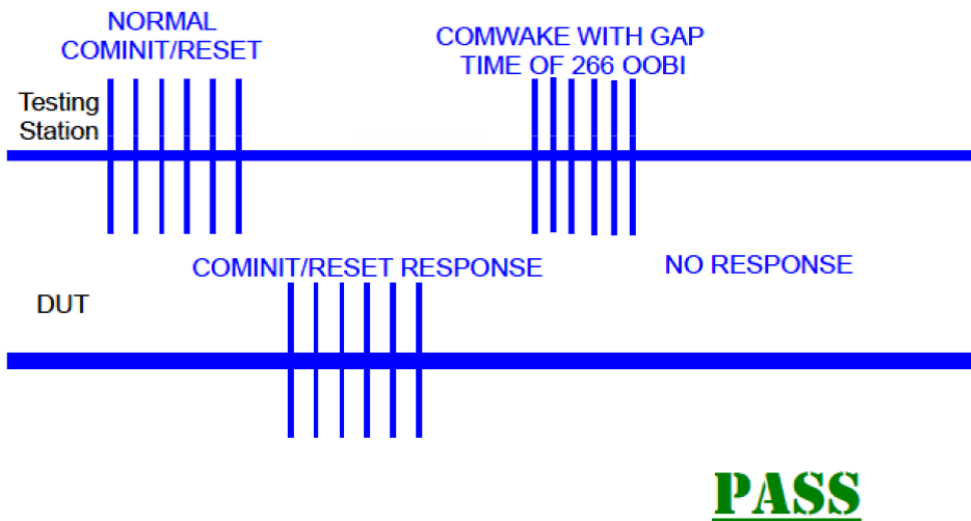
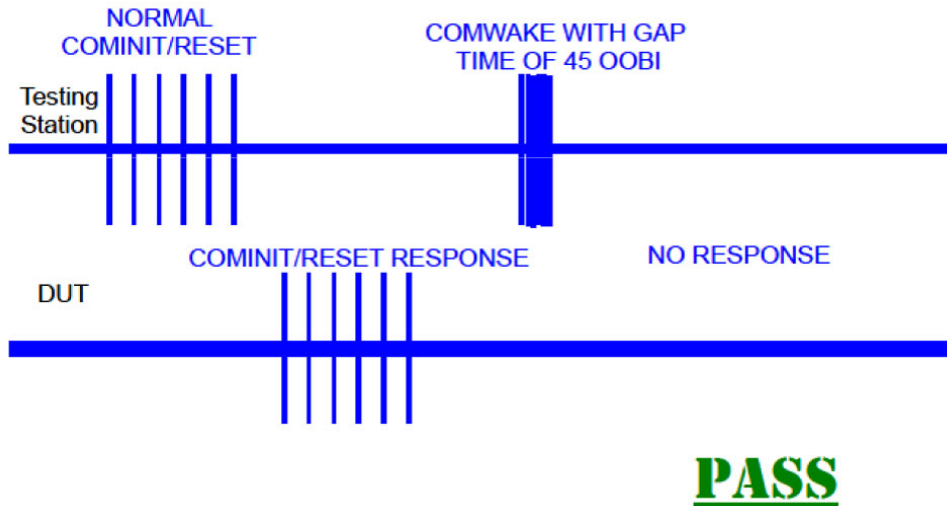
FAIL

OOB06 - COMWAKE Gap Detection Windows

Determine that a Product responds to COMWAKE signals with a Gap Length of 155 OOB I and 165 OOB I.

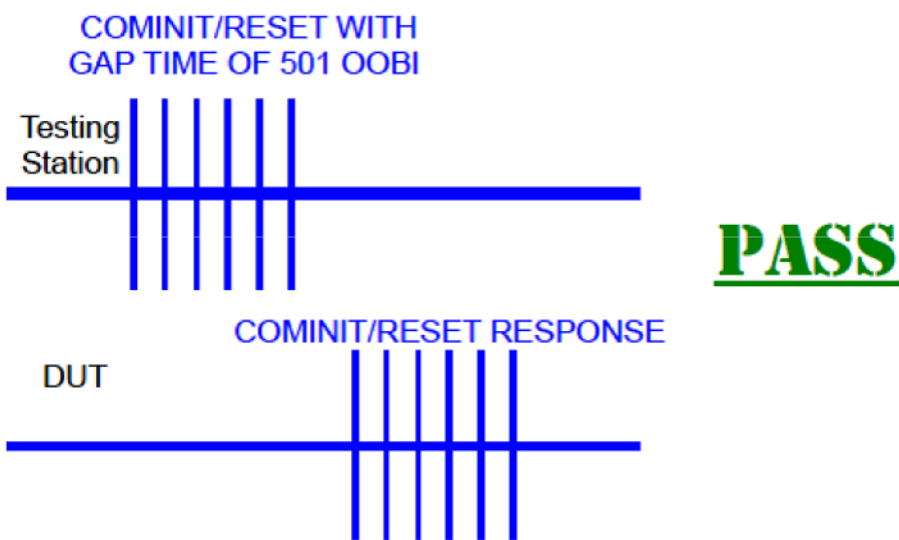
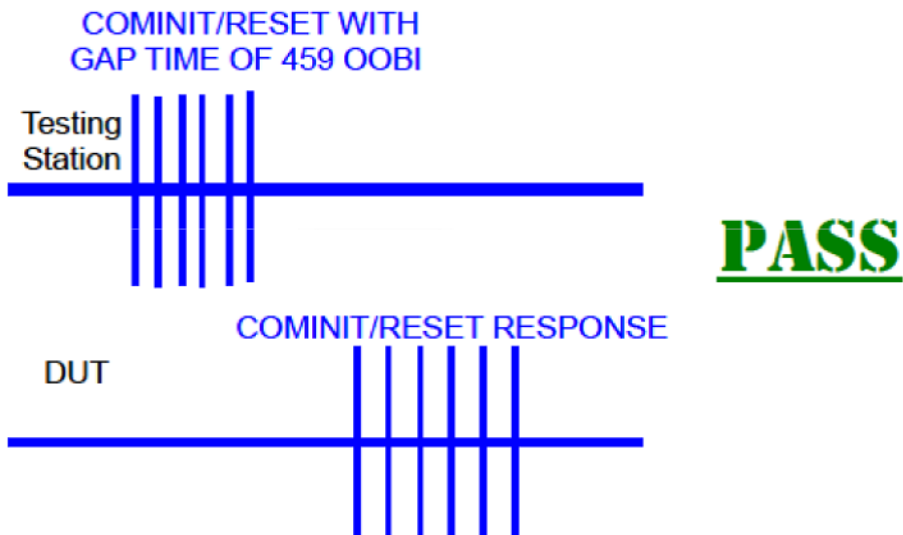


Determine that a Product ignores COMWAKE signals with a Gap Length of 45 OOB and 266 OOB.

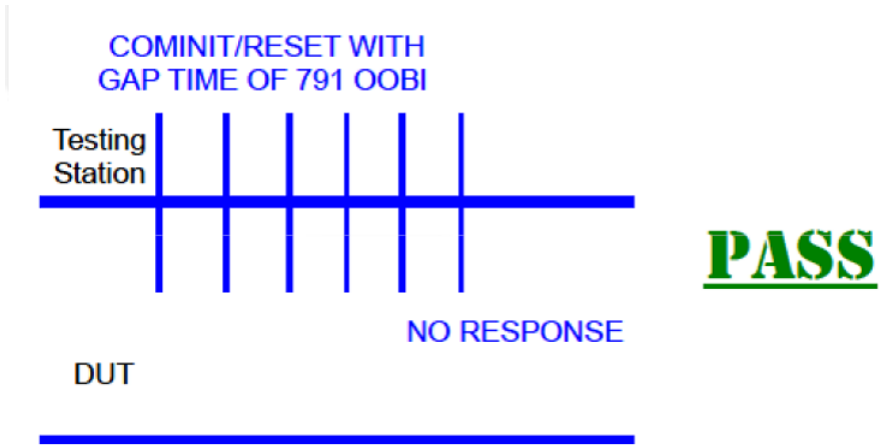
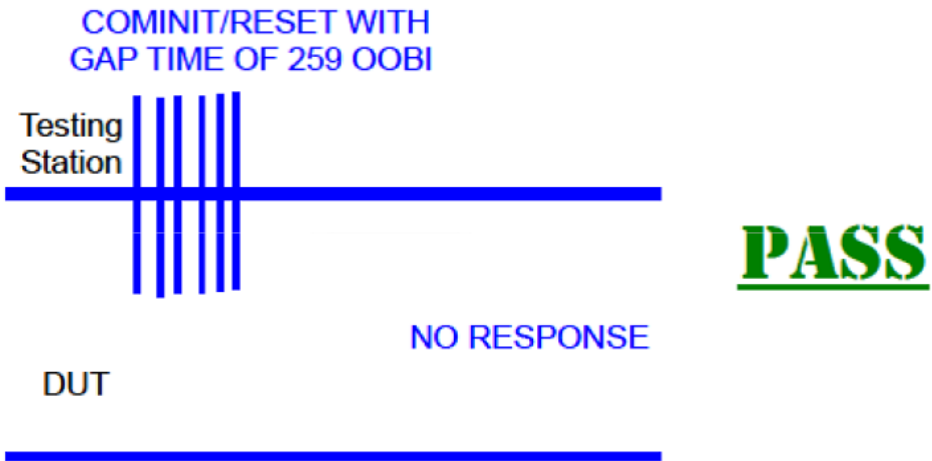


OOB07 - COMINIT Gap Detection Windows

Determine that a Product responds to COMINIT/RESET signals with a Gap Length of 459 OOB and 501 OOB.



Determine that a Product ignores COMINIT/RESET signals with a Gap Length of 259 OOB and 791 OOB.



RSG01 - GEN1 (1.5 Gb/S) Receiver Jitter Test

This test requirement is applicable to all products. (Gen1i/m/u-Gen3i/u)

The above test is based around testing the receiver channel using a directly synthesized framed composite pattern with a precise level of digital impairment using an Arbitrary Waveform Generator. SATA RSG 01 test is a Pass/Fail test on whether the receiver can tolerate a laboratory-grade impaired framed composite test signal.

This library includes all stressed patterns including RJ, Dj including CIC/ISI and with sine Pj source frequencies of 5 MHz, 10 MHz, 33 MHz and 62 MHz.

A jitter tolerance curve is then created that shows where the device fails. This information is useful in validating the design of the receiver.

RSG02 - GEN2 (3 Gb/S) Receiver Jitter Test

This test requirement is applicable to all PUTs claiming ability to run Gen2i/u and Gen3i/u rates.

The above test is based around testing the receiver channel using a directly synthesized framed composite pattern with a precise level of digital impairment using an Arbitrary Waveform Generator.

SATA RSG 02 test is a Pass/Fail test on whether the receiver can tolerate a laboratory-grade impaired framed composite test signal.

This library includes all stressed patterns including RJ, Dj including CIC/ISI and with sine Pj source frequencies of 5 MHz, 10 MHz, 33 MHz and 62 MHz.

RSG03 - GEN3 (6 Gb/S) Receiver Jitter Test

This test requirement is applicable to all PUTs claiming ability to run Gen3i/u rates.

The above test is based around testing the receiver channel using a directly synthesized framed composite pattern with a precise level of digital impairment using an Arbitrary Waveform Generator.

SATA RSG 03 test is a Pass/Fail test on whether the receiver can tolerate a laboratory-grade impaired framed composite test signal.

This library includes all stressed patterns including RJ, Dj including CIC/ISI and with sine Pj source frequencies of 5 MHz, 10 MHz, 33 MHz and 62 MHz.

RSG05 - Asynchronous Receiver Stress Test

This test requirement is applicable to all PUTs.

The above test is based around testing the receiver channel using a directly synthesized framed composite pattern with a precise level of digital impairment using an Arbitrary Waveform Generator.

SATA RSG 01 test is a Pass/Fail test on whether the receiver can tolerate a laboratory-grade impaired framed composite test signal.

This library includes all stressed patterns including RJ, Dj including CIC/ISI and with sine Pj source frequencies of: 62 MHz.

RSG06 - Asynchronous Receiver Stress Test with SSC

This test requirement is applicable to all PUTs.

The above test is based around testing the receiver channel using a directly synthesized framed composite pattern with a precise level of digital impairment using an Arbitrary Waveform Generator.

SATA RSG 01 test is a Pass/Fail test on whether the receiver can tolerate a laboratory-grade impaired framed composite test signal.

This library includes all stressed patterns including RJ, Dj including CIC/ISI and with sine Pj source frequencies of 62 MHz.

RMT01 - Gen1 (1.5 Gb/S) Receiver Margin Test

This test requirement is applicable to Gen1 products. (Gen1i/m/u)

When the RMT01 test is selected, jitter is synthesized at different frequencies and amplitudes increasingly until errors are detected on the frame error analyzer.

We have different ways to provide jitter:

1. Linear pass to Fail
2. High Performance Scan Pass to Fail
3. Select Scan Parameter Preset

The above test is based around testing the receiver channel using a directly synthesized framed composite pattern with a precise level of digital impairment using an Arbitrary Waveform Generator.

A jitter tolerance curve is then created that shows where the device fails. This information is useful in validating the design of the receiver.

RMT02 - Gen2 (3.0 Gb/S) Receiver Margin Test

This test requirement is applicable to Gen2 products. (Gen2i/m/u)

When the RMT02 test is selected, jitter is synthesized at different frequencies and amplitudes increasingly until errors are detected on the frame error analyzer.

We have different ways to provide jitter:

1. Linear pass to Fail
2. High Performance Scan Pass to Fail
3. Select Scan Parameter Preset

The above test is based around testing the receiver channel using a directly synthesized framed composite pattern with a precise level of digital impairment using an Arbitrary Waveform Generator.

A jitter tolerance curve is then created that shows where the device fails. This information is useful in validating the design of the receiver.

RMT03 - Gen3 (6.0 Gb/S) Receiver Margin Test

This test requirement is applicable to Gen3 products. (Gen3i/ u) When the RMT03 test is selected, jitter is synthesized at different frequencies and amplitudes increasingly until errors are detected on the frame error analyzer.

We have different ways to provide jitter:

1. Linear pass to Fail
2. High Performance Scan Pass to Fail
3. Select Scan Parameter Preset

The above test is based around testing the receiver channel using a directly synthesized framed composite pattern with a precise level of digital impairment using an Arbitrary Waveform Generator.

A jitter tolerance curve is then created that shows where the device fails. This information is useful in validating the design of the receiver.

Index

A

- About TekExpress, 5
- Acquire Status, 25
- Acquisitions tab, 21
- Activating the license, 3
- Analysis Status, 25
- Application commands, 58
- Application controls, 10
- Application directories, 7
- Application features, 5
- Application version
 - information, 4
- Application window, 10
- ApplicationStatus(), 77
- Auto Scroll, 26

C

- Channel selection commands, 72
- CheckSessionSaved(), 83
- Clear test log, 26
- Client programmatic interface
 - flowchart, 53
- Client proxy object, 51
- Code example, remote access, 56
- Command buttons, 10
- Commands
 - Connect(), 63
 - GetAcquireParameter(), 69
 - GetAnalyzeParameter(), 69
 - GetDutId(), 66
 - GetGeneralParameter(), 69
 - LockSession(), 64
 - SelectTest(), 71
 - SetAcquireParameter(), 67
 - SetAnalyzeParameter(), 67
 - SetDutId(), 66
 - SetGeneralParameter(), 67
- Commands list, 58
- Common measurements, 23
- Configuration Parameter
 - query, 69
- Configure Selected Measurement
 - commands, 73

- Configuring tests, 23
- Connect(), 63
- Connected instruments, 39
- Connection requirements, 36
- Conventions, 1

D

- Default directory, 7
- Deskew utility, 14
- Directories, 7
- Disable Popups command, 65
- Disconnect from Server
 - command, 84
- Disconnect(), 84
- DUT connection setup, 36
- DUT ID commands, 66
- DUT settings, 17
- DUT tab, 17
- DUT term, 1

E

- Email notifications, 13
- Email settings, 12
- Equipment setup, 36
- Exit the application, 9

F

- Features, 5
- File name extensions, 8

G

- Gear selection, 17
- GetAcquireParameter(), 69
- GetAnalyzeParameter(), 69
- GetCurrentStateInfo(), 78
- GetDutId(), 66
- GetGeneralParameter(), 69
- GetPassFailStatus(), 79
- GetReportParameter(), 80
- GetResultsValue(), 79
- GetResultsValueForSubMeasurements(), 80

- GetTimeOut(), 76
- Global settings, 23

I

- Installing the software, 3
- Instrument reset before each test
 - run, 14
- Instruments
 - selecting, 23
 - viewing connected, 39
- Instruments detected, 23
- Interface error codes, 75
- IP address connection
 - commands, 63

L

- Lane source, 21
- License activation, 3
- License information, 4
- LockSession(), 64
- Log files, 26
- Log View, 25

M

- Measurement selection
 - commands, 71
- Menus, 10
 - Options, 11
 - preferences, 27
- Message History, 26
- Mode, 23
- Move icon
 - moving the application, 10
- My TekExpress folder, 28
 - locating, 43

O

- Options menu, 11
- Options Menu
 - Instrument Control
 - Settings, 12
- Oscilloscopes supported, 5

- P**
- Panels
 - overview, 16
 - plots, 28
 - reports, 29
 - results, 26
 - Results, 27
 - Setup overview, 16
 - Status, 25
 - Pass/Fail summary, 31
 - Position the application, 10
 - Pre-run checklist, 45
 - Pre-Run checklist, 43
 - Program example, 56
 - Programmatic interface, 49
- Q**
- QueryStatus(), 77
- R**
- RecallSession(), 83
 - Related documentation, 1
 - Remote proxy object, 51
 - Report contents, 32
 - selecting, 30
 - Report file type, 30
 - Report names, 30
 - Report options, 30
 - Reports
 - view, 31
 - Reports panel, 29
 - Reset before test run, 14
 - Restore oscilloscope settings, 14
 - Results panel, 27
 - Run commands, 74
 - Run the application, 9
 - Run(), 74
- S**
- SATA test definitions, 85
 - Save oscilloscope settings before running tests, 14
 - SaveSession(), 83
 - SaveSessionAs(), 83
 - Saving test setups, 47
 - Schematic button, 18
 - Select lane source, 21
 - Select Required button, 18
 - Selecting DUT parameters, 17
 - SelectSuite(), 70
 - SelectTest(), 71
 - SendResponse(), 78
 - Server control, 50
 - Server locking commands, 64
 - Session files, 28
 - Session folders, 28
 - Session Status commands, 83
 - Set Configuration Parameters commands, 67
 - SetAcquireParameter(), 67
 - SetAnalyzeParameter(), 73
 - SetDutId(), 66
 - SetGeneralParameter(), 72
 - SetTimeout(), 76
 - Setting up tests, 41
 - Setup tabs
 - Acquisitions, 21
 - DUT, 17
 - Test Selection, 18
 - SetVerboseMode(), 65
 - Show application version, 4
 - Signal Path Compensation (SPC), 43
 - Signal source validation
 - overview, 21
 - setting, 21
 - Software, 6
 - Software installation
 - SATA software, 3
 - Specification version, 17
 - Status Panel, 25
 - Stop commands, 74
 - Stop(), 74
 - Suite selection commands, 70
 - Supported equipment, 5
 - System requirements, 6
 - Test Selection, 18
 - Technical support, 2
 - TekExpress client, 49
 - TekExpress client requirements, 50
 - TekExpress server, 49
 - Test configuration, 23
 - Test groups, 35
 - Test notifications, 22
 - Test parameters, 23
 - Test reports, 31
 - Test results, 27
 - Test results commands, 79
 - Test run instrument reset, 14
 - Test selection controls, 18
 - Test Selection tab, 18
 - Test setup files, 28
 - Test setup overview, 35
 - Test setup steps, 41
 - Test setups, 47
 - creating based on existing, 48
 - recalling, 48
 - saving, 47
 - Test Status, 25
 - Test Status commands, 77
 - Tests, 23
 - running, 43
 - selecting, 18
 - setting up, 35
 - Timeout Value commands, 76
 - TransferImages(), 81
 - TransferReport(), 81
- U**
- Unlock Server command, 84
 - UnlockSession(), 84
 - Untitled Session folder, 43
 - User comments
 - including in test reports, 30
 - location in reports, 31
 - User Comments, 17