

# TekExpress® 100GBASE-SR4 Transmitter Compliance Solution Printable Application Help





**TekExpress® 100GBASE-SR4 Transmitter Compliance Solution  
Printable Application 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.

### **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](http://www.tektronix.com) to find contacts in your area.

---

# Table of Contents

Welcome .....	v
---------------	---

## Getting help and support

Related documentation .....	1
Conventions .....	1
Technical support .....	2

## Getting started

Minimum system requirements .....	3
Instruments and accessories required .....	4
Installing the software .....	5
View software version .....	5
My TekExpress folder settings .....	5
Map My TekExpress folder to drive X: .....	5
Set My TekExpress folder permissions .....	6
Application directories .....	7
File name extensions .....	8

## Operating basics

Launch the application .....	9
Application panels overview .....	10
Global application controls .....	11
Application controls .....	11
Options menu overview .....	13
TekExpress instrument control settings .....	14
View connected instruments .....	14
Configure email settings .....	15
Setup panel .....	17
Setup panel overview .....	17
Set DUT parameters .....	17
Select tests .....	19
Set acquisition tab parameters .....	20
Set configuration tab parameters .....	21
Set preferences tab parameters .....	23
Status panel overview .....	24

Results panel .....	25
Results panel overview .....	25
View test-related files .....	26
Reports panel .....	27
Reports panel overview .....	27
Select report options .....	28
View a report .....	30
Report contents .....	30

## Pre-measurement calibration

Equipment setup .....	31
Oscilloscope compensation .....	32
Instrument noise .....	32
External attenuation calibration .....	33
Running tests .....	35

## Saving and recalling test setup

Test setup files overview .....	37
Save a test setup .....	37
Open (load) a saved test setup .....	37
Create a test setup from default settings .....	38
Create a test setup using an existing one .....	38

## 100GBASE-SR4 Transmitter compliance measurements

Signaling Rate .....	39
Average Launch Power .....	39
Optical Modulation Amplitude .....	41
Transmitter and Dispersion Eye Closure .....	42
Launch Power in OMA minus TDEC .....	46
Average Launch Power of Off Transmitter .....	46
Extinction Ratio .....	47
Transmitter Eye Mask .....	48

## TekExpress programmatic interface

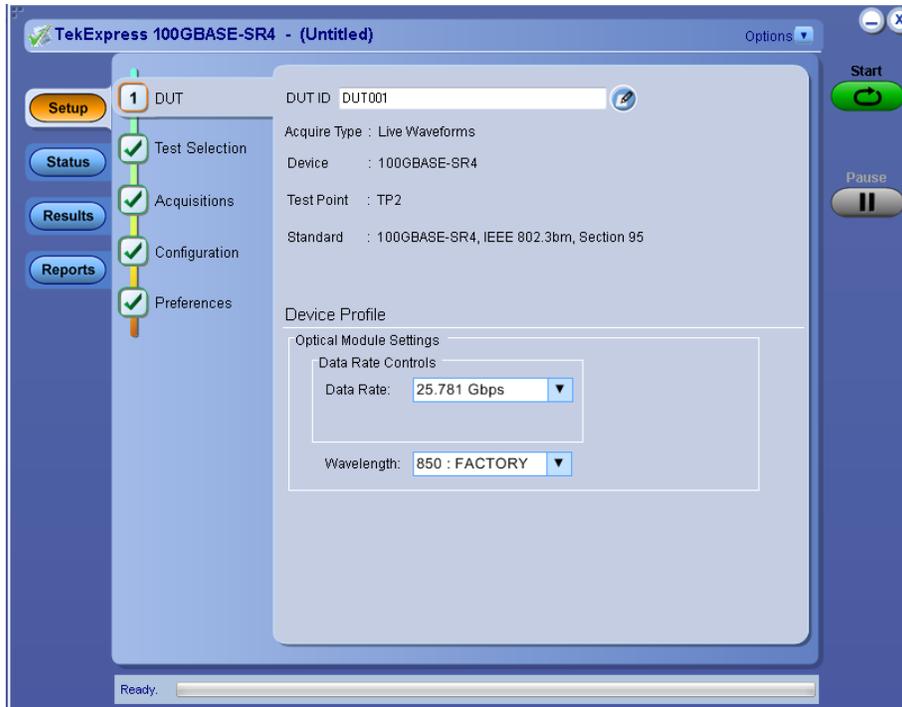
About the programmatic interface .....	51
To enable remote access .....	51
Requirements for developing TekExpress client .....	53

Remote proxy object .....	54
Client proxy object .....	55
Client programmatic interface example .....	56
Program remote access code example .....	58
100GBASE-SR4 programmer interface commands .....	59
ApplicationStatus() .....	59
ChangeDutId() .....	60
CheckSessionSaved() .....	61
Connect() .....	62
Disconnect() .....	63
GetCurrentStateInfo() .....	63
GetDutId() .....	65
GetPassFailStatus() .....	66
GetReportParameter() .....	66
GetResultsValue() .....	68
GetTimeOut() .....	69
LockSession() .....	69
QueryStatus() .....	70
RecallSession() .....	72
Run() .....	72
SaveSession() .....	73
SaveSessionAs() .....	74
SendResponse() .....	74
SelectDevice() .....	76
SelectSuite() .....	76
SelectTest() .....	77
SetDutId() .....	77
The SetAcquireParameter command .....	79
paramString values for SetAcquireParameter command .....	80
The SetGeneralParameter command .....	81
SetTimeOut() .....	85
setVerboseMode() .....	86
Status() .....	87
Stop() .....	87
TransferImages() .....	88
TransferReport() .....	89
UnlockSession() .....	90

## References

100GBASE-SR4 technology overview .....	93
Optical configurations .....	94
Clock Recovery Unit (CRU) .....	94
Trigger Source .....	95
Phase reference characterization .....	95
Parameters .....	95
About application parameters .....	95
Setup panel configuration parameters .....	95
Reports panel parameters .....	98

# Welcome



Welcome to the TekExpress 100GBASE-SR4 Automated Measurement Solution Software application. The TekExpress 100GBASE-SR4 provides turnkey testing and characterization of Tx Optical properties key to the SR4 Short Reach Ethernet Specifications outlined in IEEE 802.3bm. Automation options help the customers to meet their conformance testing needs and generate detailed reports. User-defined Mode lets customers make changes to the test limits and perform margin testing as part of extended product characterization.

## Key features of TekExpress 100GBASE-SR4 include:

- 80SSR4 is a comprehensive set of 100GBASE-SR4 (IEEE 802.3bm) optical transmitter characterization measurements, including TDEC. It's designed to offer a checklist of all relevant tests to verify conformance to the SR4 optical specification.
- 80STDEC offers streamlined high performance Transmitter and Dispersion Eye Closure (TDEC) on a variety of Tektronix optical acquisition modules. Ideal for manufacturing settings and minimal conformance validation applications.

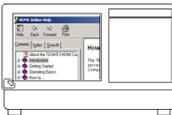


# Getting help and support

## Related documentation

The following documentation is available as part of the TekExpress® 100GBASE-SR4 Solution application.

**Table 1: Product documentation**

Item	Purpose	Location
Help	Application operation and User Interface help	
PDF of the help	Printable version of the compiled help	 www.Tektronix.com PDF file that ships with 100GBASE-SR4 Solution software distribution ( <i>TekExpress 100GBASE-SR4-Automated-Test-Solution-Software-Printable-Help-EN-US.pdf</i> ).

### See also

[Technical support](#)

## Conventions

Help uses the following conventions:

- The term "Application," and "Software" refers to the TekExpress 100GBASE-SR4 Solution application.
- The term "DUT" is an abbreviation for Device Under Test.
- The term "select" is a generic term that applies to the two methods of choosing a screen item (button, control, list item): using a mouse or using the touch screen.

**Table 2: Icon descriptions**

Icon	Meaning
	This icon identifies important information.
	This icon identifies conditions or practices that could result in loss of data.
	This icon identifies additional information that will help you use the application more efficiently.

## 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](#) at the front of this document for contact 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
- Modules used
- Your name, company, mailing address, phone number, FAX number
- Please indicate if you would like to be contacted by Tektronix about your suggestion or comments.

### Application specific information

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

# Getting started

## Minimum system requirements

The following table shows the minimum system requirements to install and run the TekExpress 100GBASE-SR4 Solution.

**Table 3: System requirements**

Component	Description
<b>Oscilloscope</b>	<ul style="list-style-type: none"><li>■ Tektronix DSA8300 Digital Serial Analyzer</li><li>■ Firmware Version: 6.3.1.3 or greater</li><li>■ 80SJNB Software Version: 3.2.4.0 or greater</li><li>■ Opt ADVTRIG</li><li>■ Opt JNB01/02</li><li>■ 80CXX series Optical Sampling module</li></ul>
<b>Processor</b>	Same as the oscilloscope
<b>Operating System</b>	Same as the oscilloscope:
<b>Memory</b>	Same as the oscilloscope
<b>Hard Disk</b>	Same as the oscilloscope
<b>Display</b>	Super VGA resolution or higher video adapter (800 x 600 minimum video resolution for small fonts or 1024 x 768 minimum video resolution for large fonts). The application is best viewed at 96 dpi display settings <sup>1</sup>
<b>Firmware</b>	<ul style="list-style-type: none"><li>■ TekScope 6.3.1.3 or greater (for Windows 7)</li><li>■ 80SJNB Software Version: 3.2.4.0 or greater</li></ul>
<b>Software</b>	<ul style="list-style-type: none"><li>■ IronPython 2.7.3 installed</li><li>■ PyVisa 1.0.0.25 installed</li><li>■ Microsoft .NET 4.0 Framework</li><li>■ Microsoft Internet Explorer 7.0 SP1 or greater, or other Web browser for viewing reports</li><li>■ Adobe Reader software 7.0 or greater for viewing portable document format (PDF) files</li></ul>
<b>Other Devices</b>	<ul style="list-style-type: none"><li>■ Microsoft compatible mouse or compatible pointing device.</li><li>■ Two USB ports (four USB ports recommended).</li></ul>

<sup>1</sup> If TekExpress is running on an instrument that has a video resolution less than 800x600, connect and configure a second monitor to the instrument.

## Instruments and accessories required

100GBASE-SR4 application is launched on DSA8300 sampling scope. The following table lists the instruments and accessories required for this application.

**Table 4: Instruments and accessories required for 100GBASE-SR4 application**

Instrument/Accessory	Model number	Quantity
Sampling Oscilloscope	DSA8300	1
Clock Recovery Unit	CR286A	1
Optical Modules	80C15-CRTP-MM for Compliance mode 80C10/80C14/80C15/80C15-CRTP-MM for User Defined mode	1
Optical Power Meter	Any make which supports 850 nm wavelength	1
Phase Reference	82A04B (Optional) <sup>2</sup>	1
SMA Cables	174-6023-00 : To connect sub rate clock out of CRU to trigger <sup>3</sup>	1
	174-6023-00 : To connect clock out of CRU to phase reference module <sup>3</sup>	1
	174-6023-00 : To connect electrical output of the 80C15 module to CRU input. Deskew the cables before use.	2
VBR (Variable Back Reflector)	A Splitter and a Variable Back Reflector is required for TDEC measurement, as specified in <a href="#">Equipment setup</a> . The splitter should be the smallest split possible to meet the back reflection requirement. For most accurate result measure the back reflection.	1

<sup>2</sup> Required to reach jitter noise floors below 100fsec

<sup>3</sup> Available with 80A08 accessory kit

## Installing the software

Follow the steps to download and install the latest TekExpress 100GBASE-SR4 Solution. See [Minimum system requirements](#) for compatibility.

1. Type the URL [www.tek.com](http://www.tek.com) in the address bar of web browser and click Software Downloads
2. Enter **TekExpress 100GBASE-SR4 Solution** in the *Enter your keywords* field, and click **Search**
3. Select the latest version of software and follow the instructions to download. Copy the executable file into the oscilloscope.
4. Double-click the executable and follow the on-screen instructions. The software is installed at C:\Program Files\Tektronix\TekExpress\TekExpress 100GBASE-SR4\
5. Map *My TekExpress folder to drive X:* and Set *My TekExpress folder permissions*
6. Select **Application > 100GBASE-SR4** from the TekScope menu to *launch the application*

## View software version

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

To view version information for 100GBASE-SR4, click  button in the TekExpress application and select **About TekExpress**.



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

## My TekExpress folder settings

### Map My TekExpress folder to drive X:

The first time you run TekExpress 100GBASE-SR4, it creates the following folders on the oscilloscope:

- \My Documents\My TekExpress\100GBASE-SR4
- \My Documents\My TekExpress\100GBASE-SR4\Untitled Session

Shared **My TekExpress** folder is mapped to drive **X:** on the instrument running the 100GBASE-SR4 application. 100GBASE-SR4 uses this shared folder to save session waveform files and for other application file transfer operations.

Follow the below procedure to map the My TekExpress folder on the instrument to be drive X:

1. Open Microsoft Windows Explorer.
2. From the Windows Explorer menu, click **Computer** and select **Map network drive**.

3. Select the Drive letter as **X:** (if there is any previous connection on X:, disconnect it first through **Tools > Disconnect Network drive** menu of Windows Explorer. If you do not see the Tools menu, press the **Alt** key).
4. In the **Folder** field, enter the remote My TekExpress folder path (for example, \\192.158.97.65\My TekExpress).

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

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

---

**NOTE.** The My TekExpress folder has the share name format *<domain><user ID>My TekExpress*.

If the instrument is not connected to a domain, the share name format is *<instrument name><user ID>My TekExpress*.

---

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

---

**See also.** [Set My TekExpress folder permissions](#)

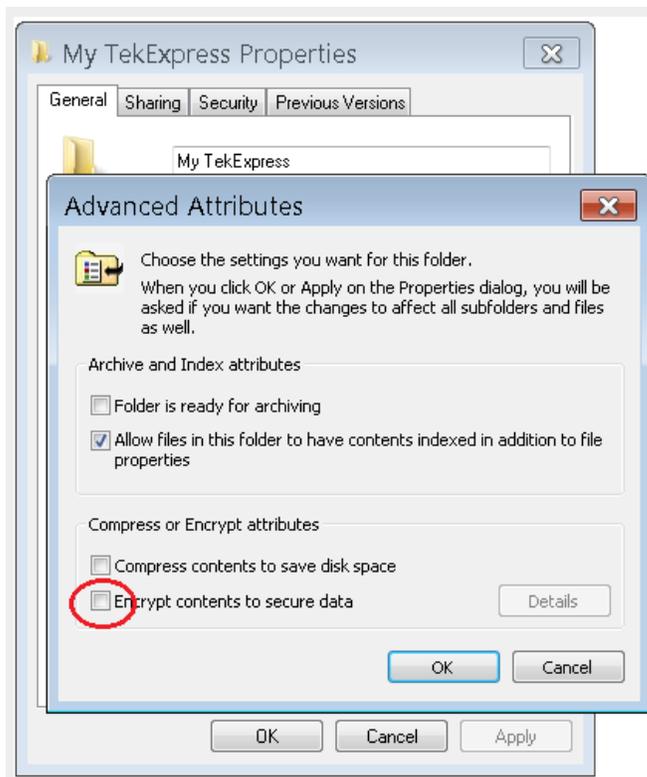
[Application directories](#)

[File name extensions](#)

## Set My TekExpress folder permissions

Make sure that the My TekExpress folder has read and write access. Also verify that the folder is not set to be encrypted:

1. Right-click the folder and select **Properties**.
2. Select the **General** tab and then click **Advanced**.
3. In the Advanced Attributes dialog box, make sure that the option **Encrypt contents to secure data** is NOT selected.



4. Click the **Security** tab and verify that the correct read and write permissions are set.

See also. [Map My TekExpress folder to Drive X:](#)

[Application directories](#)

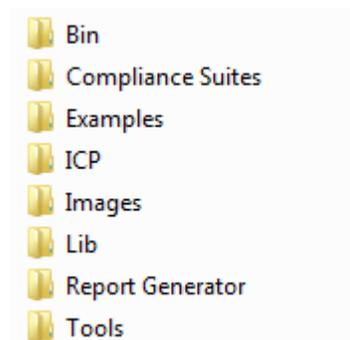
[File name extensions](#)

## Application directories

### TekExpress 100GBASE-SR4 application

The TekExpress 100GBASE-SR4 application files are installed at the following location:

C:\Program Files\Tektronix\TekExpress\TekExpress 100GBASE-SR4



The following table lists the application directory names and their purpose.

**Table 5: Application directories and usage**

Directory names	Usage
Bin	Contains TekExpress 100GBASE-SR4 application libraries
Compliance Suites	Contains compliance-specific files
Examples	Contains various support files
ICP	Contains instrument and TekExpress 100GBASE-SR4 application-specific interface libraries
Images	Contains images of the TekExpress 100GBASE-SR4 application
Lib	Contains utility files specific to the TekExpress 100GBASE-SR4 application
Report Generator	Contains style sheets for report generation
Tools	Contains instrument and TekExpress 100GBASE-SR4 application-specific files

**See also**[View test-related files](#)[File name extensions](#)**File name extensions**

The TekExpress 100GBASE-SR4 application uses the following file name extensions:

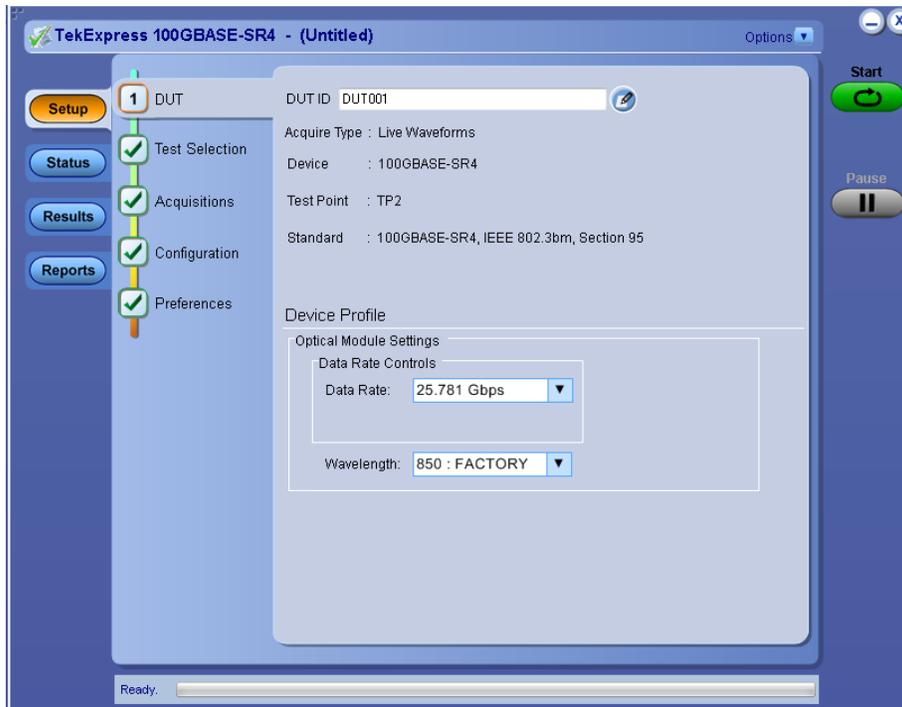
File name extension	Description
.TekX	Application session files (the extensions may not be displayed)
.py	Python sequence file
.xml	Test-specific configuration information (encrypted) files Application log files
.csv	Test waveform files
.mht	Test result reports (default) Test reports can also be saved in HTML format
.pdf	Test result reports Application help document
.xslt	Style sheet used to generate reports

**See also**[View test-related files](#)[Application directories](#)

# Operating basics

## Launch the application

To launch the TekExpress 100GBASE-SR4 Solution, select **Application > 100GBASE-SR4** from the TekScope menu.



When you launch the application for the first time, the file `C:\Users\\My Documents\My TekExpress\100GBASE-SR4\Resources.xml` is mapped to drive X:. This file contains information about available network-connected instruments. The session files are stored in `X:\100GBASE-SR4\`. If this file is not found, then the application runs Instrument Discovery Program to detect the connected instruments before launching 100GBASE-SR4 Solution.

---

**NOTE.** Map *My TekExpress folder to drive X:* and set *My TekExpress folder permissions* before running tests for the first time.

---

If the application goes behind the oscilloscope application, click **Application > 100GBASE-SR4** to bring it to the front. To keep the 100GBASE-SR4 application window on top, select **Keep On Top** from the 100GBASE-SR4 *Options menu*.

## See also

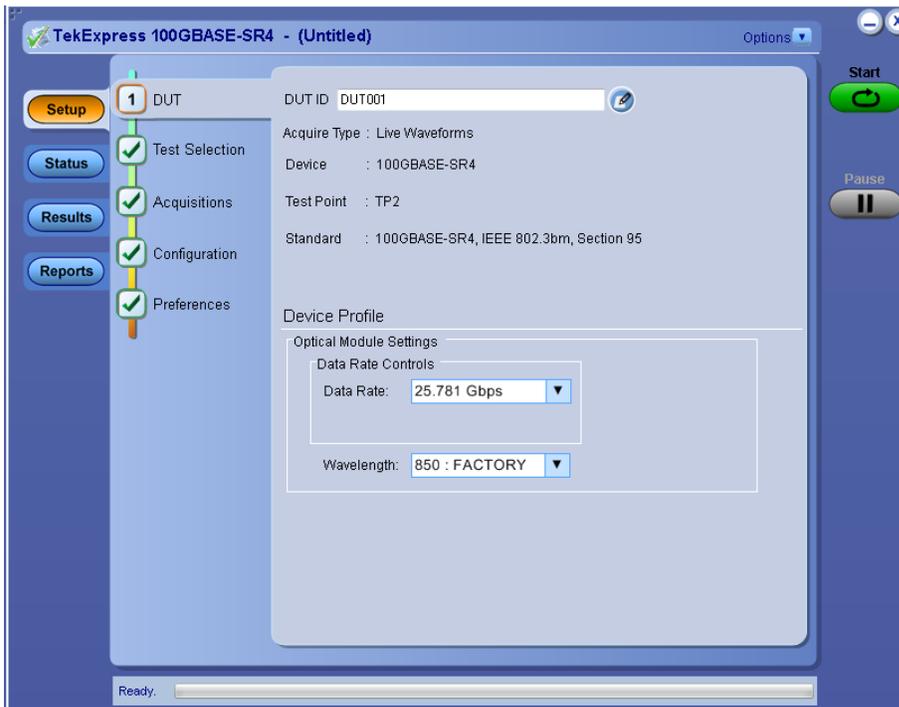
[Required My TekExpress folder settings](#)

[Application controls](#)

[Application panel overview](#)

## Application panels overview

TekExpress 100GBASE-SR4 Solution uses panels to group related configuration, test, and results settings. Click any button to open the associated panel. A panel may have one or more tabs that list the selections available in that panel. Controls in a panel can change depending on settings made in that panel or another panel.



**Table 6: Application panels overview**

Panel Name	Purpose
<a href="#">Setup panel</a>	The Setup panel shows the test setup controls. Click the <b>Setup</b> button to open this panel. Use this panel to: <ul style="list-style-type: none"> <li>■ <a href="#">Set DUT tab parameters</a></li> <li>■ <a href="#">Select tests</a></li> <li>■ <a href="#">Set acquisition tab parameters</a></li> <li>■ <a href="#">Set configuration tab parameters</a></li> <li>■ Set test notification parameters in the <a href="#">Preferences tab</a></li> </ul>
<a href="#">Status panel</a>	View the progress and analysis status of the selected tests, and view test logs.
<a href="#">Results panel</a>	View a summary of test results and select result viewing preferences.
<a href="#">Reports panel</a>	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](#)

## Global application controls

### Application controls

This section describes the application controls.

**Table 7: Application controls description**

Item	Description
<a href="#">Options menu</a> 	Menu to display global application controls.
<a href="#">Test Panel buttons</a> 	Controls that open panels for configuring test settings and options.

Item	Description
<p>Start / Stop button</p> 	<p>Use the Start button to start the test run of the measurements in the selected order. If prior acquired measurements are not cleared, then new measurements are added to the existing set. The button toggles to the Stop mode while tests are running. Use the Stop button to abort the test.</p>
<p>Pause / Continue button</p> 	<p>Use the Pause button to temporarily pause the acquisition. When a test is paused, this button changes to "Continue."</p>
<p>Clear button</p> 	<p>Use the Clear button to clear all existing measurement results. Adding or deleting a measurement, or changing a configuration parameter of an existing measurement, also clears measurements. This is to prevent the accumulation of measurement statistics or sets of statistics that are not coherent. This button is available only on <a href="#">Results panel</a>.</p>
<p>Application window move icon</p> 	<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>
<p>Minimize icon</p> 	<p>Click to minimize the application.</p>
<p>Close icon</p> 	<p>Click to close the application.</p>

See also. [Application panel overview](#)

## Options menu overview

To access Options menu, click  in the upper-right corner of the application. It has the following:

### Options menu

Default Test Setup
Open Test Setup
Save Test Setup
Save Test Setup As
Open Recent <span style="float: right;">▶</span>
Instrument Control Settings
Keep On Top
Email Settings
Open Current Suite RunSession
Help
About TekExpress

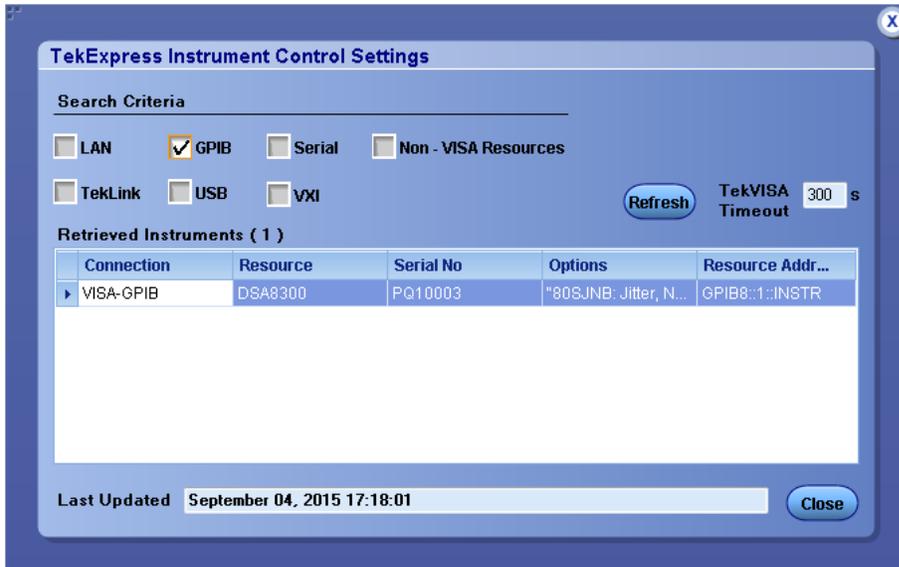
Menu	Function
Default Test Setup	Opens an untitled test setup with defaults selected <b>Data Rate:</b> 25,781 Gbps <b>Wavelength:</b> 850 nm <b>OMA Method:</b> OME-Eye <b>Trigger Source:</b> TekCRU <b>Phase Reference:</b> Checked <b>SR4 Filter:</b> 100GBASE-R4 <b>SR4 Bandwidth:</b> None <b>TDEC Filter:</b> None <b>TDEC Bandwidth:</b> 16.8 GHz Values displayed in <b>Wavelength</b> , <b>Filter</b> , and <b>Bandwidth</b> fields depend on the installed optical module.
Open Test Setup	Opens a saved test setup
Save Test Setup	Saves the current test setup
Save Test Setup As	Saves the current test setup with a different file name or file type
Open Recent	Displays the recently opened test setups to open
<i>Instrument Control Settings</i>	Detects, lists, and refreshes the connected instruments found on specified connections (LAN, GPIB, USB, and so on)
Keep On Top	Keeps the TekExpress 100GBASE-SR4 application on top in the desktop
<i>Email Settings</i>	Use to configure email options for test run and results notifications
Open Current Suite RunSession	Allows the user to select the specific run setup, from the saved session
Help	Displays the TekExpress 100GBASE-SR4 help
<i>About TekExpress</i>	<ul style="list-style-type: none"> <li>■ Displays application details such as software name, version number, and copyright</li> <li>■ Provides a link to the end-user license agreement</li> <li>■ Provides a link to the Tektronix Web site</li> </ul>

See also. [Application controls](#)

## TekExpress instrument control settings

Use TekExpress Instrument Control Settings dialog box to search the instruments (resources) connected to the application. You can use the Search Criteria to search the connected instruments depending on the connection type. The details of the connected instrument is displayed in the Retrieved Instruments window.

You can access this dialog box from the **Options** menu.



The connected instruments displayed here can be selected for use under Global Settings in the test configuration section.

**NOTE.** Under **Instrument Control Settings**, select GPIB Option (Default setting), when using TekExpress 100GBASE-SR4 application.

See also. [Options menu overview](#)

## View connected instruments

Use TekExpress Instrument Control Settings dialog box to search the instruments (resources) connected to the application. The application uses TekVISA to discover the connected instruments.

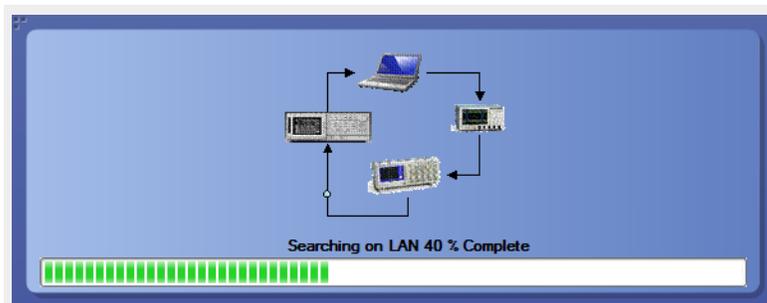
**NOTE.** The correct instruments required for the test setup must be connected and recognized by the application before running the test.

To refresh the list of connected instruments:

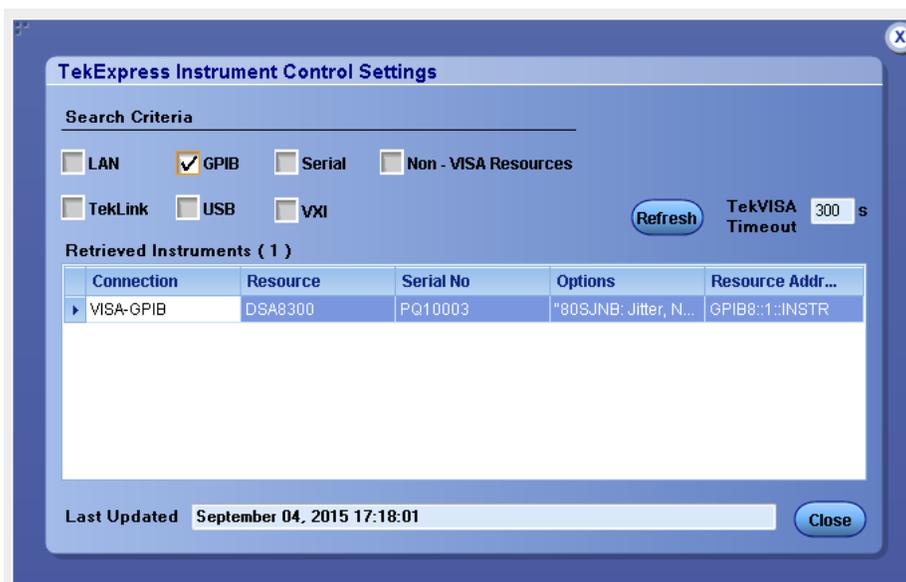
1. From the Options menu, select **Instrument Control Settings**.
2. In the **Search Criteria** section of the Instrument Control Settings dialog box, select the connection types of the instruments to search.

Instrument search is based on the VISA layer, but different connections 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.

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



4. After searching, the dialog box lists the instrument-related details based on the search criteria. For example, For the Search Criteria as LAN and GPIB, the application displays all LAN and GPIB instruments connected to the application.



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. [Equipment connection setup](#)

### Configure email settings

Use the Email Settings utility to get notified by email when a measurement completes, or produces any error condition. Follow the steps to configure 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 recipient email addresses. To include multiple addresses, separate the addresses with commas.
3. (Required) For Sender's Address, enter the email address used by the instrument. This address consists of the instrument name, followed by an underscore, followed by the instrument serial number, then the @ symbol, and the email server ID. For example: DSA8300\_B130099@yourcompany.com.

4. (Required) In the Server Configuration section, type the SMTP Server address of the Mail server configured at the client location, and the SMTP Port number, in the corresponding fields.

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

---

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

---

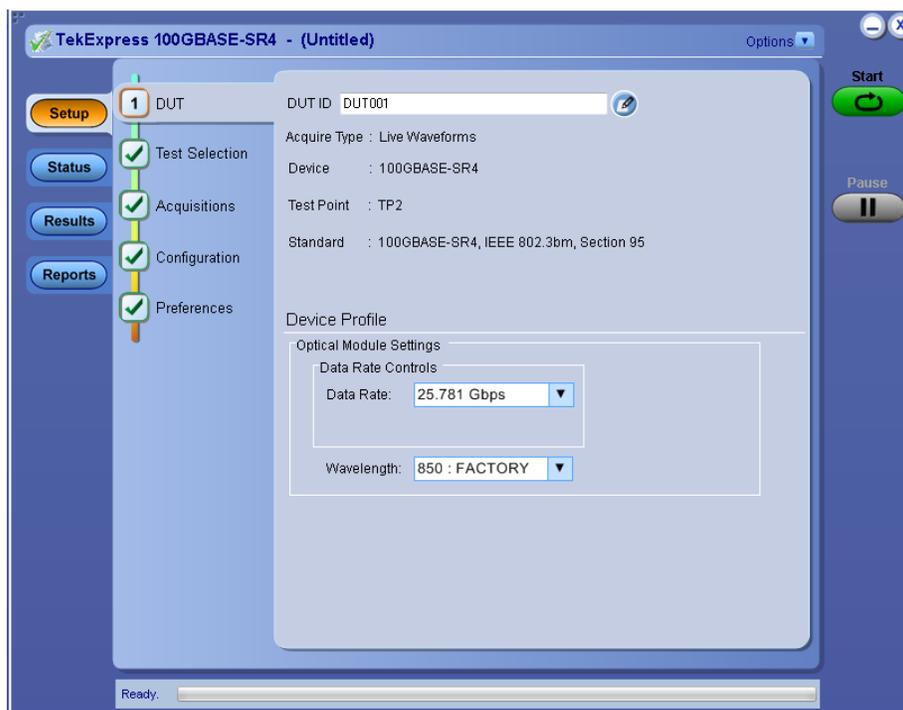
5. In the Email Attachments section, select from the following options:
  - **Reports:** Select to receive the test report with the notification email.
  - **Status Log:** Select to receive the test status log with the notification email. If you select this option, then also select whether you want to receive the full log or just the last 20 lines.
6. In the Email Configuration section:
  - Enter a maximum file size for the email message. Messages with attachments larger than this limit will not be sent. The default is 5 MB.
  - 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.

### Email Settings

## Setup panel

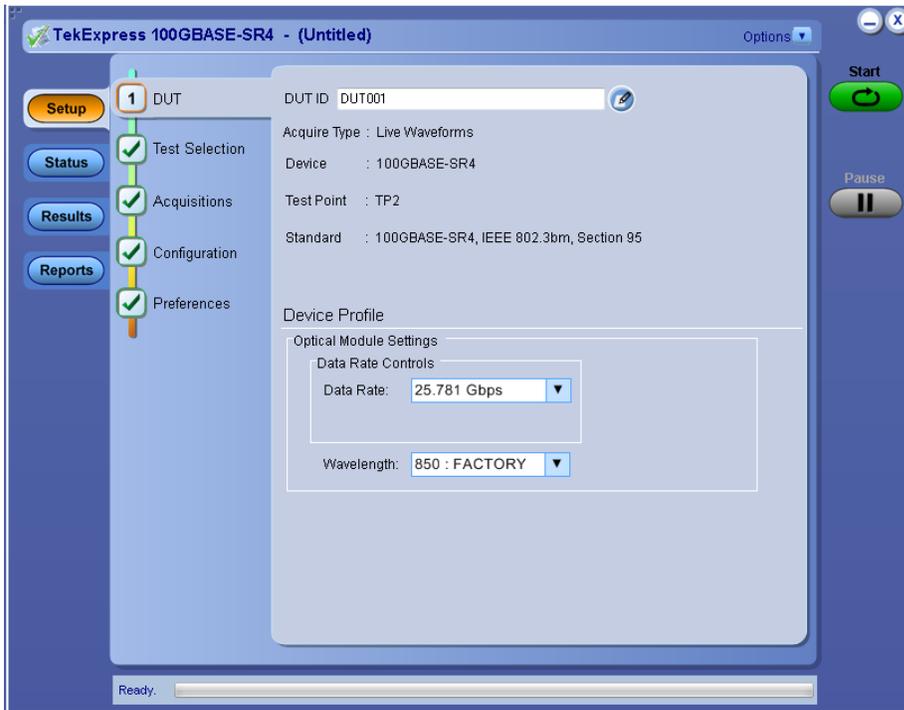
### Setup panel overview

The Setup panel contains sequentially ordered tabs that help you guide through the test setup and execution process.



### Set DUT parameters

Use the DUT tab to select parameters for the device under test. These 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.



Click **Setup > DUT** to access the DUT parameters:

**Table 8: DUT tab settings**

Setting	Description
DUT ID	Adds an optional text label for the DUT to reports. The default value is DUT001. The maximum number of characters is 32. You cannot use the following characters in an ID name: (.,,.,.,.,.,.,.,./,:?"<> *)
 Comments icon (to the right of the DUT ID field)	Opens a Comments dialog box in which to enter optional text to add to a report. Maximum size is 256 characters. To enable or disable comments appearing on the test report. See <a href="#">Select report options</a> for details.
Acquire Type	Live waveforms. This application performs analysis on live waveforms only. This field is not editable.
Device	100GBASE-SR4 measurements are taken on Device only. This field is not editable.
Test Point	TP2. This field is not editable.
Standard	100GBASE-SR4, IEEE 802.3bm, Section 95. This field is not editable.
<b>Device Profile</b>	
<b>Optical Module Settings</b>	
Data Rate Controls - Data Rate	Set the data rate to be tested.
Wavelength	Select the wavelength from the drop-down list.

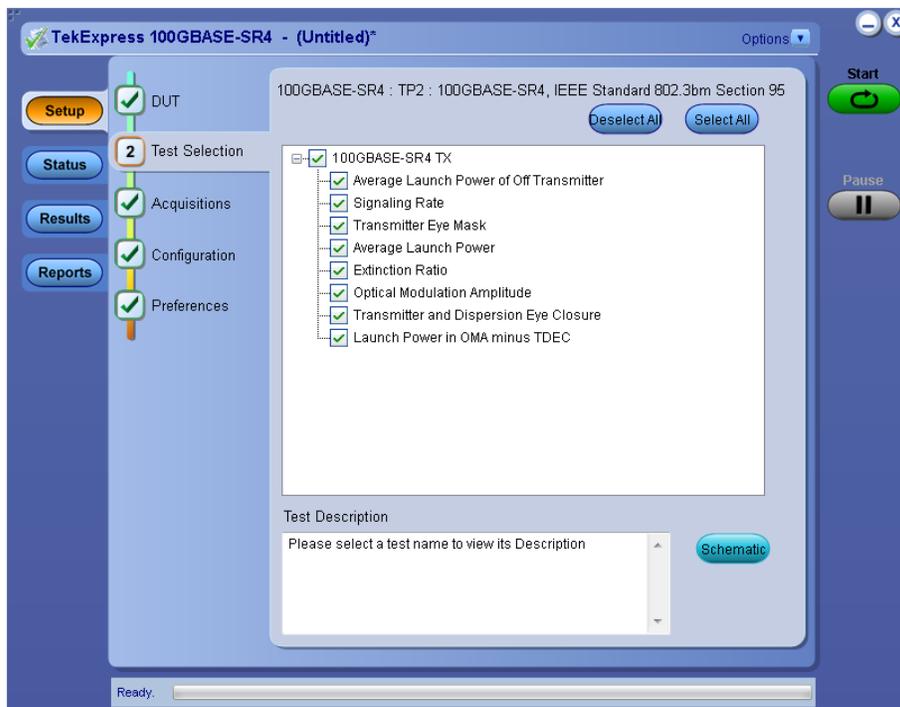
**NOTE.** The installed optical module sets the available wavelength, filter, and bandwidth selections.

You can set either the filter or bandwidth value, but not both.

See also. [Select a test](#)

### Select tests

Use the Test Selection tab to select the TekExpress 100GBASE-SR4 tests.



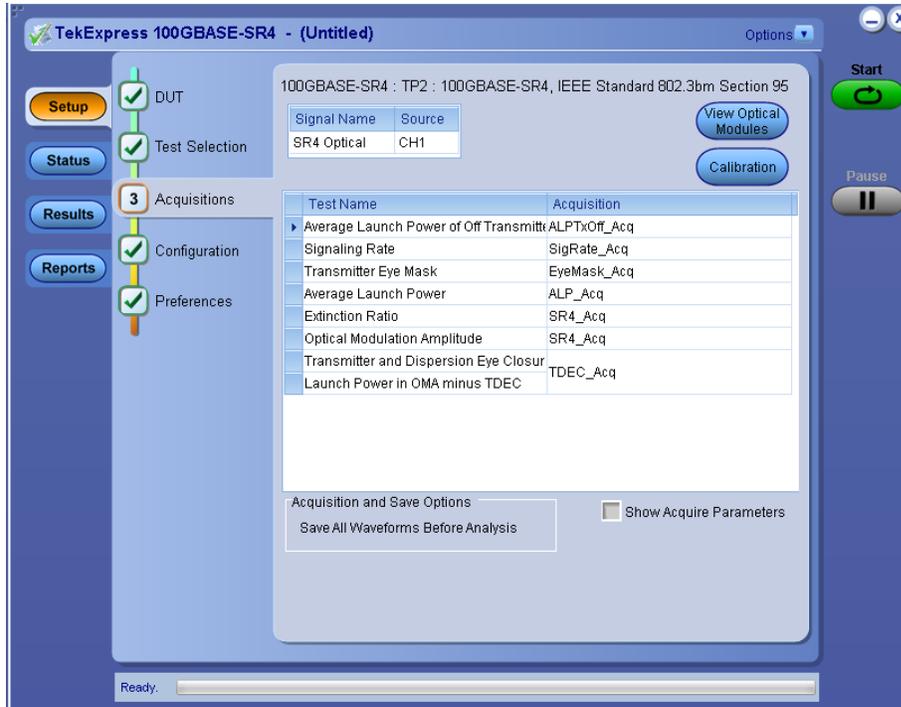
**Table 9: Test Selection tab settings**

Setting	Description
Tests	Click on a test to select or unselect. Highlight a test to show details in the Test Description pane.
Test Description	Shows a brief description of the highlighted test in the Test field.

See also. [Set acquisition tab parameters](#)

## Set acquisition tab parameters

Use the Acquisitions tab to view test acquisition parameters. The contents displayed on this tab depends on the DUT type and the tests selected.



**NOTE.** 100GBASE-SR4 acquires all waveforms needed by each test group before performing analysis.

**Table 10: Acquisitions tab settings**

Setting	Description
<b>View Optical Modules</b> button	Click to view the detected optical modules that are installed in the instrument.
<b>Calibration</b> button	Click to view the status of Oscilloscope calibration, External attenuation calibration, and Instrument noise calibration. Update these parameters by clicking the associated Refresh or Measure button. See <a href="#">Pre-measurement calibration</a> for details.
Show Acquire Parameters	Select to view the acquisition parameters.



TekExpress 100GBASE-SR4 saves all acquisition waveforms to files by default. Waveforms are saved in a unique folder for each session (a session starts when you click the Start button). The folder path is `x:\100GBASE-SR4\Untitled Session \<dutid>\<date>_<time>`. Images created for each analysis, CSV files with result values, reports, and other information specific to that particular execution are also saved in this folder.

Saving a session moves the session file contents from the Untitled Session folder to the specified folder name, and changes the session name to the specified name.

### Set configuration tab parameters

Use the Configuration tab to view the instruments detected (Global Settings).

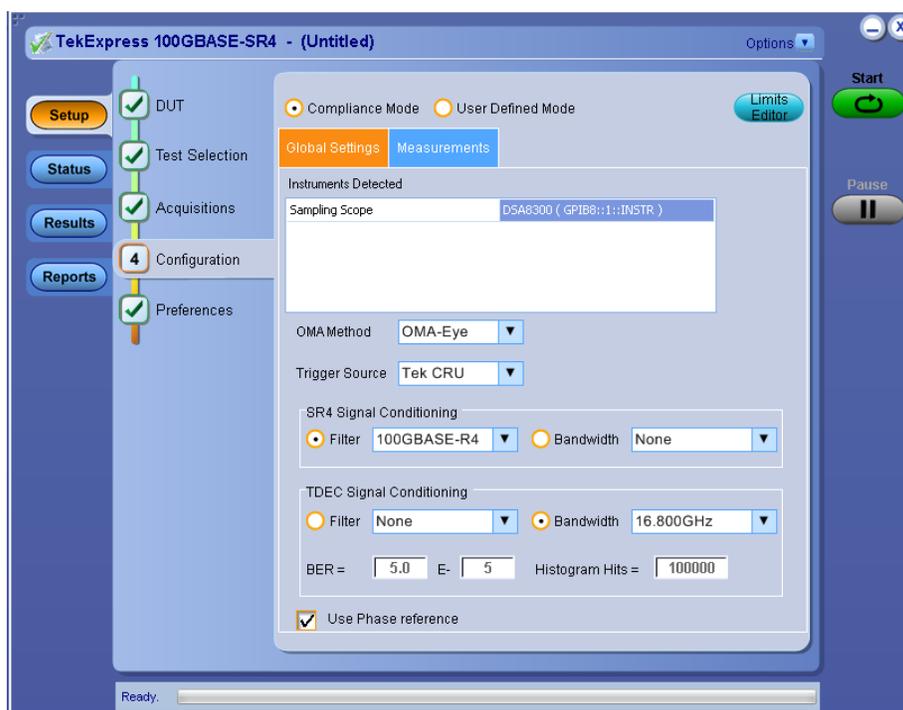


Figure 1: Configuration tab: Global Settings

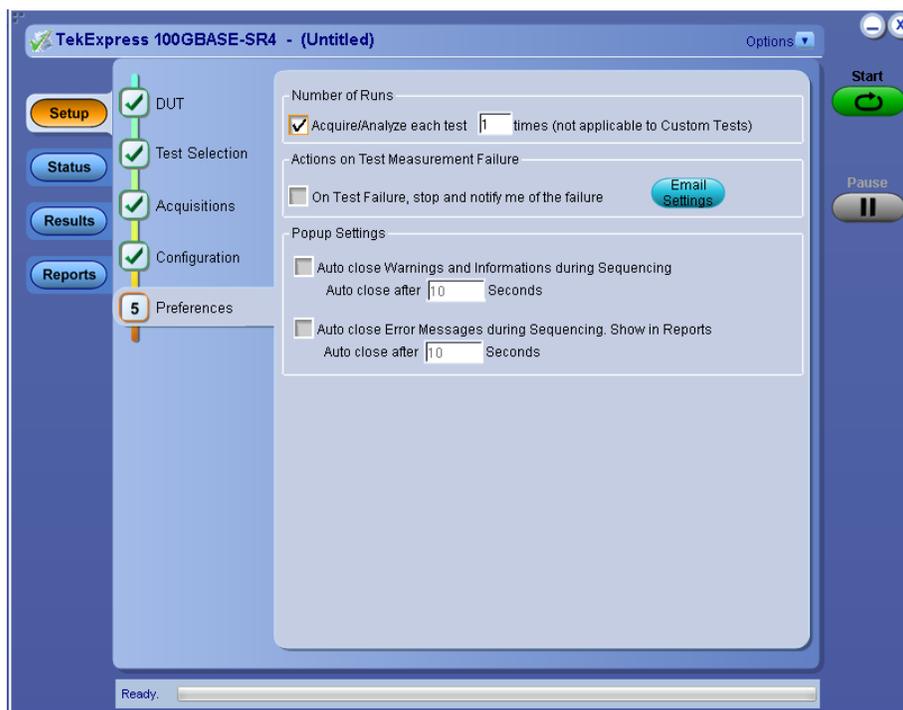
**Table 11: Configuration tab settings**

Setting		Description
Compliance Mode		Select to view compliance mode. By default Compliance Mode is selected.
User Defined Mode		Select to view user defined mode
<b>Global Settings</b>		
Instruments Detected		<p>Displays the instruments connected to this application. Click on the instrument name to open a list of available (detected) instruments.</p> <p>Select <b>Options &gt; Instrument Control Settings</b> and click Refresh to update the instrument list.</p> <hr/> <p><b>NOTE.</b> Verify that the <b>GPIB</b> search criteria (default setting) in the Instrument Control Settings is selected when using TekExpress 100GBASE-SR4 application.</p>
OMA Method		Select the OMA method from the drop-down list. By default, it is OMA-Eye.
Trigger Source		Select the trigger source from the drop-down list. By default, it is Tek CRU.
SR4 Signal Conditioning	Filter	<p>Select the filter from the drop-down list. If 80C15 module is present in main frame, then default value is a function of data rate.</p> <ul style="list-style-type: none"> <li>■ 25.78125 Gbps -- 100GBASE-R4</li> <li>■ 27.952 Gbps - OTU-4</li> <li>■ 28.05 Gbps - 32 GFC r0</li> </ul> <p>If 80C15 module is not present, then the default value will be None</p>
	Bandwidth	Select the bandwidth from the drop-down list.
TDEC Signal Conditioning	Filter	Select the filter from the drop-down list.
	Bandwidth	<p>Select the bandwidth from the drop-down list.</p> <p>If 80C15 is present in main frame, and if the module has TDEC filter (16.8 GHz), then the default will be 16.8 GHz.</p> <p>If 80C15 is not present in main frame then default value will be None</p>
	BER	Set the BER value. By default it is 5.0E-5.
	Histogram Hits	Set the Histogram Hits. By default it is 100000.

Setting		Description
Use Phase reference		Select to perform phase reference characterization.
<b>Measurements</b>		
Average launch Power of Off Transmitter		No configuration available for this measurement.
Signaling Rate		No configuration available for this measurement.
Transmitter Eye Mask	Mask	Mask file path
Average Launch Power	Method	Select the Method. By default it is Histogram
Extinction Ratio	Time Limit	Set the time limit. By default it is 24 hrs.
	Temperature Drift Tolerance	Set the temperature drift tolerance. By default it is 5°C
Optical Modulation Amplitude		No configuration available for this measurement.
Transmitter and Dispersion Eye Closure		No configuration available for this measurement.
Launch Power in OMA minus TDEC		No configuration available for this measurement.

### Set preferences tab parameters

Use the Preferences tab to set the application action on completion of a measurement.



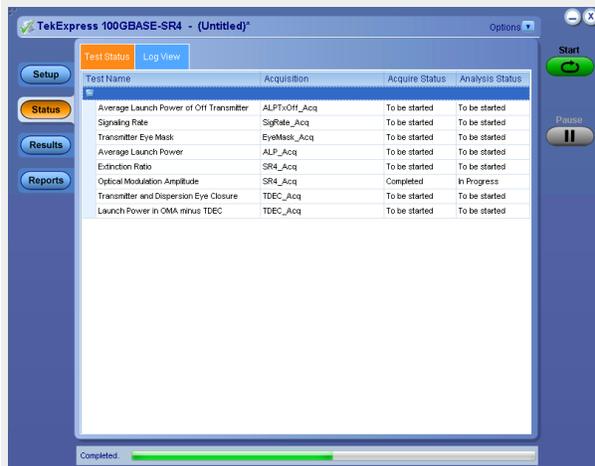
**Table 12: Preferences tab settings**

Setting	Description
<b>Number of Runs</b>	
Acquire/Analyze each test <no> times (not applicable to Custom Tests)	Select to repeat the test run by setting the number of times. By default, it is selected with 1 run.
<b>Actions on Test Measurement Failure</b>	
On Test Failure, stop and notify me of the failure	Select to stop the test run on Test Failure, and to get notified via email. By default, it is unselected. Click Email Settings to configure.
<b>Popup Settings</b>	
Auto close Warnings and Informations during Sequencing Auto close after <no> Seconds	Select to auto close warnings/informations during sequencing. Set the Auto close time. By default it is unselected.
Auto close Error Messages during Sequencing. Show in Reports Auto close after <no> Seconds	Select to auto close Error Messages during Sequencing. Set the Auto close time. By default it is unselected.

## Status panel overview

The Status button accesses the Test Status and Log View tabs, which provide 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.

### Test status view



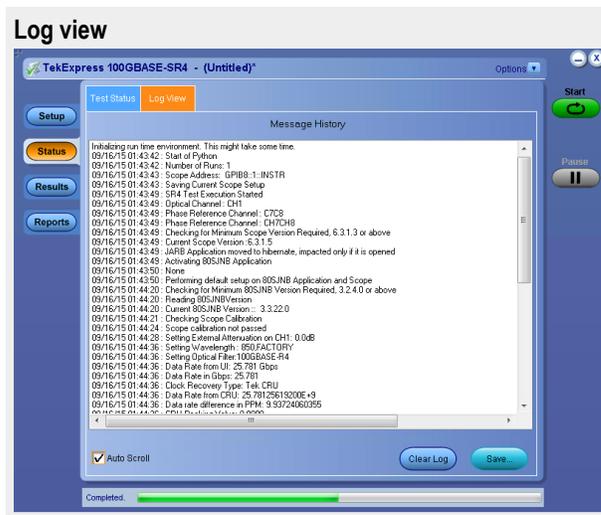


Table 13: Status panel Log View controls

Control	Description
Message History	Lists all executed test operations and timestamp information.
Auto Scroll	Enables automatic scrolling of the log view as information is added to the log during the test.
Clear Log	Clears all messages from the log view.
Save	Saves the log file to a text file. Use the standard Save File window to navigate to and specify the folder and file name to which to save the log text.

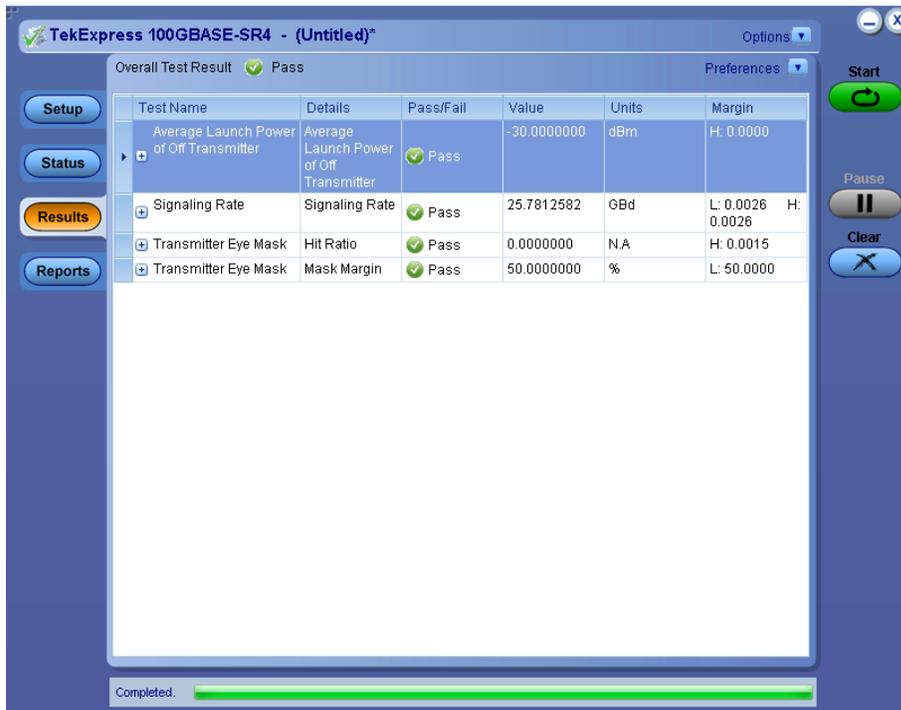
## See also

[Application panel overview](#)

## Results panel

### Results panel overview

When a test finishes, the application automatically opens the **Results** panel to display a summary of test results.



See also. [View a report](#)

[Application panels overview](#)

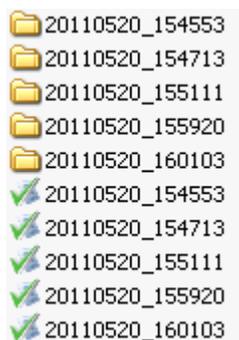
### View test-related files

Files related to tests are stored in My TekExpress\100GBASE-SR4\ . Each test setup in this folder has both a test setup *file* and a test setup *folder*, both with the test setup name.

The test setup file is preceded by the TekExpress icon and usually has no visible file name 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 also has a folder and file pair, both named for the test session using the naming convention (date)\_(time). Each session file is stored outside its matching session folder:



Each session folder contains image files of any plots generated from running the test session. If you selected to save all waveforms or ran tests using prerecorded waveform files, these are included here.

The first time you run a new, unsaved session, the session files are stored in the `Untitled Session` folder located at `.. \My TekExpress\100GBASE-SR4\`. When you name and save the session, the files are placed in a folder with the name that you specify. A copy of the test files stay in the `Untitled Session` folder until you run a new test or until you close the 100GBASE-SR4 application.

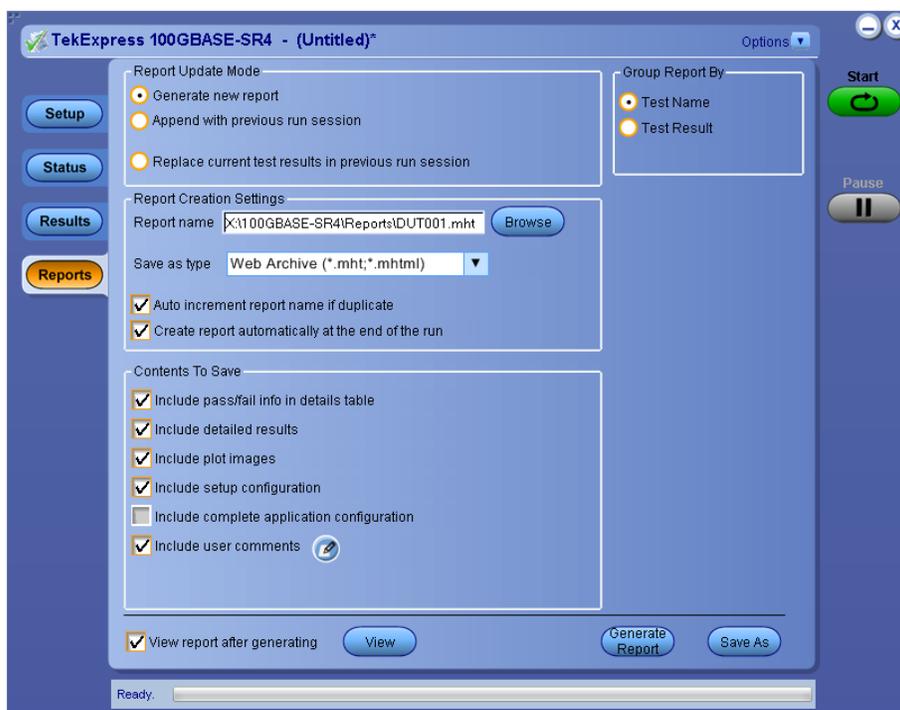
See also. [File name extensions](#)

[My TekExpress folder settings](#)

## Reports panel

### Reports panel overview

Use the Reports panel to browse for reports, name and save reports, select test content to include in reports, and select report viewing options.



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

See also. [Applications panel overview](#)

## Select report options

Click the **Reports** button and use the Reports panel controls to select which test result 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.

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 report options:

**Table 14: Report options**

Setting	Description
<b>Report Update Mode</b>	
Generate new report	Creates a new report. The report can be in either .mht or .pdf file formats.
Append with previous run session	Appends the latest test results to the end of the current test results report.
Replace current test in previous run session	Replaces the previous test results with the latest test results. Results from newly added tests are appended to the end of the report.
<b>Report Creation Settings</b>	
Report name	<p>Displays the name and location from which to open a 100GBASE-SR4 report. The default location is at <i>My TekExpress\100GBASE-SR4\Untitled Session</i>. 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"> <li>■ In the Report Path field, type over the current folder path and name.</li> <li>■ Double-click in the Report Path field and then make selections from the popup keyboard and click the <b>Enter</b> button.</li> </ul> <p>Be sure to include the entire folder path, the file name, and the file extension. For example: C:\Documents and Settings\your user name\My Documents\My TekExpress\100GBASE-SR4\DUT001.mht.</p> <p><b>NOTE.</b> You cannot set the file location using the Browse button.</p> <p>Open an existing report.</p> <p>Click <b>Browse</b>, locate and select the report file and then click <b>View</b> at the bottom of the panel.</p>

Setting	Description
Save as type	Saves a report in the specified file type, selected from the drop-down list.  <b>NOTE.</b> If you select a file type different from the default, be sure to change the report file name extension in the Report Name field to match.
Auto increment report name if duplicate	Sets the application to automatically increment the name of the report file if the application finds a file with the same name as the one being generated. For example: DUT001, DUT002, DUT003. This option is enabled by default.
Create report automatically at the end of the run	Creates report at the end of the run.
<b>Contents To Save</b>	
Include pass/fail info in details table	Includes pass/fail info in the details table of the report.
Include detailed results	Includes detailed results in the report.
Include plot images	Includes plot images in the report.
Include setup configuration	Sets the application to include hardware and software information in the summary box at the top of the report. Information includes: the oscilloscope model and serial number, the oscilloscope firmware version, and software versions for applications used in the measurements.
Include complete application configuration	Includes complete application configuration in the report.
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.
<b>Group Report By</b>	
Test Name	Select to group the tests in the report by test name.
Test Result	Select to group the tests in the report by test results
View report after generating	Automatically opens the report in a Web browser when the test completes. This option is selected by default.
View	Click to view the most current report.
Generate Report	Generates a new report based on the current analysis results.
Save As	Specify a name for the report.

## 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 to view a different test report, do the following:

1. Click the **Reports** button.
2. Click the **Browse** button and locate and select the report file to view.
3. In the Reports panel, click **View**.

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

## Report contents

A report shows detailed results and plots, as set in the Reports panel.

Tektronix TekExpress 100GBASE-SR4 Test Report – (TP2)			
<b>Setup information</b>			
DUT ID	DUT001	Scope Model Number	DSA8300
Date/Time	2015-09-07 01:13:52	Scope Serial Number	PQ10016
TekExpress Version	100GBASE-SR4: 0.0.1.123 Framework: 4.0.2.213	Scope F/W Version	6.3.1.5
Specification Version	IEEE 802.3bm - 2015	Optical Module Model Number	CH1 "80C15-P"
Compliance Mode	True	Optical Module Serial Number	CH1 "B010225"
Overall Test Result	Pass	PhaseRef Module Model Number	CH7CH8 "82A04B-60G"
Overall Execution Time	0:04:30	PhaseRef Module Serial Number	CH7CH8 "Q0005"
		Trigger Source	"CRUB171562", SW Ver: "4.0"
		80S/NB Version	N.A
		Data Rate	25.78126 Gbps
DUT Comment:	100GBASE-SR4 Tests		
<b>Test Name Summary Table</b>			
Average Launch Power of Off Transmitter	Pass		
Signaling Rate	Pass		
Transmitter Eye Mask	Pass		
Average Launch Power	Pass		
Extinction Ratio	Pass		
Optical Modulation Amplitude	Pass		
Transmitter and Dispersion Eye Closure	Pass		
Launch Power in OMA minus TDEC	Pass		

### Setup configuration information

The summary box at the beginning of the report lists setup configuration information. This information includes the oscilloscope model and serial number, optical module model and serial number, and software version numbers of all associated applications.

To exclude this information from a report, clear the **Include Setup Configuration** check box in the Reports panel before running the test.

### User comments

If you selected to include comments in the test report, any comments you added in the DUT tab are shown at the top of the report.

**See also.** [Results panel overview](#)

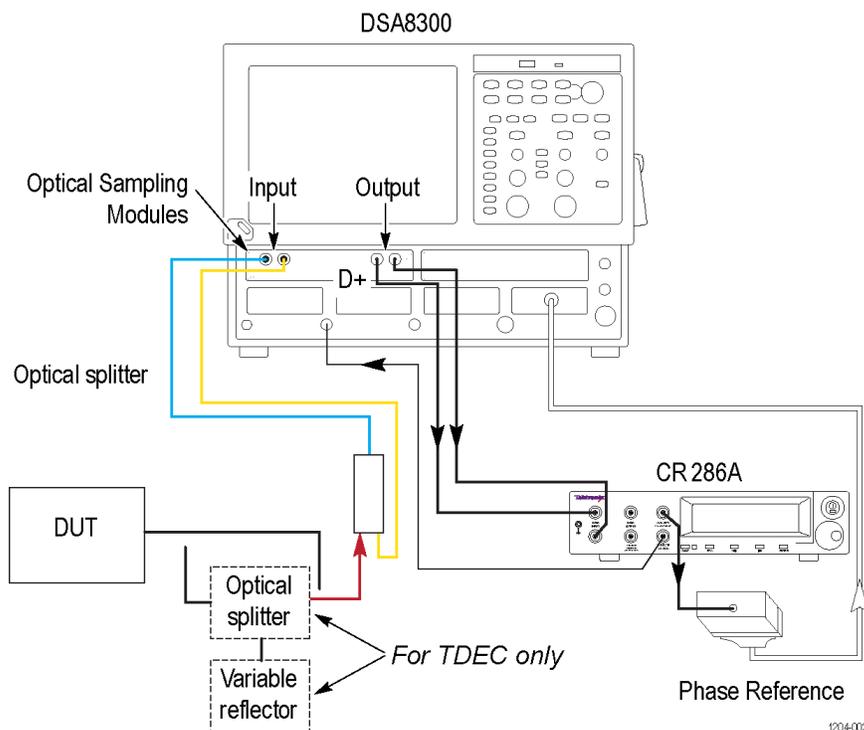
[View test-related files](#)

# Pre-measurement calibration

Perform the Pre-measurement calibration before you run a measurement, and when you make changes to the setup configuration such as moving any sampling modules, cables, or connectors.

## Equipment setup

Click **Setup** > **Test Selection** > **Schematic** to view the equipment setup diagram(s).



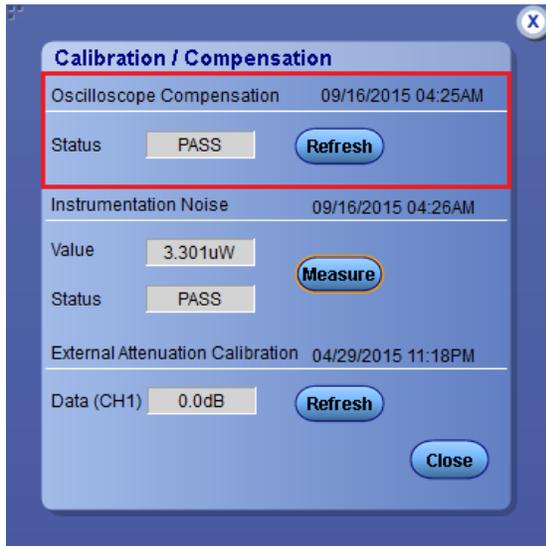
### See also

[Minimum system requirements](#)

## Oscilloscope compensation

Use the following procedure to check oscilloscope calibration status:

1. Select **TekExpress 100GBASE-SR4 > Setup > Acquisition panel > Calibration** to open the calibration dialog box.
2. Click **Refresh** (in the Oscilloscope Calibration area).



**NOTE.** It is recommended to perform Scope Compensation in addition after 20 minutes of warm up. Scope compensation can be accessed from the Oscilloscope main menu, **Utilities > Instrument Compensation**. Click **Help** in the compensation window for further details.

## Instrument noise

The following procedure is used by the 100GBASE-SR4 application to measure Instrument noise calibration:

1. Disconnect all signals connected to the sampling oscilloscope
2. Select **Setup > Vert > waveform C1 to On**
3. Define **MATH1** as Ch1, and switch on MATH1
4. Set the Trigger Source to **Free Run**
5. Select measurement **Setup > Meas > Meas 1 > Pulse Amplitude: AC RMS**
6. Set **Setup > Meas > Source: MATH1**
7. Set WaveformdB source as MATH1
8. Enable and switch on the display of WaveformdB
9. Query the result of measurement1 (AC RMS)

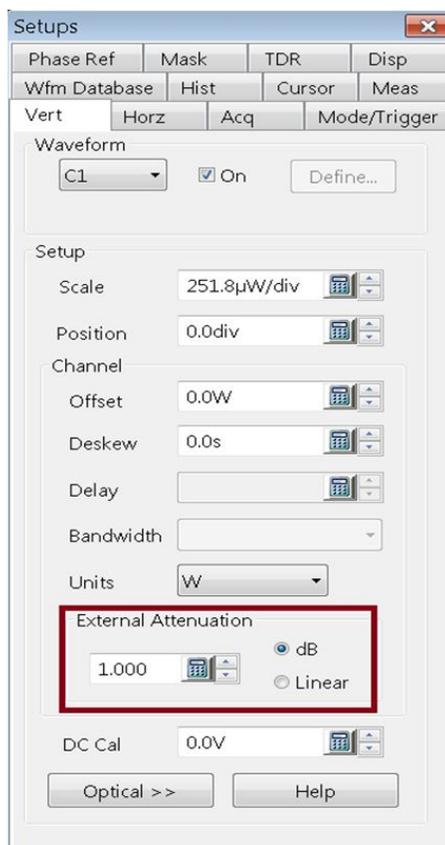
**NOTE.** Measured noise limit is a function of optical settings (Bandwidth and Filter).

If the noise level measurement is not within the limits, perform an oscilloscope compensation and then perform the instrument noise measurement again. If the measured noise level is still outside of the above limits, please [contact Tektronix customer support](#).

## External attenuation calibration

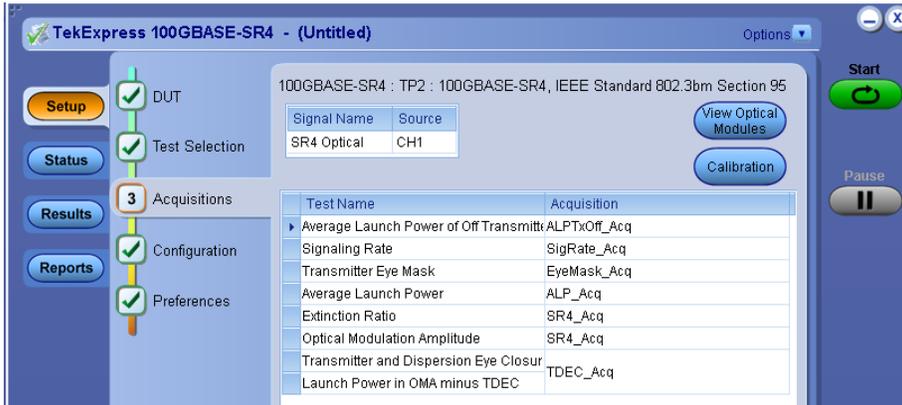
Follow the steps to set the external attenuation:

1. In DSA8300, set the optical source as **Ch1**
2. Enter the External Attenuation value for the scope as shown in the following image.

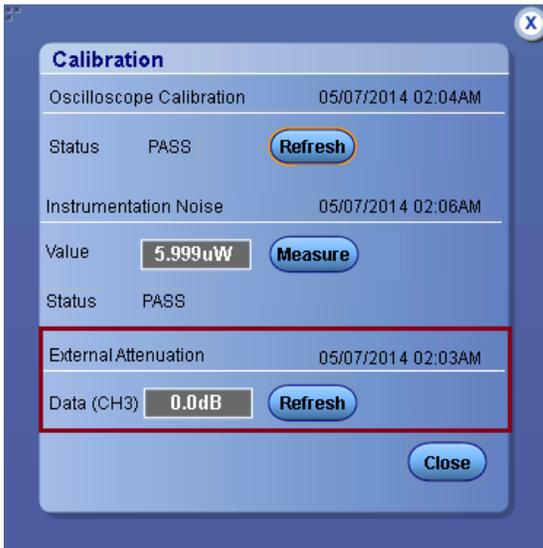


3. Select **Ch1** from the **TekExpress 100GBASE-SR4 > Setup > DUT > Source**

- Click **TekExpress 100GBASE-SR4 > Setup > Acquisition > Calibration** to open the calibration dialog box



- Click **Refresh** (in External Attenuation area) and check the value



- Repeat steps 1 to 5 by selecting **Ch3** and check the value

## Running tests

After selecting and configuring tests, click **Start** to run the tests. While tests are running, you cannot access the Setup or Reports panels. To monitor the test progress, switch between the Status panel and the Results panel.

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 **Alt + Tab** key combination. To keep the TekExpress 100GBASE-SR4 application on top, select **Keep On Top** from the TekExpress Options menu.

The application displays report when the tests execution is complete.

---

**NOTE.** *The 100GBASE-SR4 software stores the DSA8300 oscilloscope setup before test execution and restores the oscilloscope to the same state after test execution.*

---

### Prerun checklist

1. Map [My TekExpress folder to drive X:](#) and set [My TekExpress folder permissions](#) before running tests for the first time.
2. Make sure that the instruments are warmed up (approximately 20 minutes) and stabilized.
3. Perform compensation: In the oscilloscope main menu, select **Utilities > Instrument Compensation**. Click **Help** in the compensation window for steps to perform instrument compensation.



---

# Saving and recalling test setup

## Test setup files overview

Saved test setup information (such as the selected oscilloscope, general parameters, acquisition parameters, measurement limits, waveforms (if applicable), and other configuration settings) is saved under the setup name at **X:\100GBASE-SR4**.

Use test setups to:

- Run a new session, acquiring live waveforms, using a saved test configuration.
- Create a new test setup based on an existing one.
- View all the information associated with a saved test, including the log file, the history of the test status as it executed, and the results summary.
- Run a saved test using saved waveforms.

### See also

[Save a test setup](#)

[Open \(load\) a saved test setup](#)

## Save a test setup

You can save a test setup before or after running a test. You can create a test setup from [already opened test setup](#), or using [default test setup](#). When you select the default test setup, the parameters are set to the application's default value.

Select **Options > Save Test Setup** to save the opened setup.

Select **Options > Save Test Setup As** to save the setup with different name.

## Open (load) a saved test setup

To Open (load) a saved test setup, do the following:

1. Select **Options > Open Test Setup**.
2. Select the setup from the list and click **Open**. Setup files are located at **X:\100GBASE-SR4**.

### See also

[About test setups](#)

[Create a test setup using an existing one](#)

[Create a test setup from default settings](#)

## Create a test setup from default settings

To create a test setup using default settings, follow the steps:

1. Select **Options > Default Test Setup**. For default test setup, the parameters are set to the application's default value.
2. Click application [Setup](#) and set the parameters
3. Click application [Reports](#) and set the report options
4. Optional: Click **Start** to run the test and verify that it runs correctly and captures the specified test information and reports. If it does not, then edit the parameters and repeat this step until the test runs to your satisfaction
5. Select **Options > Save Test Setup**. Enter the file name and click Save. The application saves the file to X:\100GBASE-SR4\*session\_name*

## Create a test setup using an existing one

To create a test setup using an existing one, follow the steps:

1. Select **Options > Open Test Setup**
2. Select a setup from the list and then click **Open**
3. Click application [Setup](#) and modify the parameters
4. Click application [Reports](#) and modify the report options
5. Select **Options > Save Test Setup As**
6. Enter test setup name, and click **Save**

---

# 100GBASE-SR4 Transmitter compliance measurements

## Signaling Rate

This test verifies that the signaling speed of the Device Under Test (DUT) is within conformance limit, specified in Table 95-6 of 802.3bm - IEEE Standard for Ethernet.

### Required test equipment

[Click here](#) to check the Instruments and accessories required.

Connect the equipment as shown in the [equipment setup diagram](#).

### Measurement algorithm

1. Feed the External Clock Recovery Unit (CRU) with data signal, configure and lock the clock recovery unit. Connect the external CRU using USB interface to the sampling scope, and launch the 100GBASE-SR4 application.
2. Data rate of locked CRU is used as result for this measurement. If CRU is not locked, then error is reported. See [Clock Recovery Unit](#) for locking mechanism.
3. If Tektronix external clock recovery unit is not available in test setup, then pop-up is launched to enter the data rate. The entered data rate should be the data rate to which the non-Tektronix CRU is locked.
4. The data rate limits are:
  - Low Limit: 25.7786719 Gb/s
  - High Limit: 25.78382813 Gb/s

## Average Launch Power

This test verifies that the Average Launch Power of the Device Under Test (DUT) is within conformance limit, as specified in Table 95-6 of 802.3bm - IEEE Standard for Ethernet.

### Required test equipment

[Click here](#) to check the Instruments and accessories required.

---

**NOTE.** This measurement requires any commercial 850nm Power Meter.

---

Connect the equipment as shown in the [equipment setup diagram](#).

### Measurement algorithm

Average launch power is defined as the power with which the optical signal is launched at the transmitter end. It is the function of the brightness of the optical source. Average launch power is expressed in dBm with 1mW acting as the reference level. The Average Launch Power can be computed by the following methods:

#### Histogram based computation

**Pre-requisite:** Clock signal synchronous with data has to be provided as clock pre-scale/trigger input to the sampling scope.

1. Switch on the channel which is connected to 100GBASE-SR4 optical signal
2. Configure optical settings of the channel – wavelength, filter and bandwidth
3. Set the bit rate in horizontal sub system to user configured bit rate in GUI

4. Run **AutoSet** to set optimal vertical, horizontal settings. It also provides an eye occupying six vertical divisions and one bit (two eye crossings) displayed over about 7.5 horizontal divisions
5. Check On and Display check boxes of "WfmDB1". Set source of WfmDB1 as the optical source selected by user
6. Run **NRZ Timing - Bit Time** measurement with following settings:
  - a. Select Source as the optical source selected by user
  - b. Check **Use Wfm Database**
  - c. Select Signal Type as **NRZ**
7. Run **NRZ Timing - Crossing Time** measurement with the following settings:
  - a. Select Source as the optical source selected by user
  - b. Check **Use Wfm Database**
  - c. Select Signal Type as **NRZ**
8. Run **NRZ Amplitude - Max** measurement with the following settings:
  - a. Select Source as the optical source selected by user
  - b. Check **Use Wfm Database**
  - c. Select Signal Type as **NRZ**
9. Run **NRZ Amplitude - Min** measurement with the following settings:
  - a. Select Source as the optical source selected by user
  - b. Check **Use Wfm Database**
  - c. Select Signal Type as **NRZ**
10. In Histogram setup,
  - Select source of histogram as the optical source selected by the user
  - Enable **Use Wfm Database**. Switch on **Histogram** of Linear type in display options
  - Enable vertical histogram with absolute co-ordinates as:
    - Top = Result of Max measurement + (vertical scale / 2)
    - Bottom = Result of Min measurement – (vertical scale / 2)
    - Left = Result of Cross Time Measurement
    - Right = (Result of Cross Time measurement + Result of bit time measurement)
11. Mean of the histogram is taken as Average Launch Power in Watts
12. Convert measured average launch power to dBm scale Average launch power in dBm =  $10 \cdot \log_{10}$  (average launch power in Watts/1mWatts)
13. Average Launch Power in dBm is compared with limits to indicate pass/fail

#### **Optical Power Meter based computation**

1. Connect the optical power meter to the DUT at TP2 (end of optical patch cord)
2. Determine the reading of the optical power meter
3. Make sure to apply any external amplifications/attenuations on the optical power meter reading
4. Convert the result of step 3 into dBm

5. Enter the average optical power value (dBm) in the pop-up displayed by 100GBASE-SR4 application
6. The Average Launch Power (dBm) entered is compared with limits to indicate pass/fail

Limits:

- Low Limit: -8.4 dBm
- High Limit: 2.4 dBm

## Optical Modulation Amplitude

This test verifies that the Optical Modulation Amplitude (OMA) of the Device Under Test (DUT) is within conformance limit, as specified in Table 95-6 of 802.3bm - IEEE Standard for Ethernet.

### Required test equipment

[Click here](#) to check the Instruments and accessories required.

Connect the equipment as shown in the [equipment setup diagram](#).

### Measurement algorithm

If Average Launch Power is measured externally, then measure Extinction Ratio (ER) and compute OMA as given below:

1.  $OMA = \text{Average Launch Power} * \frac{(\text{Extinction Ratio} - 1)}{(\text{Extinction Ratio} + 1)}$ . See [Extinction Ratio](#) measuring procedure
2. OMA is compared with limits for pass/fail
3. Measured OMA is given in report comments

If Average Launch Power is measured internally, then OMA is computed using the method configured by user in GUI.

The Optical Modulation Amplitude can be computed by the following methods:

#### OMA-Eye

**Pre-requisite:** Clock signal synchronous with data has to be provided as clock pre-scale/trigger input to sampling scope.

1. Switch on the channel which is connected to 100GBASE-SR4 optical signal
2. Configure optical settings of the channel – wavelength, filter, and bandwidth
3. Set the bit rate in horizontal sub system to user configured bit rate in GUI
4. Run **AutoSet** to set optimal vertical, horizontal settings. It also provides an eye occupying six vertical divisions and one bit (two eye crossings) displayed over about 7.5 horizontal divisions
5. Check On and Display check boxes of "WfmDB1". Set source of WfmDB1 as the optical source selected by user
6. Run **NRZ Timing - OMA** measurement with following settings:
  - a. Select Source as the optical source selected by user
  - b. Check **Use Wfm Database**
  - c. Select Signal Type as **NRZ**
7. **NRZ Amplitude – OMA** results in watts is converted into dBm and then compared with limits for pass/fail

#### OMA-Pattern

**Supported patterns:** PRBS7, PRBS9, PRBS11, PRBS15

**Pre-requisite:** Clock signal synchronous with data has to be provided as clock pre-scale/trigger input to sampling scope.

1. Export the waveform using 80SJNB
2. Determine the longest duration of ones(positive pulse)
3. Collect the samples present in middle 20% of longest positive pulse
4. Determine the average value of amplitude of the collected samples. This gives high level
5. Determine the longest duration of zeros (negative pulse)
6. Collect samples present in middle 20% of longest negative pulse
7. Determine the average value of amplitude of collected samples. This gives low level
8. OMA = (High level – low level)

### Pattern type for positive and negative pulse

**Table 15: Long One and Zero patterns (PRBS) for OMA**

Pattern type	Pattern for longest duration of ones (positive pulse)	Pattern for longest duration of zeros (negative pulse)
PRBS7	11111110	0000001
PRBS9	1111111110	000001111
PRBS11	111111111110	00000000011
PRBS15	1111111111111110	000000000000001

If signal is inverted, then 100GBASE-SR4 application will search for inverted patterns (1 maps to 0 and 0 maps to 1). The positive and negative pulses will get interchanged and high low levels is also interchanged.

Limits:

- Low Limit: -6.4 dBm
- High Limit: 3 dBm

## Transmitter and Dispersion Eye Closure

This test verifies that the Transmitter and Dispersion Eye Closure (TDEC) of the Device Under Test (DUT) is within the conformance limit, as specified in Table 95-6 of 802.3bm - IEEE Standard for Ethernet

### Required test equipment

[Click here](#) to check the Instruments and accessories required.

Connect the equipment as shown in the [equipment setup diagram](#).

### Measurement algorithm

This measurement has two steps:

1. Pre-processing: Setting up an eye, histogram boxes and exporting histogram data
2. Post-processing: Sigma and/or TDEC computation

**Pre-processing:** Setting up an eye, histogram boxes and exporting histogram data

1. Switch on the channel which is connected to 100GBASE-SR4 optical signal
2. Configure optical settings of the channel – wavelength, filter and bandwidth
3. Set the bit rate in horizontal sub system to user configured bit rate in GUI
4. Run **AutoSet** to set optimal vertical, horizontal settings. It also provides an eye occupying six vertical divisions and one bit (two eye crossings) displayed over about 7.5 horizontal divisions
5. Check On and Display check boxes of “WfmDB1”. Set source of WfmDB1 as the optical source selected by user
6. If phase reference module is present, dial-in the data rate and perform phase reference characterization
7. Run **NRZ Timing - Bit Time** measurement with following settings:
  - a. Source as optical source selected by user
  - b. Check **Use Wfm Database**
  - c. Signal Type as **NRZ**
8. Run **NRZ Timing - Crossing Time** measurement with the following settings:
  - a. Source as optical source selected by user
  - b. Check **Use Wfm Database**
  - c. Signal Type as **NRZ**
9. Run **NRZ Amplitude - Max** measurement with the following settings:
  - a. Source as optical source selected by user
  - b. Check **Use Wfm Database**
  - c. Signal Type as **NRZ**
10. Run **NRZ Amplitude - Min** measurement with the following settings:
  - a. Source as optical source selected by user
  - b. Check **Use Wfm Database**
  - c. Signal Type as **NRZ**
11. In Histogram setup,
  - a. Select source of histogram as the optical source selected by the user
  - b. Enable **Use Wfm Database**. Switch on **Histogram** of Linear type in display options
  - c. Enable vertical histogram with absolute co-ordinates as:
    - Top = Result of Max measurement + (vertical scale / 2)
    - Bottom = Result of Min measurement – (vertical scale / 2)
    - Left = Result of Cross Time Measurement
    - Right = (Result of Cross Time measurement + Result of bit time measurement)
12. Mean of the histogram is taken as Average Launch Power in Watts
13. Determine the histogram box co-ordinates for early top histogram:
  - Early top histogram Left = crossing time + bit time \* (Early histogram box position – (histogram width/2))
  - Early top histogram Right = crossing time + bit time \* (Early histogram box position + (histogram width/2))
  - Early top histogram Base = AOP

- Early top histogram Top = Maximum amplitude value + (vertical scale/2)
  - Set histogram configuration same as in Step 11.a and 11.b
14. Set acquisition configuration:
- Stop After Mode as **Condition**
  - Stop After Condition as **HISTOGRAM HITS**
  - Stop After Count as number of histogram hits configured by user in GUI
  - Clear the screen and switch on Acquire state. Once acquisition is complete, export the early top histogram data to a text file.
15. Determine the histogram box co-ordinates for early bottom histogram:
- Early bottom histogram Left = crossing time + bit time \* (Early histogram box position – (histogram width/2))
  - Early bottom histogram Right = crossing time + bit time \* (Early histogram box position + (histogram width/2))
  - Early bottom histogram Base = AOP
  - Early bottom histogram Top = Maximum amplitude value + (vertical scale/2)
- Do not re-run the acquisition. Export the early bottom histogram data to a text file.
16. Determine the histogram box co-ordinates for late top histogram:
- Late top histogram Left = crossing time + bit time \* (Late histogram box position – (histogram width/2))
  - Late top histogram Right = crossing time + bit time \* (Late histogram box position + (histogram width/2))
  - Late top histogram Base = AOP
  - Late top histogram Top = Maximum amplitude value + (vertical scale/2)
- Do not re-run the acquisition. Export the Late bottom histogram data to a text file.
17. Determine the histogram box co-ordinates for late bottom histogram:
- Late bottom histogram Left = crossing time + bit time \* (Late histogram box position – (histogram width/2))
  - Late bottom histogram Right = crossing time + bit time \* (Late histogram box position + (histogram width/2))
  - Late bottom histogram Base = AOP
  - Late bottom histogram Top = Maximum amplitude value + (vertical scale/2)
- Do not re-run the acquisition. Export the Late bottom histogram data to a text file.
18. Determine OMA. See [Optical Modulation Amplitude](#) for the procedure.

**Post-processing:**

- Sigma computation
  1. Create histogram Y-axis for both top histograms and bottom histograms (Y-axis does not change with reference to early or late):
    - Dumped/exported histogram will have 400 points corresponding to the rows of waveform database.
    - Histogram Width = (Top Screen Voltage – Bottom Screen Voltage) / 400
    - Y-axis of full screen is computed as Top screen voltage to bottom screen voltage with step size as histogram width

- Y-axis corresponding to top histogram is computed as Y-axis points between early top histogram top and early top histogram base
  - Y-axis corresponding to bottom histogram is computed as Y-axis points between early bottom histogram top and early bottom histogram base
2. Normalize histograms for all four so that, the sum of the area of the histogram of bins is 1
  3. Compute earlySigma value using normalized early top histogram and normalized early bottom histogram

$$\text{top\_scalingFactor}(i) = Q([\text{tophistogramYaxis}(i) - (\text{AOP})]/\text{earlySigma})$$

$$\text{bottom\_scalingFactor}(i) = Q([\text{AOP}] - \text{bottomhistogramYaxis}(i)]/\text{earlySigma})$$

where i = 0 to number of bins

$$\text{BER} = \frac{\sum_i[\text{normalizedTophistogram}(i) \cdot \text{top\_scalingFactor}(i)]}{\sum_i[\text{normalizedTophistogram}(i)]} + \frac{\sum_i[\text{normalizedbottomhistogram}(i) \cdot \text{bottom\_scalingFactor}(i)]}{\sum_i[\text{normalizedbottomhistogram}(i)]}$$

Adjust earlySigma until absolute difference between BER and target BER is less than 1e-20

4. Repeat step 3 for lateSigma value computation. Use normalized late top histogram and normalized late bottom histogram
5. Determine the larger sigma value, Sigma Value = Maximum (earlySigma, lateSigma)
6. Compute R using scope noise, M1 and M2

$$R = (1 - M1) * \sqrt{\text{sigmaValue}^2 + \text{scopeNoise}^2 - M2^2}$$

Where M1 = 0.04

M2 = 0.0175 \* Pav

M1 and M2 account for mode partition noise and modal noise that could be added by optical channel.

- TDEC computation

1. 
$$\text{TDEC} = 10 * \log_{10} \left( \frac{\text{OMA}}{(2 * R * (\sqrt{2} * \text{erfcinv}(\text{targetBER} * 2)))} \right)$$

TDEC is computed using the formula,

Computed TDEC is compared with limits to indicate pass/fail

Limits:

- Low Limit: NA
- High Limit: 4.3 dB

## Launch Power in OMA minus TDEC

This test verifies that the Launch Power in OMA minus TDEC of the Device Under Test (DUT) is within conformance limit, as specified in Table 95-6 of 802.3bm - IEEE Standard for Ethernet.

### Required test equipment

[Click here](#) to check the Instruments and accessories required.

Connect the equipment as shown in the [equipment setup diagram](#).

### Measurement algorithm

Launch power in OMA minus TDEC (dBm) = OMA measurement (dBm) – Result of TDEC measurement (dB)

Launch power in OMA minus TDEC (dBm) is compared with limits to indicate pass/fail

Limits:

- Low Limit: -7.3 dBm
- High Limit: NA

## Average Launch Power of Off Transmitter

This test verifies that the Average Launch Power of Off Transmitter of the Device Under Test (DUT) is within conformance limit, as specified in Table 95-6 of 802.3bm - IEEE Standard for Ethernet.

### Required test equipment

[Click here](#) to check the Instruments and accessories required.

Connect the equipment as shown in the [equipment setup diagram](#).

### Measurement algorithm

Average Launch Power with optical transmitter off can be measured only using external optical power meter. The 100GBASE-SR4 application will provide the connection diagram. Setup the device, and get the result from power meter and convert to dBm scale. Enter the result in the pop-up given by the application and compare to indicate pass/fail.

Limits:

- Low Limit: NA
- High Limit: -30 dBm

## Extinction Ratio

This test verifies that the Extinction Ratio of the Device Under Test (DUT) is within conformance limit, as specified in Table 95-6 of 802.3bm - IEEE Standard for Ethernet.

### Required test equipment

[Click here](#) to check the Instruments and accessories required.

Connect the equipment as shown in the [equipment setup diagram](#).

### Measurement algorithm

Extinction Ratio (ER) is defined as the ratio of average power levels of logic 1 level (High) to the logic 0 level (low) of an optical NRZ signal.

Extinction Ratio =  $P1 / P0$  where

P1 average power level of logic 1 level and

P0 average power level of logic 0 level

Extinction Ratio measurement accuracy will be increased if dark level compensation is done as pre-requisite. 100GBASE-SR4 application will perform dark level compensation at the start of execution of application. Decision of performing dark level compensation will be based on temperature drift limit and time limit set by user in GUI.

100GBASE-SR4 application shall execute "Extinction Ratio (dB) – Calibrated" measurement. If this measurement gives error, then the application will execute "Extinction Ratio (dB)" measurement.

1. Switch on the channel which is connected to 100GBASE-SR4 optical signal
2. Configure optical settings of the channel – wavelength, filter and bandwidth
3. Set the bit rate in horizontal sub system to user configured bit rate in GUI
4. Run **AutoSet** to set optimal vertical, horizontal settings. It also provides an eye occupying six vertical divisions and one bit (two eye crossings) displayed over about 7.5 horizontal divisions
5. Check On and Display check boxes of "WfmDB1". Set source of WfmDB1 as the optical source selected by user
6. Run **NRZ Amplitude - Extinction Ratio (dB) - Calibrated** measurement with following settings:
  - a. Select Source as the optical source selected by user
  - b. Check **Use Wfm Database**
  - c. Select Signal Type as **NRZ**
7. If NRZ Amplitude – Extinction Ratio (dB) – Calibrated measurement gives an error, then change the measurement to NRZ Amplitude – Extinction Ratio (dB)
8. Extinction Ratio result is compared with limits to indicate pass/fail

Limits:

- Low Limit: 2 dBm
- High Limit: NA

## Transmitter Eye Mask

This test verifies that the Transmitter Eye Mask of the Device Under Test (DUT) is within conformance limit, as specified in Table 95-6 of 802.3bm - IEEE Standard for Ethernet.

### Required test equipment

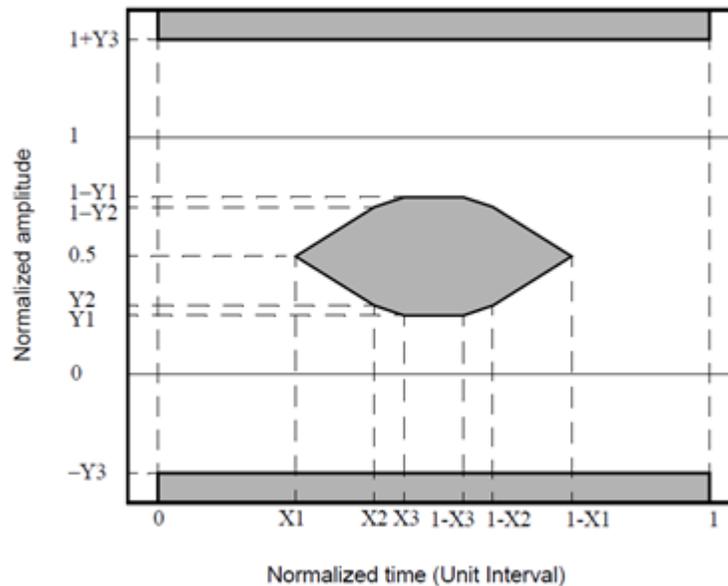
[Click here](#) to check the Instruments and accessories required.

Connect the equipment as shown in the [equipment setup diagram](#).

### Measurement algorithm

**Table 16: 100GBASE-SR4 mask co-ordinates**

Co-ordinates	Normalized values
X1	0.3 UI
X2	0.38 UI
X3	0.45 UI
Y1	0.35 normalized amplitude
Y2	0.41 normalized amplitude
Y3	0.5 normalized amplitude



**Figure 2: 100GBASE-SR4 Eye Mask**

Both amplitude (y-axis) and unit interval (x-axis) are normalized. Normalized levels of 0 and 1 represent logic zero and logic one respectively. These are defined by the means of the lower and upper halves of the central 0.2UI of the eye.

100GBASE-SR4 mask is available at C:\Users\Public\Documents\Tektronix\Masks\IEEE\Ethernet 802.3\100G\_40G\LAN\100GBASE\_SR4Draft3p3\100GbE\_SR4TXc33.m8k

1. Switch on the channel which is connected to 100GBASE-SR4 optical signal
2. This test is performed with CRU high frequency corner bandwidth as 10 MHz

3. Configure optical settings of the channel – Wavelength, filter and bandwidth. In this test eye is measured with respect to mask specified by co-ordinates given in table [Table 16: 100GBASE-SR4 mask co-ordinates](#) on page 48 using a receiver with fourth order Bessel Thomson filter with reference frequency as 19.34 GHz.
4. Set the bit rate in horizontal sub system to user configured bit rate in GUI
5. Run **AutoSet** to set optimal vertical, horizontal settings. It also provides an eye occupying six vertical divisions and one bit (two eye crossings) displayed over about 7.5 horizontal divisions
6. Check On and Display check boxes of “WfmDB1”. Set source of WfmDB1 as the optical source selected by user
7. Import custom mask
8. In Mask sub system
  - Set the mask source as the channel corresponding to the optical module to which 100GBASE-SR4 optical signal is connected
  - Select **Use Wfm Database**
  - Set communication standard as **Custom**
  - Do an **Autoset Data to Mask**
9. In acquisition sub system
  - Set acquisition mode to **Sample**
  - Set stop after to **Condition** and set condition as **Mask UI Samples**. Set the mask UI samples as (1/hit ratio high limit)\*5
10. Once the acquisition is stopped, query the hit ratio
11. Hit ratio is compared with limits to indicate pass/fail of the test.

Limits:

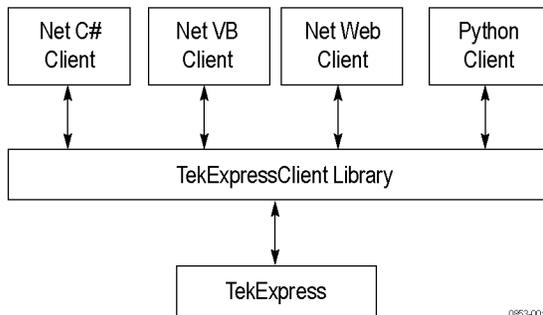
- Low Limit: NA
- High Limit: 1.5E-03



# TekExpress programmatic interface

## About the programmatic interface

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



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

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

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

### See also

[Requirements for developing TekExpress client](#)

## To enable remote access

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

1. Access the Windows Control Panel and open the Windows Firewall tool (**Start > Control Panel > All Control Panel Items > Windows Firewall**).
2. Click **Advance Settings > Inbound Rules**.

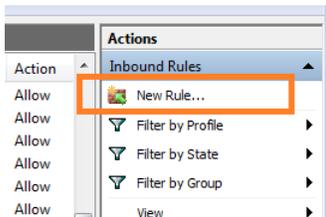
3. Scroll through the **Inbound Rules** list to see if the following items (or with a similar name) are shown:
  - TekExpress 100GBASE-SR4
  - TekExpress

Inbound Rules			
Name	Group	Profile	Enabled
✓ SCCMRT_TCP_2704		Domain	Yes
✓ secondary HTTP Port		All	Yes
✓ Status Port		All	Yes
✓ TekExpress		All	Yes
✓ TekExpress 100GBASE-SR4		All	Yes
✓ TekExpress CEI-VSR Application		All	Yes
✓ TekExpressSocketClient		All	Yes
✓ TekScope		All	Yes

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

### Run the New Inbound Rule Wizard

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



2. Verify that **Program** is selected in the Rule Type panel and click **Next**.
3. Click **Browse** in the Program panel and navigate to and select one of the following TekExpress applications (depending on the one for which you need to create a rule):
  - TekExpress 100GBASE-SR4.exe
  - TekExpress.exe

---

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

---

4. Click **Next**.
5. Verify that **Allow the connection** is selected in the Action panel and click **Next**.
6. Verify that all fields are selected (**Domain**, **Private**, and **Public**) in the Profile panel and click **Next**.

7. Use the fields in the Name panel to enter a name and optional description for the rule. For example, a name for the TekExpress 100GBASE-SR4 application could be **TekExpress 100GBASE-SR4 Application**. Add description text to further identify the rule.
8. Click **Finish** to return to the main Windows Firewall screen.
9. Scroll through the Inbound Rules list and verify that the list shows the rule that you just entered.

Inbound Rules			
Name	Group	Profile	Enabled
✔ SCCMRT_TCP_2704		Domain	Yes
✔ secondary HTTP Port		All	Yes
✔ Status Port		All	Yes
✔ TekExpress		All	Yes
✔ TekExpress 100GBASE-SR4		All	Yes
✔ TekExpress CEI-VSR Application		All	Yes
✔ TekExpressSocketClient		All	Yes
✔ TekScope		All	Yes

10. Repeat steps 1 through 9 to enter the other TekExpress executable if it is missing from the list. Enter **TekExpress PI** as the name.
11. Scroll through the Inbound Rules list and verify that the list shows the rule that you just entered.
12. Exit the Windows Firewall tool.

### To use the remote access:

1. Obtain the IP address of the instrument on which you are running TekExpress 100GBASE-SR4. For example, 192.168.34.10.
2. On the PC from which you are accessing the remote instrument, use the instrument IP address as part of the TekExpress 100GBASE-SR4 PI code to access that instrument. For example:

```
object obj = piClient.Connect("192.168.34.10",out clientid);
```

## Requirements for developing TekExpress client

Use the TekExpressClient.dll to develop your client. The client can be a VB .Net, C# .Net, Python, or Web application. The examples for interfaces in each of these applications are in the Samples folder.

### References required

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

---

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

---

## Required steps for a client

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

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

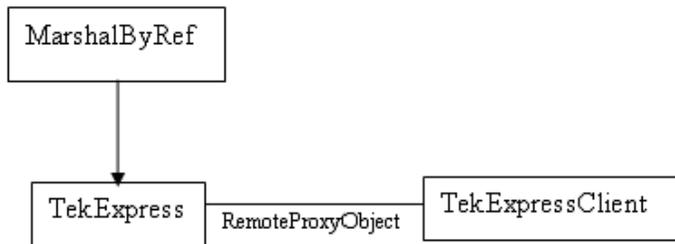
1. To connect to the server, the client provides the IP address of the PC where the server is running.
2. The client locks the server application to avoid conflict with any other Client that may try to control the server simultaneously. "Lock" would also disable all user controls on the server so that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

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

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

## Remote proxy object

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



The following is an example:

```

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

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

For example,

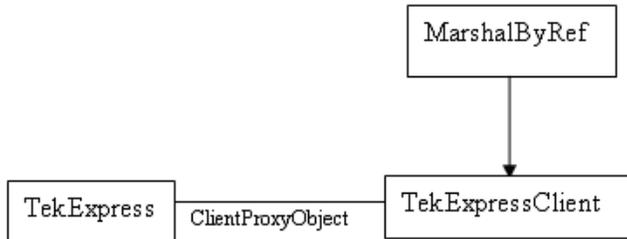
```

//Get a reference to the remote object

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

## Client proxy object

Client exposes a proxy object to receive certain information.



For example,

```
//Register the client proxy object
```

```
wellKnownServiceTypeEntry[] e =
RemotingConfiguration.GetRegisteredWellKnownServiceTypes();
```

```
clientInterface = new ClientInterface();
```

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

```
//Expose the client proxy object through marshalling
```

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

The client proxy object is used for the following:

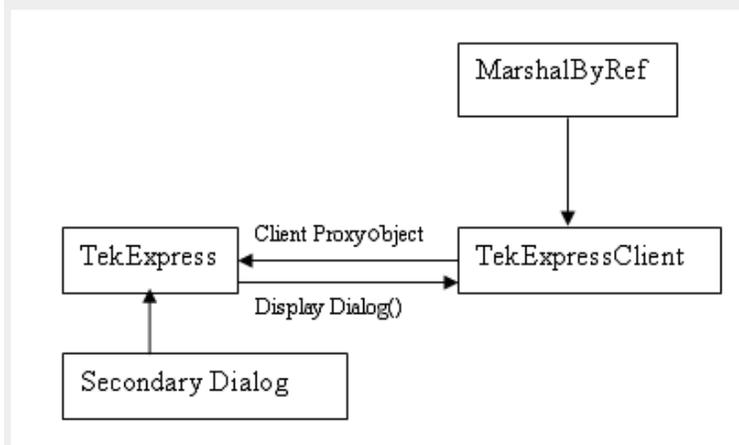
- To get the secondary dialog messages from the server.
- To get the file transfer commands from the server while transferring the report.

Examples

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

For more information, click the following links:

[Secondary Dialog Message Handling](#)



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

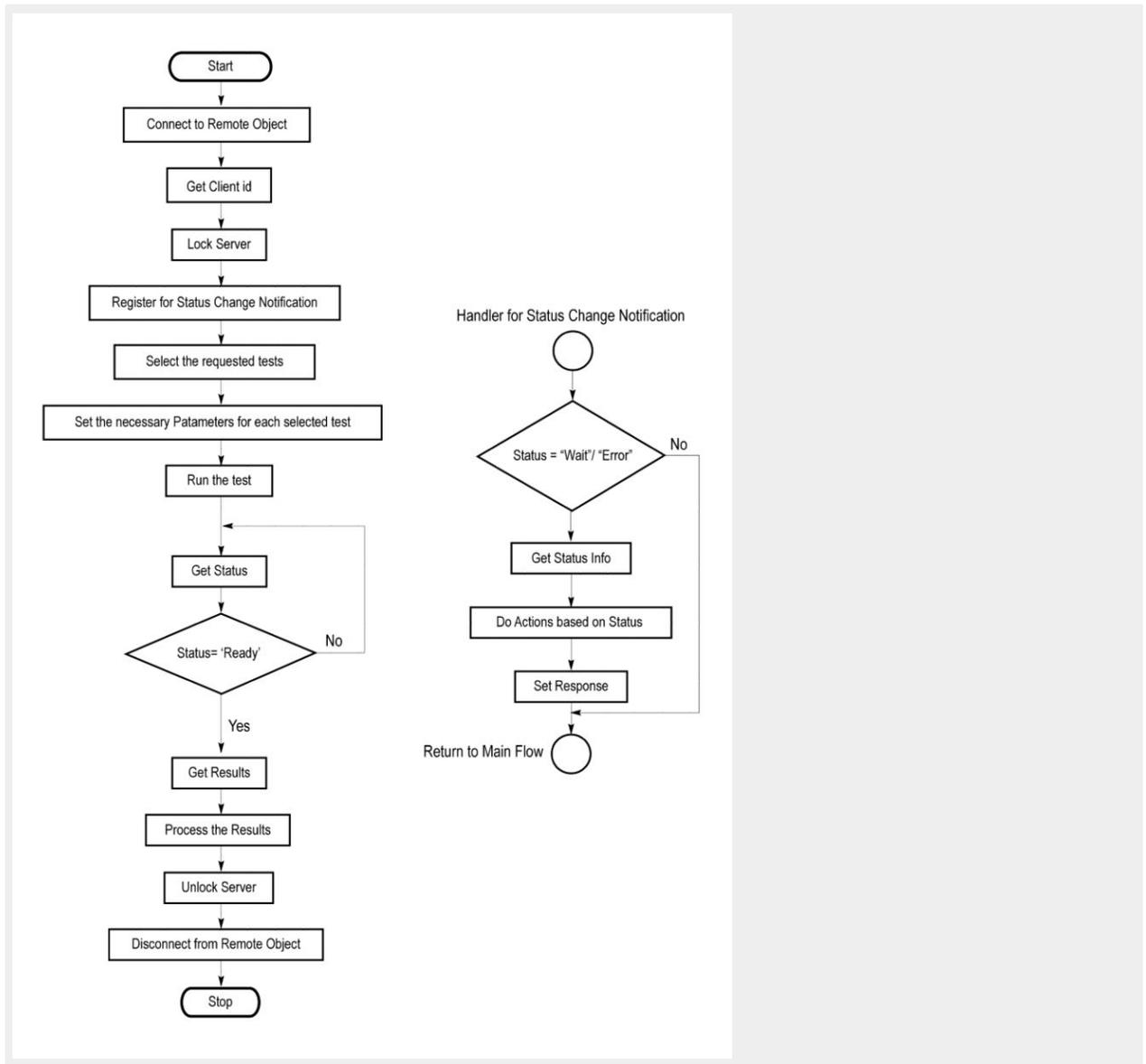
In the secondary dialog library, the assembly that is calling for the dialog box to be displayed is checked and if a remote connection is detected, the messages are directed to the remote end.

#### File Transfer Events

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

## Client programmatic interface example

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



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

---

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

---

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

---

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

---

4. Register for receiving notifications on status change events on the server. To register you need to give a handler as a parameter.

---

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

---

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

**NOTE.** Skip step 8 if you are registered for the status change notification and the status is Ready.

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

### Handler of status change notification

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

### See also

[Program remote access code example](#)

## Program remote access code example

This code example shows how to communicate between a remote PC and TekExpress 100GBASE-SR4.

**Table 17: Remote access code example**

Task	Code
Start the application	
Connect through an IP address.	'assigns client IP address to variable clientID; address valid until connection or measurement session ends (Disconnect). See Connect() clientID = " " m_Client.Connect("localhost",out clientID)'True or False
Lock the server	m_Client.LockServer(clientID)
Disable the Popups	m_Client.SetVerboseMode(clientID, false)
Set the DUT ID	m_Client.SetDutId(clientID, "DUT_Name")
Run with set configurations	m_Client.Run(clientID)
Wait for the test to complete.	Do Thread.sleep(500) m_Client.Application_Status(clientID) Select Case status Case "wait"

Task	Code
Get the current state information	<code>mClient.GetCurrentStateInfo(clientID, waitingMsBxCaption, waitingMsBxMessage, waitingMsBxBUTTONtexts)</code>
Send the response	<code>mClient.SendResponse(clientID, waitingMsBxCaption, waitingMsBxMessage, waitingMsBxBUTTONResponse)</code> End Select Loop Until status = "Ready"
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	

## 100GBASE-SR4 programmer interface commands

### ApplicationStatus()

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

#### Parameters.

Name	Type	Direction	Description
clientId	string	IN	Identifier of the client that is performing the remote function. clientId variable <b>clientId variable</b> clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable: <code>clientId = " "</code> <code>m_Client.Connect("localhost",out clientId)'True or False</code> The clientId variable is stored until you call the Disconnect command.

**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...".

**Return value.** String value that gives the status of the server application.

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

```
returnval as string
```

```
returnval=m_Client.ApplicationStatus(clientID)
```

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

**Related command(s).** [GetCurrentStateInfo](#)

[QueryStatus](#)

[SendResponse](#)

[Status](#)

## ChangeDutId()

**ChangeDutId(clientId, dutName).** This command changes the DUT id of the set-up. The client has to provide a valid DUT id.

**Parameters.**

Parameter	Type	Direction	Description
clientId	String	IN	Identifier of the client that is performing the remote function. clientId variable <b>clientId variable</b> clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable: <code>clientId = " "</code> <code>m_Client.Connect("localhost",out clientId)'True or False</code> The clientId variable is stored until you call the Disconnect command.
dutName	String	IN	The new DUT id of the set-up.

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

**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...".

```

Example. If (dut Id.Length <=0 && locked == true)
return "Enter a valid DUT-ID";
returnVal = remoteObject.ChangeDutId(clientId, dutId);
if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)
return "DUT Id Changed...";
else
return CommandFailed(returnVal);

```

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

**Related command(s).** [GetDutId](#)

### CheckSessionSaved()

**CheckSessionSaved(clientID, out savedStatus).** This command checks whether the current session is saved.

**Parameters.**

Parameter	Type	Direction	Description
HostIPAddress	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.
clientId	string	IN	Identifier of the client that is performing the remote function. clientId variable
savedStatus	boolean	OUT	Boolean representing whether the current session is saved <b>clientId variable</b> clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable: <code>clientId = " "</code> <code>m_Client.Connect("localhost",out clientId)'True or False</code> The clientId variable is stored until you call the Disconnect command.

**Return value.** Return value is either True or False.

**Example.** 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)

**Related command(s).** [GetDutId](#)

**Comments.**

## Connect()

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

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

### Parameters.

Parameter	Type	Direction	Description
HostIPAddress	string	IN	Obtains the IP address of the server to which the client is trying to connect. This is required to establish the connection between the server and the client.
clientId	string	OUT	<p>Identifier of the client that is performing the remote function. clientId variable</p> <p><b>clientId variable</b></p> <p>clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable:</p> <pre>clientId = " "</pre> <pre>m_Client.Connect("localhost",out clientId)'True or False</pre> <p>The clientId variable is stored until you call the Disconnect command.</p>

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

**NOTE.** The Fail condition for 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...".

**Example.** try {

```
m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL
clientId = " "
m_Client.Connect("localhost",out clientId)'True or False
}
```

**Comments.** The server has to be active and running for the client to connect to the server. Any number of clients can be connected to the server at a time. Each client will get a unique id.

**Related command(s).** [Disconnect](#)

## Disconnect()

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

### Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	<p>Identifier of the client that is performing the remote function. clientId variable</p> <p><b>clientId variable</b></p> <p>clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable:</p> <pre>clientId = " "</pre> <pre>m_Client.Connect("localhost",out clientId)'True or False</pre> <p>The clientId variable is stored until you call the Disconnect command.</p>

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

1: Success

-1: Failure

### Example. try

```
{
string returnVal = UnlockServer (clientId);
remoteObject.Disconnect (clientId);
return 1;
}
```

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

**Related command(s).** [Connect](#)

## GetCurrentStateInfo()

**GetCurrentStateInfo(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxButtontexts).** This command gets the additional information of the states when the application is in Wait or Error state.

Except client ID, all the others are Out parameters.

---

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

---

**Parameters.**

Parameter	Type	Direction	Description
clientId	string	IN	<p>Identifier of the client that is performing t</p> <p><b>clientId variable</b></p> <p>clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable:</p> <pre>clientId = " "</pre> <pre>m_Client.Connect("localhost",out clientId)'True or False</pre> <p>The clientId variable is stored until you call the Disconnect command.</p> <p>he remote function.</p> <p>clientId variable</p> <p><b>clientId variable</b></p> <p>clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable:</p> <pre>clientId = " "</pre> <pre>m_Client.Connect("localhost",out clientId)'True or False</pre> <p>The clientId variable is stored until you call the Disconnect command.</p>
WaitingMsbBxCaption	string	OUT	The wait state or error state message sent to you
WaitingMsbBxMessage	string	OUT	The wait state/error state message sent to you
WaitingMsbBxButtontexts	string array	OUT	An array of strings containing the possible response types that you can send

**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...".

**Return value.** This command does not return any value.

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

**Example.** m\_Client = new Client() //m\_Client is a reference to the Client class in the Client DLL  
mClient.GetCurrentStateInfo(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxButtontexts)

**Comments.**

**Related command(s).** [ApplicationStatus](#)

[QueryStatus](#)

[SendResponse](#)

## GetDutId()

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

**Parameters.**

Parameter	Type	Direction	Description
clientId	string	IN	<p>Identifier of the client that is performing the remote function. clientId variable</p> <p><b>clientId variable</b></p> <p>clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable:</p> <pre>clientId = " "</pre> <pre>m_Client.Connect("localhost",out clientId)'True or False</pre> <p>The clientId variable is stored until you call the Disconnect command.</p>
dutId	string	OUT	The DUT id of the setup.

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

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

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

```
{
return id;
}
```

```
else
return CommandFailed(returnVal);
```

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

**Related command(s).** [ChangeDutId](#)

[SetDutId](#)

## GetPassFailStatus()

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

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

### Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable <b>clientId variable</b> clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable: <pre>clientId = " " m_Client.Connect("localhost",out clientId)'True or False</pre> The clientId variable is stored until you call the Disconnect command.
device	string	IN	Specifies the DUT type ( <b>100GBASE-SR4</b> )
suite	string	IN	string with device connection type. Suite type is TP2
test	string	IN	Specifies the name of the test for which to obtain the pass or fail status.

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

**Example.** GetPassFailStatus(clientId, "100GBASE-SR4", "TP2", test);

GetPassFailStatus(clientId, "Host", "TP2", test);

## GetReportParameter()

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

**Parameters.**

Parameter	Type	Direction	Description
clientId	string	IN	<p>Identifier of the client that is connected to the server clientId variable</p> <p><b>clientId variable</b></p> <p>clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable:</p> <pre>clientId = " "</pre> <pre>m_Client.Connect("localhost",out clientId)'True or False</pre> <p>The clientId variable is stored until you call the Disconnect command.</p>
device	string	IN	Specifies the DUT type ( <b>10GBASE-SR4</b> ).
suite	string	IN	string with device connection type. Suite type is TP2.
test	string	IN	Specifies the name of the test for which to obtain the pass or fail status or a test result value.
parameterString	string	IN	Specifies to return the measured value for the indicated test. Enter <b>"Scope Model"</b> , <b>"TekExpress Version"</b> , or <b>"Application Version"</b> for this argument

**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...".

**Return value.** The return value is the connected oscilloscope model, TekExpress base software version, or 10GBASE-SR4 application version.

**Example.** GetReportParameter(clientId, "Device", "suite", test, "Application Version")

## GetResultsValue()

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

### Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable <b>clientId variable</b> clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable: <code>clientId = " "</code> <code>m_Client.Connect("localhost",out clientId)'True or False</code> The clientId variable is stored until you call the Disconnect command.
device	string	IN	Specifies the DUT type ( <b>100GBASE-SR4</b> ).
suite	string	IN	string with device connection type. Suite type is TP2.
test	string	IN	Specifies the name of the test for which to obtain the test result value.
parameterString	string	IN	Specifies to return the measured value for the indicated test. Enter <b>"Value"</b> for this argument

**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...".

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

**Example.** GetResultsValue(clientId, "Device", "suite", test, "Value");

## GetTimeOut()

**GetTimeOut(clientId)**. Returns the current timeout period set by the client.

### Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	<p>Identifier of the client that is connected to the server clientId variable</p> <p><b>clientId variable</b></p> <p>clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable:</p> <pre>clientId = " "</pre> <pre>m_Client.Connect("localhost",out clientId)'True or False</pre> <p>The clientId variable is stored until you call the Disconnect command.</p>

**Return value.** String value that indicates the status of the operation upon completion. The default return value is 1800000.  
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...".

**Example.** m\_Client = new Client() //m\_Client is a reference to the Client class in the Client DLL.

returnval as string

returnval=m\_Client.GetTimeOut()

### Comments.

**Related command(s).** [SetTimeOut](#)

## LockSession()

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

**Parameters.**

Parameter	Type	Direction	Description
clientId	string	IN	<p>Identifier of the client that is performing the remote function. clientId variable</p> <p><b>clientId variable</b></p> <p>clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable:</p> <pre>clientId = " "</pre> <pre>m_Client.Connect("localhost",out clientId)'True or False</pre> <p>The clientId variable is stored until you call the Disconnect command.</p>

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

**Example.** if (locked)

```
return "Session has already been locked!";
returnVal = remoteObject.LockSession(clientId);
if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)
{
  locked = true;
  return "Session Locked...";
}
```

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

If the client locks the server and is idle for a certain amount of time, then the server is automatically unlocked from that client.

**Related command(s).** [UnlockSession](#)

**QueryStatus()**

**QueryStatus(clientID, out status).** This command transfers Analyze panel status messages from the server to the client.

**Parameters.**

Parameter	Type	Direction	Description
clientID	string	IN	<p>Identifier of the client that is connected to the server clientId variable</p> <p><b>clientId variable</b></p> <p>clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable:</p> <pre>clientId = " "</pre> <pre>m_Client.Connect("localhost",out clientId)'True or False</pre> <p>The clientId variable is stored until you call the Disconnect command.</p>
status	string array	OUT	The list of status messages generated during the run

---

**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...".

---

**Return value.** String value that indicates the status of the operation upon completion. On success the return value is "Transferred...".

**Example.** returnVal=m\_Client.QueryStatus(clientID, out statusMessages)

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

```
return "Status updated..."
```

```
else
```

```
return CommandFailed(returnVal)
```

**Related command(s).** [ApplicationStatus](#)

[GetCurrentStateInfo](#)

[SendResponse](#)

## RecallSession()

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

### Parameters.

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

### clientId variable

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

```
clientId = " "
```

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

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

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

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

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

```
return "Session Recalled...";
```

```
else
```

```
return CommandFailed(returnVal);
```

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

**Related command(s).** [SaveSession](#)

[SaveSessionAs](#)

## Run()

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

### Parameters.

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

### clientId variable

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

```
clientId = " "
```

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

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

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

```
Example. returnVal = remoteObject.Run(clientId);
if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)
return "Run started...";
else
return CommandFailed(returnVal);
```

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

## SaveSession()

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

**Parameters.**

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

### clientId variable

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

```
clientId = " "
```

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

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

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

```
Example. returnVal = remoteObject.SaveSession(clientId,sessionName);
if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)
return "Session saved...";
else
return CommandFailed(returnVal);
```

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

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

**Related command(s).** [RecallSession](#)

[SaveSessionAs](#)

## SaveSessionAs()

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

### Parameters.

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

### clientId variable

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

```
clientId = " "
```

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

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

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

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

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

```
return "Session saved...";
```

```
else
```

```
return CommandFailed(returnVal);
```

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

**Related command(s).** [RecallSession](#)

[SaveSession](#)

## SendResponse()

**SendResponse(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxButtontexts).** After receiving the additional information using the command GetCurrentStateInfo(), the client can decide which response to send and then send the response to the application using this function. The response should be one of the strings that was received earlier as a string array in the GetCurrentStateInfo function. The \_caption and \_message should match the information received earlier in the GetCurrentStateInfo function.

---

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

---

**Parameters.**

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

**clientId variable**

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

```
clientId = " "
```

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

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

---

**NOTE.** The Fail condition for 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...".

---

**Return value.** This command does not return any value.

**Example.** m\_Client = new Client() //m\_Client is a reference to the Client class in the Client DLL

```
mClient.SendResponse(clientID, out WaitingMsBxCaption, out WaitingMsBxMessage, out WaitingMsBxButtontexts)
```

**Related command(s).** [ApplicationStatus](#)

[GetCurrentStateInfo](#)

[QueryStatus](#)

## SelectDevice()

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

### Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable
device	string	IN	Specifies the DUT type ( <b>100GBASE-SR4</b> )

### clientId variable

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

```
clientId = " "
```

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

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

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

**Example.** SelectDevice(clientId,"100GBASE-SR4", True)

## SelectSuite()

**SelectSuite(clientId, device, suite, true).** This command selects TP2 suite.

### Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable
device	string	IN	Specifies the DUT type ( <b>100GBASE-SR4</b> )
suite	string	IN	string with device connection type. Suite type is TP2.

### clientId variable

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

```
clientId = " "
```

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

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

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

**Example.** SelectSuite(clientId,"100GBASE-SR4","TP2",true);

```
SelectSuite(clientId,"100GBASE-SR4","TP2",true);
```

## SelectTest()

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

### Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable <b>clientId variable</b> clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable: <code>clientId = " "</code> <code>m_Client.Connect("localhost",out clientId)'True or False</code> The clientId variable is stored until you call the Disconnect command.
device	string	IN	Specifies the DUT type (100GBASE-SR4).
suite	string	IN	string with device connection type. Suite type is TP2
test	string	IN	Name of the 100GBASE-SR4 test.

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

**Example.** `SelectTest(clientId, device, suite, "Optical Modulation Amplitude", true);`

## SetDutId()

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

### Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable <b>clientId variable</b> clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable: <code>clientId = " "</code> <code>m_Client.Connect("localhost",out clientId)'True or False</code> The clientId variable is stored until you call the Disconnect command.
newDutId	string	IN	The new DUT ID of the setup.

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

Return value is "DUT Id Changed" on success.

**Example.** `m_Client = new Client()` //m\_Client is a reference to the Client class in the Client DLL.

`returnval as string`

`return=m_Client.SetDutId(clientID,desiredDutId)`

**Comments.**

**Related command(s).** [GetDutId](#)

## The SetAcquireParameter command

**SetAcquireParameter()**. SetAcquireParameter(clientId, device, suite, testName, paramString)Parameters  
**clientId variable**

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

```
clientId = " "
```

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

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

paramString argument values

This command sets the acquire parameter and its value based on the "paramString" argument values as listed. TekExpress programmatic interface SetAcquireParameter ()

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable
device	string	IN	Specifies the DUT type ( <b>100GBASE-SR4</b> ).
suite	string	IN	Valid values are <b>TP2</b> .
test	string	IN	Specifies the name of the test for which to set the value for acquire parameter.
parameterString	string	IN	Specifies the control to set. See the following links for argument values and examples for this field.

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

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

Use the following links to see the paramString values associated with specific application settings.

[Select Mask file for Transmitter Eye Mask](#)

[Set TimeLimit value for Extinction Ratio](#)

[Set Temperature Drift Tolerance value for Extinction Ratio](#)

[Set Method for Average Launch Power](#)

## paramString values for SetAcquireParameter command

**Select Mask file for Transmitter Eye Mask.** Use paramString value to set the mask file path for Transmitter Eye Mask. This is same as selecting the mask file from Configuration > Measurements > Transmitter Eye Mask tab.

The value in bold font is the default value.

### Values:. For test Transmitter Eye Mask

- EyeMask\_Acq\$Mask File\$C:\Users\Public\Documents\Tektronix\Masks\IEEE\Ethernet 802.3\100G\_40G\LAN\100GBASE\_SR4Draft3p3\100GbE\_SR4Txv33.m8k
- EyeMask\_Acq\$Mask File\$C:\Eye\_Mask\_MaskFiles\10GBASER.m8k

**Example.** SetAcquireParameter(clientId, "100GBASE-SR4", "TP2", "Transmitter Eye Mask", "EyeMask\_Acq\$Mask File\$C:\Eye\_Mask\_MaskFiles\10GBASER.m8k");

**Set TimeLimit value for Extinction Ratio.** Use paramString value to set the TimeLimit value of Extinction Ratio. This is similar to selecting the Time limit from Configuration > Measurements > Extinction Ratio test tab.

The value in bold font is the default value.

### Values:. For Extinction Ratio

- SR4\_Acq\$TimeLimit\$5
- SR4\_Acq\$TimeLimit\$24

**Example.** SetAcquireParameter(clientId, "100GBASE-SR4", "TP2", "Extinction Ratio", "SR4\_Acq\$TimeLimit\$5");

**Set Temperature Drift Tolerance value for Extinction Ratio.** Use paramString value to set the Temperature Drift Tolerance value for Extinction Ratio. This is similar to selecting the Time limit from Configuration > Measurements > Extinction Ratio test tab.

The value in bold font is the default value.

### Values:. For Extinction Ratio

- SR4\_Acq\$TemperatureDriftTolerance\$2
- SR4\_Acq\$Temperature Drift Tolerance\$5

**Example.** SetAcquireParameter(clientId, "100GBASE-SR4", "TP2", "Extinction Ratio", "SR4\_Acq\$TemperatureDriftTolerance\$2");

**Set Method for Average Launch Power.** Use paramString value to set the Method for Average Launch Power. This is similar to selecting the Time limit from Configuration > Measurements > Extinction Ratio test tab.

The value in bold font is the default value.

### Values:. For Average Launch Power

- ALP\_Acq\$Method\$Power-Meter/External
- ALP\_Acq\$Method\$Histogram

**Example.** SetAcquireParameter(clientId, "100GBASE-SR4", "TP2", "Average Launch Power", "ALP\_Acq\$Method\$Power-Meter/External");

## The SetGeneralParameter command

**Select data rate.** Use this paramString value to set the Data Rate used by the application. This is the same as selecting the Data Rate control on the DUT tab.

The value in bold font is the default value.

**Values:.**

- **Data Rate\$25.781 Gbps**
- Data Rate\$27.952 Gbps
- Data Rate\$28.050 Gbps
- Data Rate\$Custom

**Example.** SetGeneralParameter(clientId, "100GBASE-SR4", "TP2", "", "Data Rate (Gbps)\$27.952 Gbps");

---

**NOTE.** To select the custom option in the data rate drop down following command need to be run as a pre-requisite:

- SetComplianceMode(clientID, User-Defined)
- 

**Select custom data rate.** Use this paramString value to set the custom data rate used by the application. This is the same as selecting the custom data rate control on the DUT tab.

The value in bold font is the default value.

**Values:.**

- Custom Data Rate\$<Enter any value between the low and high limits of the Custom Data rate option>

**Example.** Custom Data Rate\$15

---

**NOTE.** To run this command following pre-requisite commands should be run prior with the same following sequence:

- SetComplianceMode(clientID, User-Defined)
  - Data Rate\$Custom
- 

**Select wavelength.** Use this paramString value to set the Wavelength used by the application. This is same as selecting wavelength on the DUT tab.

The value in bold font is the default value.

**Values:.**

- WaveLength(nm)\$1550 : FACTORY
- WaveLength(nm)\$1310 : FACTORY
- WaveLength(nm)\$850 : FACTORY
- WaveLength(nm)\$1550 : USER

**Example.** SetGeneralParameter(clientId, "100GBASE-SR4", "TP2", "", "WaveLength(nm)\$1550 : FACTORY");

---

**NOTE.** The example values are provided considering the 80C15 module is connected to a scope.

---

**Select OMA method.** Use this paramString value to set the pattern for OMA method used by the application. This is same as selecting OMA method on Setup -> Configuration -> Global Settings -> OMA Method.

The value in bold font is the default value.

**Values:.**

- OMA Method\$OMA-Eye
- OMA Method\$OMA-Pattern

**Example.** `SetGeneralParameter(clientId, "100GBASE-SR4", "TP2", "", "OMA Method$OMA-Eye");`

**Select pattern for OMA-pattern.** Use this paramString value to set the pattern for OMA-pattern used by the application. This is same as selecting OMA-pattern on Setup -> Configuration -> Global Settings -> OMA Method -> OMA-pattern -> Pattern.

The value in bold font is the default value.

**Values:.**

- Data Pattern\$PRBS7
- Data Pattern\$PRBS9
- Data Pattern\$PRBS11
- Data Pattern\$PRBS15

**Example.** `SetGeneralParameter(clientId, "100GBASE-SR4", "TP2", "", "Data Pattern$PRBS9");`

---

**NOTE.** To run this command, following command is the prerequisite.

- `SetGeneralParameter(clientId, "100GBASE-SR4", "TP2", "", "OMA Method$OMA-Pattern");`
- 

**Select trigger source.** Use this paramString value to set the Trigger Source used by the application. This is same as selecting SR4 filter on Setup -> Configuration -> Global Settings -> Trigger Source.

The value in bold font is the default value.

**Values:.**

- Trigger Source\$Tek CRU
- Trigger Source\$Others

**Example.** `SetGeneralParameter(clientId, "100GBASE-SR4", "TP2", "", "Trigger Source$TekCRU");`

**Select data to clock ratio.** Use this paramString value to set the Data to Clock ratio used by the application. This is same as selecting SR4 filter on Setup -> Configuration -> Global Settings -> Trigger Source -> Others -> Clock Divider.

The value in bold font is the default value.

**Values:.**

- Data to Clock ratio\$1
- Data to Clock ratio\$2
- Data to Clock ratio\$4
- Data to Clock ratio\$8

**Example.** SetGeneralParameter(clientId, "100GBASE-SR4", "TP2", "", "Data to Clock ratio\$1");

---

**NOTE.** To run this command, following command is the prerequisite.

- SetGeneralParameter(clientId, "100GBASE-SR4", "TP2", "", "Trigger Source\$Others");
- 

**Select SR4 filter.** Use this paramString value to set the SR4 filter used by the application. This is same as selecting SR4 filter on Setup -> Configuration -> Global Settings -> SR4 Signal Conditioning -> Filter.

The value in bold font is the default value.

**Values:.**

- Filter\$OTU-4
- Filter\$32GFCr0
- Filter\$100GBASE-R4
- Filter\$INF25781

**Example.** SetGeneralParameter(clientId, "100GBASE-SR4", "TP2", "", "Filter\$OTU-4");

---

**NOTE.** The example values are provided, considering the 80C15 module is connected to a scope.

---

**Select SR4 bandwidth.** Use this paramString value to set the SR4 bandwidth used by the application. This is same as selecting SR4 bandwidth on Setup -> Configuration -> Global Settings -> SR4 Signal Conditioning -> Bandwidth.

The value in bold font is the default value.

**Values:.**

- BandWidth\$32.00GHz
- BandWidth\$28.05GHz
- BandWidth\$22.00GHz

**Example.** SetGeneralParameter(clientId, "100GBASE-SR4", "TP2", "", "BandWidth\$32.00GHz");

---

**NOTE.** The example values are provided considering the 80C15 module is connected to a scope.

---

**Select TDEC filter.** Use this paramString value to set the TDEC filter used by the application. This is same as selecting TDEC filter on Setup -> Configuration -> Global Settings -> TDEC Signal Conditioning -> Filter.

The value in bold font is the default value.

**Values:**

- Filter\_TDECGlobal\$OTU-4
- Filter\_TDECGlobal\$32GFCr0
- Filter\_TDECGlobal\$100GBASE-R4
- Filter\_TDECGlobal\$INF25781

**Example.** SetGeneralParameter(clientId, "100GBASE-SR4", "TP2", "", "Filter\_TDECGlobal\$OTU-4");

---

**NOTE.** The example values are provided considering the 80C15 module is connected to a scope.

---

**Select TDEC bandwidth.** Use this paramString value to set the TDEC bandwidth used by the application. This is same as selecting TDEC bandwidth on Setup -> Configuration -> Global Settings -> TDEC Signal Conditioning -> Bandwidth.

The value in bold font is the default value.

**Values:**

- BandWidth\_TDECGlobal\$32.00GHz
- BandWidth\_TDECGlobal\$28.05GHz
- BandWidth\_TDECGlobal\$22.00GHz

**Example.** SetGeneralParameter(clientId, "100GBASE-SR4", "TP2", "", "BandWidth\_TDECGlobal\$32.00GHz");

---

**NOTE.** The example values are provided considering the 80C15 module is connected to a scope.

---

**Select TDEC BER mantissa.** Use this paramString value to set the TDEC BER mantissa used by the application. This is same as selecting TDEC BER mantissa on Setup -> Configuration -> Global Settings -> TDEC Signal Conditioning -> BER (Mantissa editbox).

The value in bold font is the default value.

**Values:**

- BER\_Mantissa\_TDECGlobal\$<Enter any value between the low and high limits of the TDEC BER Mantissa edit box>

**Example.** SetGeneralParameter(clientId, "100GBASE-SR4", "TP2", "", "BER\_Mantissa\_TDECGlobal\$5.0");

**Select TDEC BER exponent.** Use this paramString value to set the TDEC BER exponent used by the application. This is same as selecting TDEC BER exponent on Setup -> Configuration -> Global Settings -> TDEC Signal Conditioning -> BER (Exponent editbox).

The value in bold font is the default value.

**Values:**

- BER\_Exponent\_TDECGlobal\$<Enter any value between the low and high limits of the TDEC BER Exponent edit box>

**Example.** SetGeneralParameter(clientId, "100GBASE-SR4", "TP2", "", "BER\_Exponent\_TDECGlobal\$5");

**Select TDEC histogram hits.** Use this paramString value to set the TDEC Histogram hits used by the application. This is same as selecting TDEC histogram hits on Setup -> Configuration -> Global Settings -> TDEC Signal Conditioning -> Histogram Hits.

The value in bold font is the default value.

**Values:**

- Histogram\_Hits\_TDECGlobal\$<Enter any value between the low and high limits of the TDEC BER Histogram hits edit box>

**Example.** SetGeneralParameter(clientId, "100GBASE-SR4", "TP2", "", "Histogram\_Hits\_TDECGlobal\$100000");

**NOTE.** To run this command, following command is the prerequisite.

- SetComplianceMode(clientID, User-Defined)

## SetTimeout()

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

**Parameters.**

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

### clientId variable

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

```
clientId = " "
```

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

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

**Return value.** String value that indicates the status of the operation upon completion. On success the return value is "TimeOut Period Changed".

**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...".

**Example.** m\_Client = new Client() //m\_Client is a reference to the Client class in the Client DLL.

returnval as string

```
returnval=m_Client.SetTimeout(clientID, time)
```

**Comments.**

**setVerboseMode()**

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

When the value is set to true, any message boxes that appear during the application are routed to the client machine that is controlling TekExpress.

When the value is set to false, all the message boxes are shown on the server machine.

**Parameters.**

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

**clientId variable**

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

```
clientId = " "
```

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

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

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

When Verbose mode is set to true, the return value is "Verbose mode turned on. All dialog boxes will be shown to client".

When Verbose mode is set to false, the return value is "Verbose mode turned off. All dialog boxes will be shown to server".

---

**NOTE.** The Fail condition for 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...".

---

**Example.** m\_Client = new Client() //m\_Client is a reference to the Client class in the Client DLL.

**Turn on verbose mode:**

```
return=m_Client.SetVerboseMode(clientId, true)
```

**Turn off verbose mode:**

```
returnval=m_Client.SetVerboseMode(clientId, false)
```

## Status()

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

### Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable <b>clientId variable</b> clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable: <code>clientId = " "</code> <code>m_Client.Connect("localhost",out clientId)'True or False</code> The clientId variable is stored until you call the Disconnect command.
statusMessage	string array	OUT	The list of status messages generated during run.

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

```
Example. returnVal = remoteObject.QueryStatus(clientId, out statusMessages);
if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)
return "Status updated...";
else
return CommandFailed(returnVal);
```

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

**Related command(s).** [ApplicationStatus](#)

## Stop()

**Stop(clientId).** Stops the run operation.

**Parameters.**

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

**clientId variable**

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

```
clientId = " "
```

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

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

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

```
Example. returnVal = remoteObject.Stop(clientId);
if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)
return "Stopped...";
else
return CommandFailed(returnVal);
```

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

**TransferImages()**

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

---

**NOTE.** Every time you click Start, a folder is created in the X: drive. Transfer the waveforms before clicking Start.

---

**Parameters.**

Parameter	Type	Direction	Description
clientId	string	IN	<p>Identifier of the client that is connected to the server clientId variable</p> <p><b>clientId variable</b></p> <p>clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable:</p> <pre>clientId = " "</pre> <pre>m_Client.Connect("localhost",out clientId)'True or False</pre> <p>The clientId variable is stored until you call the Disconnect command.</p>
filePath	string	IN	<p>The location where the screen shots must be saved in the client.</p> <hr/> <p><b>NOTE.</b> If the client does not provide the location to save the report, the report is saved at C:\ProgramFiles.</p> <hr/>

**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...".

**Return value.** String value that indicates the status of the operation upon completion. Transfers all the images in the form of a string.

**Example.** TransferImages(clientId, "C:\Waveforms")

## TransferReport()

**TransferReport(clientId, filePath).** This command transfers the report generated after the run to the specified folder (directory). The report contains the summary of the run. The client has to provide the location where the report is to be saved at the client-end.

**Parameters.**

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable <b>clientId variable</b> clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable: <pre>clientId = " "</pre> <pre>m_Client.Connect("localhost",out clientId)'True or False</pre> The clientId variable is stored until you call the Disconnect command.
filePath	string	IN	Path to the target folder to which to transfer the report file. Enclose the path in quotes.

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

**Example.** TransferReport(clientId, "C:\Report")

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

**UnlockSession()**

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

**Parameters.**

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

**clientId variable**

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

```
clientId = " "
```

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

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

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

```
Example. returnVal = remoteObject.UnlockSession(clientId);  
if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)  
{  
    locked = false;  
    return "Session UnLocked...";  
}
```

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

**Related commands.** [LockSession](#)



# References

## 100GBASE-SR4 technology overview

100GBASE-SR4 is a 100 Gigabit Ethernet technology based application. It provides point to point 100 Gb/s Ethernet link over four pair of multimode fiber, up to 100 meters. This application uses eight fiber links with four 25.78125 Gb/s lanes for each direction.

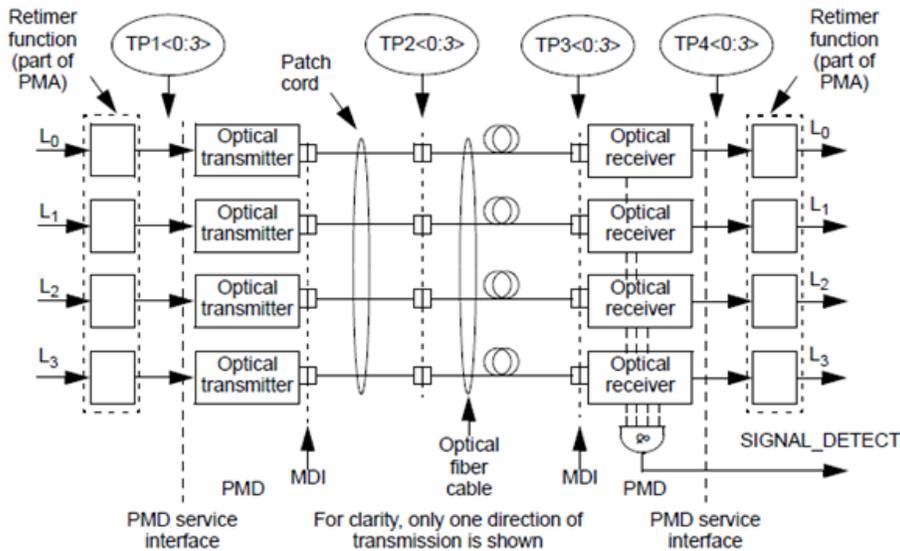
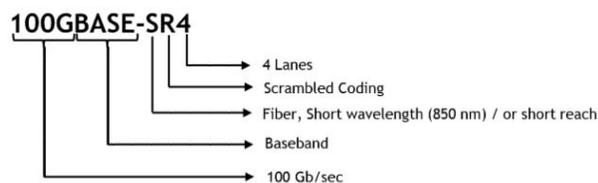


Figure 3: Block diagram of 100GBASE-SR4 Transmit/Receive paths

Interpretation of 100GBASE-SR4 is as follows:



The optical transmit signal is defined at the output end of the multimode fiber patch cord (TP2), between 2m and 5m in length. The measurements supported at TP2 are the following:

- Signaling Rate
- Average Launch Power, each lane
- Optical Modulation Amplitude (OMA)
- Transmitter and Dispersion Eye Closure (TDEC)
- Launch Power in OMA minus TDEC
- Average Launch Power of OFF Transmitter
- Extinction Ratio
- Transmitter Eye Mask

## Optical configurations

**Table 18: Optical configurations for 100GBASE-SR4 measurements except TDEC**

Data Rate (Gbps)	SR4 measurements except TDEC		
	80C15		
	Filter	Bandwidth (GHz)	Wavelength (nm)
25.78125	100GBase-R4	None	850
27.952	OTU-4	None	850
28.05	32GFCr0	None	850

If **Custom Data Rate** is selected, then select the required filter / bandwidth in 100GBASE-SR4 optical configuration. TekExpress 100GBASE-SR4 application lists all supported filters and bandwidth.

**Table 19: Optical configurations for TDEC measurement**

Data Rate (Gbps)	TDEC		
	80C15		
	Filter	Bandwidth (GHz)	Wavelength (nm)
25.78125	None	16.8	850
27.952	None	16.8	850
28.05	None	16.8	850

## Clock Recovery Unit (CRU)

Tektronix Clock Recovery Unit is configured with Nominal Data rate (set by user in GUI), Corner frequency of 10 MHz, Slope of 20 dB/decade, i.e, 0 dB peaking, and Lock range of 10 MHz.

The CRU locked data rate value is used as result for Signaling Rate measurement. The following flow diagram gives detailed flow for CRU locking mechanism followed in 100GBASE-SR4 application.

Sub-rate clock output of CRU is given to clock pre scale/trigger input of sampling scope. Clock output of CRU is given to phase reference module.

---

**NOTE.** 100GBASE-SR4 application supports only external Clock Recovery Unit.

---

## Trigger Source

The clock signal (synchronous to data) can be provided using either Tektronix external CRU, or other source having clock signal in synchronous with data, and perform similar to Tektronix external CRU. If the trigger source is others, then you can configure the clock divider parameter.

Clock divider is the ratio of data rate to frequency of clock signal, fed as input to phase reference. It is used to determine the frequency of phase characterization (used only if phase reference module is present in one of the slot in the main frame).

Phase characterization frequency = Data rate / Clock divider

## Phase reference characterization

Phase reference module is not a mandatory requirement for 100GBASE-SR4 measurements. If phase reference module is present in any sampling scope slots, then the setup provides a clock signal synchronous with data as input to phase reference module (can use recovered clock from CRU). Phase reference characterization is done with phase correction mode as “triggered” and input frequency equal to frequency of the input clock signal.

---

**NOTE.** The recovered clock frequency from CR286A is half of the data rate, when the data rate is greater than 14.3 Gb/sec.

---

The 100GBASE-SR4 application uses only one phase reference module; if the system has multiple modules, then the lower numbered slot is used and others are ignored. This slot/channel information is obtained from phase reference source query, using instrument programmatic interface internally.

If there is no phase reference module, then query results in C1C2 (default), and perform an additional query of module’s model number. If the model number is 82A04B, then proceed with phase reference characterization, else skip phase reference characterization.

## Parameters

### About application parameters

This section describes the 100GBASE-SR4 application parameters, and includes the default menu settings.

The parameters for the menus, and options list the selections available for each and include the default values.

### Setup panel configuration parameters

**DUT tab parameters.**

Parameters	Selection	Default Setting
DUTID	-	DUT001
Data Rate	25.781 Gbps, 27.952 Gbps, 28.05 Gbps, Custom (10 to 28.05 Gbps)	25.781 Gbps
Wavelength	List of wavelengths supported by the connected Optical module	850 : FACTORY (if available, else the lowest wavelength supported by the connected optical module)

**Test Selection tab parameters.**

<b>Parameters</b>	<b>Selection</b>	<b>Default Setting</b>
10GBASE-SR4 TX	<ul style="list-style-type: none"><li>■ Average Launch Power of Off Transmitter</li><li>■ Signaling Rate</li><li>■ Transmitter Eye Mask</li><li>■ Average Launch Power</li><li>■ Extinction Ratio</li><li>■ Optical Modulation Amplitude</li><li>■ Transmitter and Dispersion Eye Closure</li><li>■ Launch Power in OMA minus TDEC</li></ul>	All measurements selected

## Configuration tab parameters.

Table 20: Global settings parameters

Parameters		Selection	Default Setting
OMA Method		OMA-Eye, OMA-Pattern	OME-Eye
Trigger Source		Tek CRU, Others	Tek CRU
Clock Divider		1,2,4,8	1
<p><b>NOTE.</b> It is the ratio of Signaling Rate to Phase Reference characterisation frequency. This is displayed when Trigger Source selection is Others</p>			
SR4 Signal Conditioning	Filter	List of Filters supported by the connected optical module	<p>If 80C15 module is selected, then Filter default setting is:</p> <ul style="list-style-type: none"> <li>■ For Data Rate 25.78125 Gbps, Filter is 100GBASE-R4</li> <li>■ For Data Rate 27.952 Gbps, Filter is OTU-4</li> <li>■ For Data Rate 28.05 Gbps, Filter is 32GFCr0</li> </ul> <p>For other optical modules, Filter default setting is selected based on the available filter options.</p>
	Bandwidth	List of Bandwidths supported by the connected optical module	None
TDEC Signal Conditioning	Filter	List of Filters supported by the connected optical module	Filter default setting is selected based on the available filter options.
	Bandwidth	List of Bandwidths supported by the connected optical module	If 80C15 module is selected, then it is 16.84 GHz or None
	BER	1.0E-18 to 9.0E-0	5.0E-5
	Histogram Hits	1000 to 10000000	100000

**Table 21: Measurements parameters**

Parameters		Selection	Default Setting
Transmitter Eye Mask	Mask	-	C:\Users\Public\Documents\ \Tektronix\Masks\IEEE \Ethernet 802.3\100G_40G \LAN \100GBASE_SR4Draft3p3\100 GbE_SR4TXc33.m8k
Average Launch Power	Method	Histogram, Power Meter/ External	Histogram
Extinction Ratio	Time Limit	1 to 100	24 Hours
	Temperature Drift Tolerance	-20 to 20	5°C

**Table 21: Measurements parameters**

Parameters		Selection	Default Setting
Transmitter Eye Mask	Mask	-	C:\Users\Public\Documents\ \Tektronix\Masks\IEEE \Ethernet 802.3\100G_40G \LAN \100GBASE_SR4Draft3p3\100 GbE_SR4TXc33.m8k
Average Launch Power	Method	Histogram, Power Meter/ External	Histogram
Extinction Ratio	Time Limit	1 to 100	24 Hours
	Temperature Drift Tolerance	-20 to 20	5°C

**Preferences tab parameters.**

Parameters	Selection	Default Setting
Acquire/Analyze each test X times	1 to 100	1
Auto close Warnings and Information during Sequencing Auto close after X Seconds	1 to 100	10
Auto close Error Messages during Sequencing, Show in Reports Auto close after X Seconds	1 to 100	10

**Reports panel parameters**

Parameters	Selection	Default Setting
Report name	-	x:\100GBASE-SR4\Reports\DUT001.mht
Save as Type	PDF, MHT	Web Archive (*.mht; *.mhtml)

---

# Index

100GBASE-SR4 features, v

## A

About Application Parameters, 95

About TekExpress, v

Acquire parameters

including in test reports, 28

viewing in reports, 30

Acquisition tab, 20

Analysis options, 23

Application panels overview, 10

Application setup (required steps)

map My TekExpress to drive X:, 5

Application version (show), 5

## B

Button

calibration, 20

clear log, 24

Email settings, 23

save, 24

view optical modules, 20

## C

Calibration button, 20

Client proxy object, 55

Code example, remote access, 58

Configuration tab, 17

Configuration tab parameter

instruments detected, 21

Configuration tab parameters

global settings, 21

Connected instruments

searching for, 14

Connection requirements, 31

Contacting Tektronix, 2

Create a test setup from default settings, 38

Create a test setup using an existing one, 38

## D

DUT ID, 18

DUT parameter

device, 18

device profile, 18

optical module settings, 18

optical power, 18

DUT type

device, 18

DUT-instrument setup, 31

## E

Email notification and setup, 15

Enable remote access, 51

Equipment setup, 31

Extensions, file names, 8

## F

Features (100GBASE-SR4), v

File name extensions, 8

Firewall (remote access), 51

## G

GetReportParameter (PI command), 66

GPIO, 14

## H

Help conventions, 1

## I

Inbound rule wizard (remote access), 51

Installing the software

TekExpress 100GBASE-SR4 Solution, 5

Instrument-DUT setup, 31

Instruments

discovering connected, 14

viewing connected, 14

Instruments and accessories required, 4

Instruments detected, 21

Interface, 51

Interface commands  
  GetReportParameter, 66  
  Select Data Rate, 81–85  
  Set Pattern Type, 80  
  Set Record Length, 80

## K

Keep on top, 9

## L

LAN, 14  
License agreement (show), 5  
Live waveforms, 18  
Loading a test setup, 37  
Log view  
  save file, 24

## M

Map X drive (do after software install), 5  
Menus  
  Options, 13  
Minimum system requirements, 3  
My TekExpress  
  folder permissions (required), 6  
My TekExpress folder  
  files stored in, 27  
  mapping, 5

## N

Names, file extensions, 8  
New inbound rule wizard, 51  
Non-VISA, 14

## O

Opening a saved test setup, 37  
Options menu  
  Instrument control settings, 14  
  keep on top, 9  
Oscilloscope calibration, 32  
Oscilloscope compensation, 32

## P

Panels, 10

Pattern, 18  
Pattern type, 18  
PI commands  
  GetReportParameter, 66  
  Select Data Rate, 81–85  
  Set Pattern Type, 80  
  Set Record Length, 80  
Preferences menu, 25  
Preferences tab  
  send an Email, 23  
  setup panel, 23  
Program example, 58  
Programmatic interface, 51

## R

Recalling a test setup, 37  
Related documentation, 1  
Remote access firewall settings, 51  
Remote proxy object, 54  
Report contents, 30  
Report name, 28  
Report options, 28  
Report sections, 30  
Reports  
  receiving in email notifications, 15  
Reports panel, 10, 27  
Resource file, 9  
Results panel  
  summary of test results, 25  
  test name, 25

## S

Save log file, 24  
Saved sessions location (map drive- do after software install), 5  
Saving tests, 27  
Schematic button (DUT-instrument setup), 31  
Search for connected instruments, 14  
Select Data Rate (PI command), 81–85  
Selecting test report contents, 28  
Selecting tests, 19  
Serial, 14  
Server, 53

- Session folders and files, 27
- Set Pattern Type (PI command), 80
- Set Record Length (PI command), 80
- Set remote access, 51
- Setting up equipment, 31
- Setup
  - acquisition tab, 20
- Setup (required steps)
  - map My TekExpress to drive X:, 5
  - set My TekExpress folder permissions, 6
- Setup files, 37
- Setup panel
  - DUT parameter, 17
  - preferences tab, 17
  - test selection, 17
- Show acquire parameters, 20
- Signal conditioning
  - bandwidth, 18
  - filter, 18
- Software installation
  - TekExpress 100GBASE-SR4 Solution, 5
- Source, 18
- Status panel
  - log view, 24
  - message history, 24
  - test status tab, 24
- Support, 2
- System requirements, 3

## T

- Technical support, 2
- Tek Link, 14
- TekExpress
  - server, 51
  - client, 51
  - client requirements, 53
- Test Name
  - RIN, 20
  - RIN CW, 20
  - RINxOMA, 20
- Test reports, 30
- Test results
  - send by email, 15

- Test selection
  - optical tests, 19
  - RIN, 19
  - RINxOMA, 19
  - test description, 19
- Test selection controls, 19
- Test setup files, 27, 37
- Test setups
  - load, 37
  - open, 37
  - recalling, 37
- Test status
  - acquire status, 24
  - analysis status, 24
  - auto scroll, 24
- Test-related files, 27
- Tests
  - running, 35
  - selecting, 19

## U

- USB, 14
- User Comments
  - including in reports, 29

## V

- View a report, 30
- View application license agreement, 5
- View application software version, 5
- View optical modules button, 20
- VXI, 14

## W

- Waveform files
  - locating and storing, 27
- Wavelength, 18

## X

- X drive (location of saved sessions), 5

