TekExpress® CEI-VSR
Compliance and Debug Solution

Printable Application Help

077-0872-01

Tektronix

TekExpress® CEI-VSR
Compliance and Debug Solution

Printable Application Help

www.tektronix.com
077-0872-01

Contacting Tektronix

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

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

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

Table of Contents

Getting help and support

Getting started

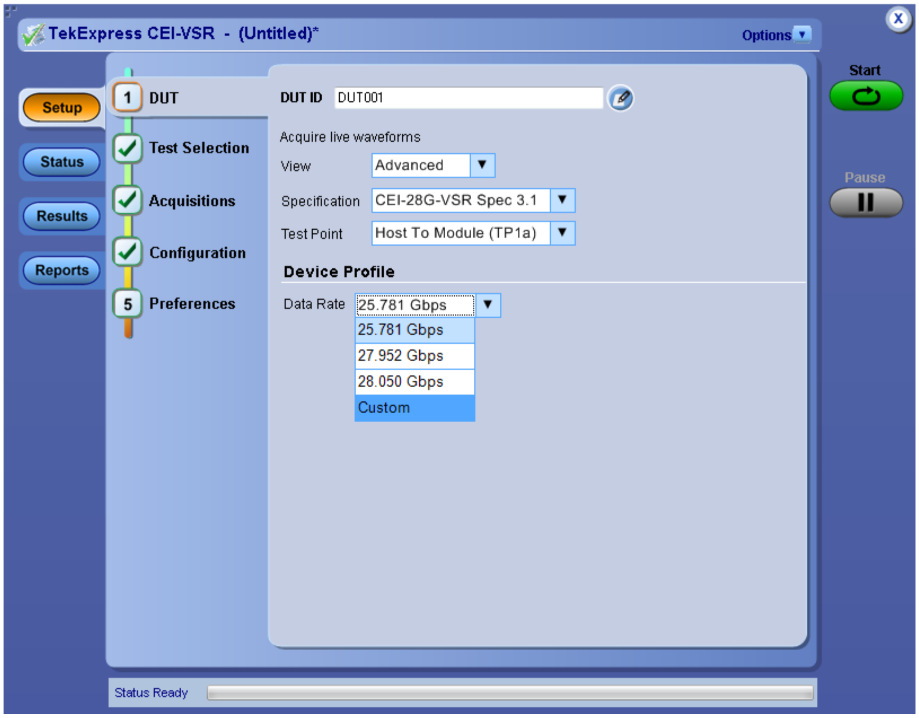
Operating basics

Pre-measurement calibration procedures

Saving and recalling test setup files

TekExpress programmatic interface

TekExpress CEI-VSR key features



DUT ID DUT001

Acquire live waveforms

View: Advanced

Specification: CEI-28G-VSR Spec 3.1

Test Point: Host To Module (TP1a)

Device Profile

Data Rate: 25.781 Gbps

Status Ready

Welcome to the TekExpress® CEI-VSR Automated Test Solution Software application (referred to as TekExpress CEI-VSR or CEI-VSR in the rest of the document). TekExpress CEI-VSR provides an automated, simple, and efficient way to test CEI-VSR Host to Module and Module to Host Interfaces to the requirements of the CEI-VSR specifications.

Key features and benefits of TekExpress CEI-VSR include:

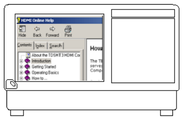
- Comprehensive test coverage; select or deselect individual tests
- Accurate and reliable results
- Automatically scans and finds the best CTLE
- Complete programmatic interface enables automation scripts to call CEI-VSR functions
- Custom data rate (14 to 28.6 Gbps) support in user defined mode
- Support of custom CTLE filters (SDLA is used for creating filter) for custom data rate

Getting help and support

Related documentation

The following manuals are available as part of the TekExpress® CEI-VSR Automated Test and Compliance Solution documentation set.

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 CEI-VSR software distribution (*CEI-VSR-Automated-Test-Solution-Software-Printable-Help-EN-US.pdf*).

See also

Technical support

Conventions used in help

Online Help uses the following conventions:

- The term “DUT” is an abbreviation for Device Under Test.
- The term “select” is a generic term that applies to the two 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 for more information.

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

General information

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

Application specific information

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

Getting started

Installing the software

Minimum system requirements

The following table shows the minimum system requirements to run TekExpress CEI-VSR.

Table 3: System requirements

Component	Description
Oscilloscope	Tektronix DSA8300 Digital Serial Analyzer *Required equipment*
Processor	Same as the oscilloscope
Operating System	Same as the oscilloscope: ■ Windows 7 *Windows 7 user account settings*
Memory	Same as the oscilloscope
Hard Disk	Same as the oscilloscope
Display	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 [1]
Firmware	■ TekScope 6.3.1.3 and later (for Windows 7)
Software	■ TekExpress Framework (version 3.0.x or later) installed. ■ Microsoft .NET 4.0 Framework ■ Opt JNB01 – 80SJNB Advanced (required for CEI-VSR testing) ■ Opt ADVTRIG – Advanced triggers with pattern sync (required for CEI-VSR testing) ■ Microsoft Internet Explorer 7.0 SP1 or later, or other Web browser for viewing reports. ■ Adobe Reader software 7.0 or later for viewing portable document format (PDF) files.
Other Devices	■ Microsoft compatible mouse or compatible pointing device. ■ Two USB ports (four USB ports recommended). ■ PCI-GPIB or equivalent interface for instrument connectivity [2].

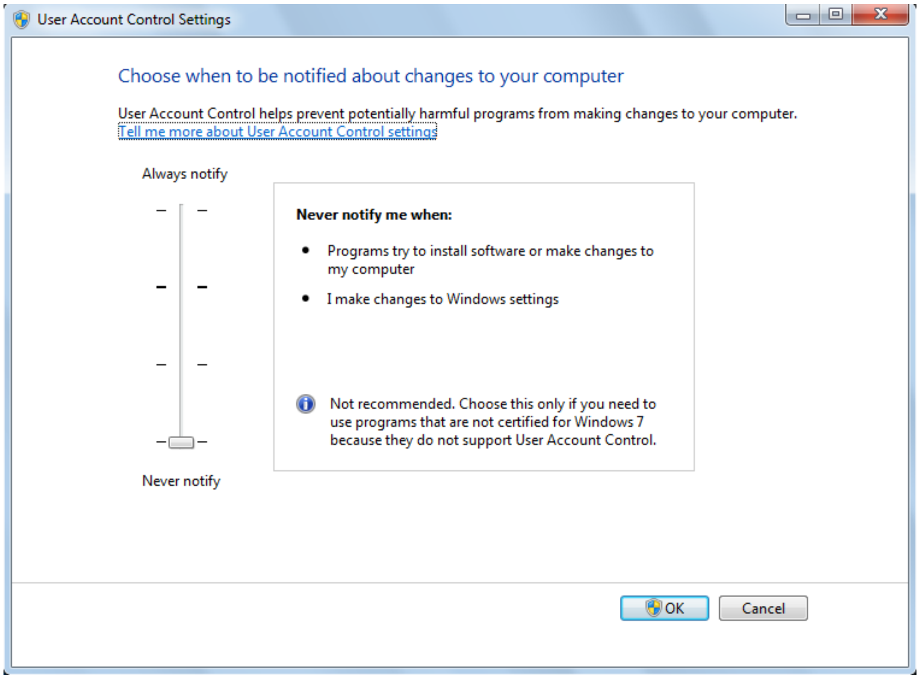
[1] If TekExpress is running on an instrument that has a video resolution less than 800x600, connect and configure a second monitor to the instrument.

[2] If TekExpress is installed on a Tektronix oscilloscope, TekExpress cannot use the virtual GPIB port to communicate with oscilloscope applications. If using external devices for instrument connectivity (such as USB-GPIB adapters or equivalent), enable the Talker Listener utility in the GPIB menu of the Tektronix DSA8300 oscilloscope.

Windows 7 user account settings

Windows 7 instruments need to have the User Account Control Settings set to **Never Notify**. To set User Account Control Settings:

1. Go to **Control Panel > User Accounts > Change User Account Control settings**.
2. set it to **Never Notify** as shown in the image.



User Account Control Settings

Choose when to be notified about changes to your computer

User Account Control helps prevent potentially harmful programs from making changes to your computer.

Tell me more about User Account Control settings

Always notify

Never notify me when:

- Programs try to install software or make changes to my computer
- I make changes to Windows settings

Not recommended. Choose this only if you need to use programs that are not certified for Windows 7 because they do not support User Account Control.

Never notify

See also. *Required equipment*

Required equipment

Table 4: Required equipment

Resource	Model supported
Sampling oscilloscope	Tektronix DSA8300 Digital Serial Analyzer FW Version: 6.3.1.3 or greater 80SJNB SW Version: 3.1.4.0 or greater
Sampling oscilloscope modules	80E09, 80E09B (Dual channel, 60 GHz, w/TDR) 80E10, 80E10B (Dual channel, 50 GHz) 80E11, 80E11B (Dual channel, 70 GHz w/TDR) 82A04, 82A04B (Phase reference)
Clock Recovery Module	CR286A
Module extender cables	80X01 (1 meter) 80X02 (2 meters)
Other accessories	80A08 accessory kit PSPL5510K Opt 6 dB attenuator 40 GHz (for vertical gain calibration)

See also. *Minimum system requirements*

Install the software

Use the following steps to obtain the latest CEI-VSR software from the Tektronix Web site and install on any compatible instrument running Microsoft Windows 7 (32-bit). See *Minimum system requirements* for details.

1. Close all applications (including the TekScope application).
2. Go to the www.tek.com Web site and locate the **Downloads** fields.
3. Enter **TekExpress CEI-VSR** in the *Model or Keyword* field, select **Software** from the *Select Download Type* list, and click **GO**.
4. Select the latest version of software. Follow instructions to download the software file.
5. Copy or download the CEI-VSR installer executable file to the oscilloscope.
6. Double-click the installer .exe file to extract the installation files and launch the InstallShield Wizard. Follow the on-screen instructions.

   Software is installed at `C:\Program Files\Tektronix\TekExpress\TekExpress CEI-VSR`
7. *Verify application installation*

See also. *Minimum system requirements*

Supported instruments

Required My TekExpress folder settings

Verify application installation

To verify the installation was successful:

1. Open the TekScope application.
2. Click the **Application** menu.
3. Verify that **CEI-VSR** is listed in the Application menu.
4. Click **CEI-VSR** to open the CEI-VSR application. Verify that the application opens successfully.

See also. *Activate the license*

Required My TekExpress folder settings

Activate the license

Activate the license using the **Option Installation** wizard in the TekScope application:

1. In the TekScope application menu bar, click **Utilities > Option Installation**.

   The TekScope Option Installation wizard opens.
2. Push the **F1** key on the oscilloscope keyboard to open the Option Installation help topic.
3. Follow the directions in the help topic to activate the license.

See also. *View version and license information*

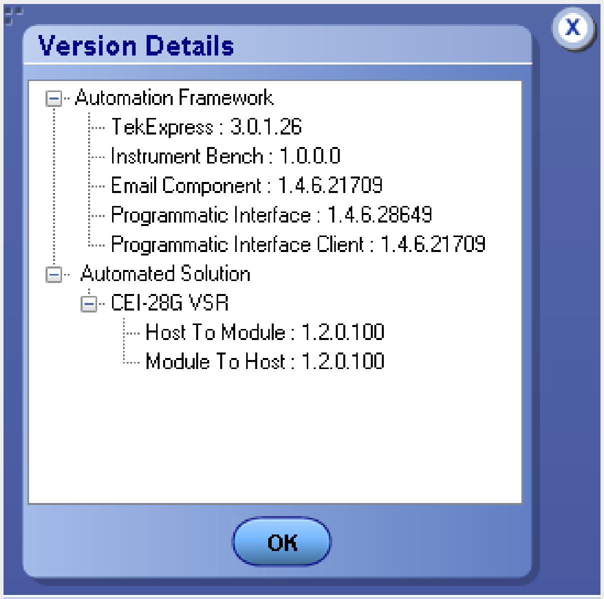
Required My TekExpress folder settings

View software version and license information

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 CEI-VSR:

1. In the CEI-VSR application, click the **Options** button and select **About TekExpress**.
2. Click the View Version Details link to view the version numbers of the installed test suites.



Version Details

- Automation Framework
  - TekExpress : 3.0.1.26
  - Instrument Bench : 1.0.0.0
  - Email Component : 1.4.6.21709
  - Programmatic Interface : 1.4.6.28649
  - Programmatic Interface Client : 1.4.6.21709
- Automated Solution
  - CEI-28G VSR
    - Host To Module : 1.2.0.100
    - Module To Host : 1.2.0.100

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

To view license and option key information:

1. In the TekScope application, select **Help > About DSA8300**.
2. Scroll through the **Options** section list to locate **CEI-VSR: TekExpress CEI-VSR**.
3. To view the Option key, look below the **Options** list.

See also. *Activate the license*

Options menu

Required My TekExpress folder settings

Before you run tests for the first time, do the following:

1. *Map My TekExpress folder to Drive X*
2. *Set My TekExpress folder permissions*

See also. *Application directories and usage*

File name extensions

Map My TekExpress folder to drive X

The first time you run TekExpress CEI-VSR, it creates the following folders on the oscilloscope:

- `\My Documents\My TekExpress\CEI-VSR`
- `\My Documents\My TekExpress\CEI-VSR\Untitled Session`

You need to map the shared **My TekExpress** folder as drive **X:** on the instrument running the CEI-VSR application. CEI-VSR uses this shared folder to save session waveform files and for other application file transfer operations.

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*

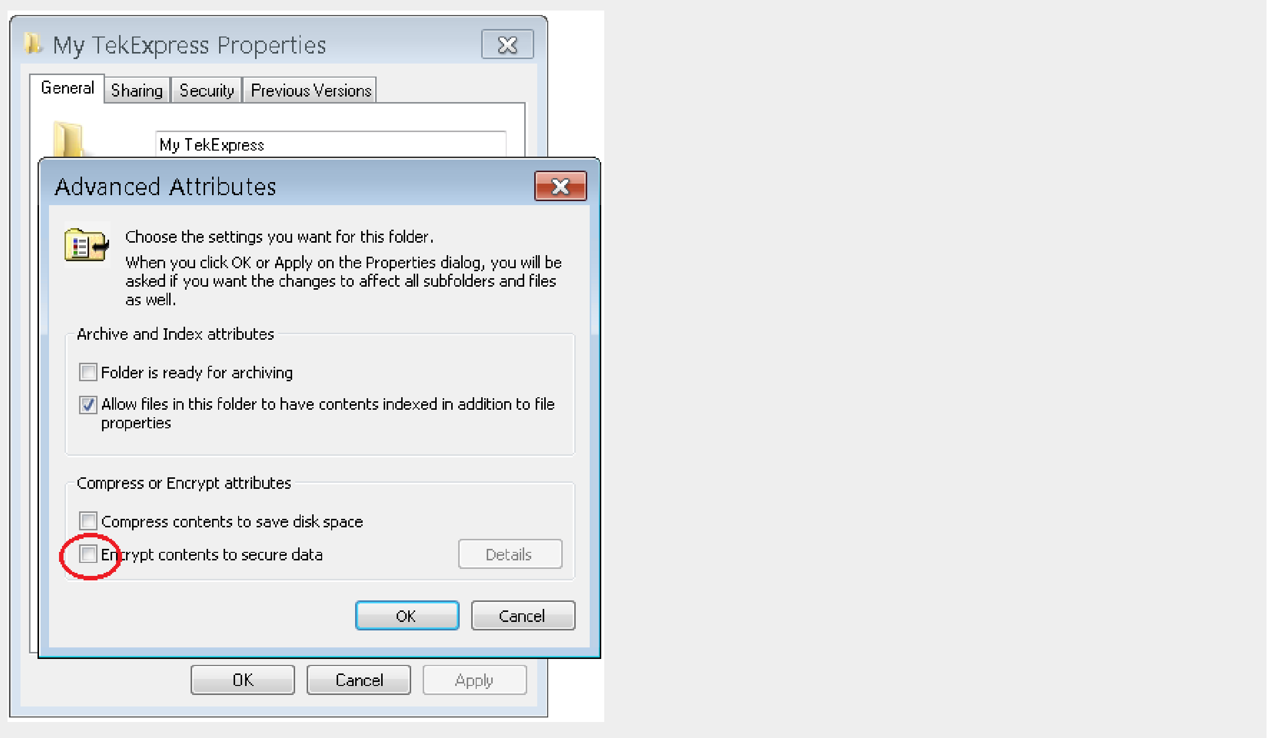
Application directories and usage

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.



My TekExpress

Choose the settings you want for this folder.

When you click OK or Apply on the Properties dialog, you will be asked if you want the changes to affect all subfolders and files as well.

Archive and Index attributes

□ Folder is ready for archiving

☑ Allow files in this folder to have contents indexed in addition to file properties

Compress or Encrypt attributes

□ Compress contents to save disk space

□ Encrypt contents to secure data

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*

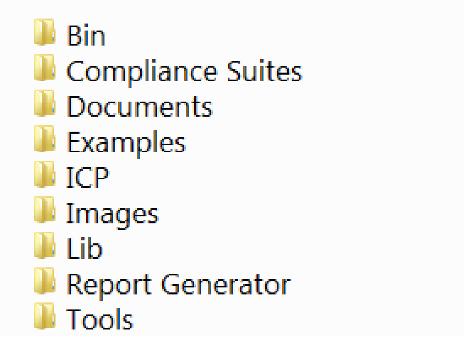
Application directories and usage

File name extensions

Application directories and their contents

TekExpress CEI-VSR application. The TekExpress CEI-VSR application files are installed at the following location:

```
C:\Program Files\Tektronix\TekExpress\TekExpress CEI-VSR
```



Bin
Compliance Suites
Documents
Examples
ICP
Images
Lib
Report Generator
Tools

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

Table 5: Application directories and usage

Directory names	Usage
Bin	Contains CEI-VSR application libraries
Compliance Suites	Contains compliance-specific files
Documents	Contains the technical documentation for the CEI-VSR application
Examples	Contains various support files
ICP	Contains instrument and CEI-VSR application-specific interface libraries
Images	Contains images of the CEI-VSR application
Lib	Contains utility files specific to the CEI-VSR application
Report Generator	Contains style sheets for report generation
Tools	Contains instrument and CEI-VSR application-specific files

See also. *View test-related files*

File name extensions

File name extensions

The TekExpress CEI-VSR 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
.wfm	Test waveform files
.mht	Test result reports (default) Test reports can also be *saved in HTML format*

File name extension	Description
.flt	Filter files
.xslt	Style sheet used to generate reports

See also. *View test-related files*

Application directories and their contents

Where test files are stored

When you launch TekExpress CEI-VSR for the first time, it creates the following folders on the oscilloscope:

- `\My Documents\My TekExpress\CEI-VSR`
- `\My Documents\My TekExpress\CEI-VSR\Untitled Session`

Every time you launch TekExpress CEI-VSR, an `Untitled Session` folder is created in the `CEI-VSR` folder. The `Untitled Session` folder is automatically deleted when you exit the `CEI-VSR` application. To preserve your test session files, save the test setup before exiting the TekExpress application.

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

See also. *Map My TekExpress folder to Drive X*

Set My TekExpress folder permissions

Application directories and usage

File name extensions

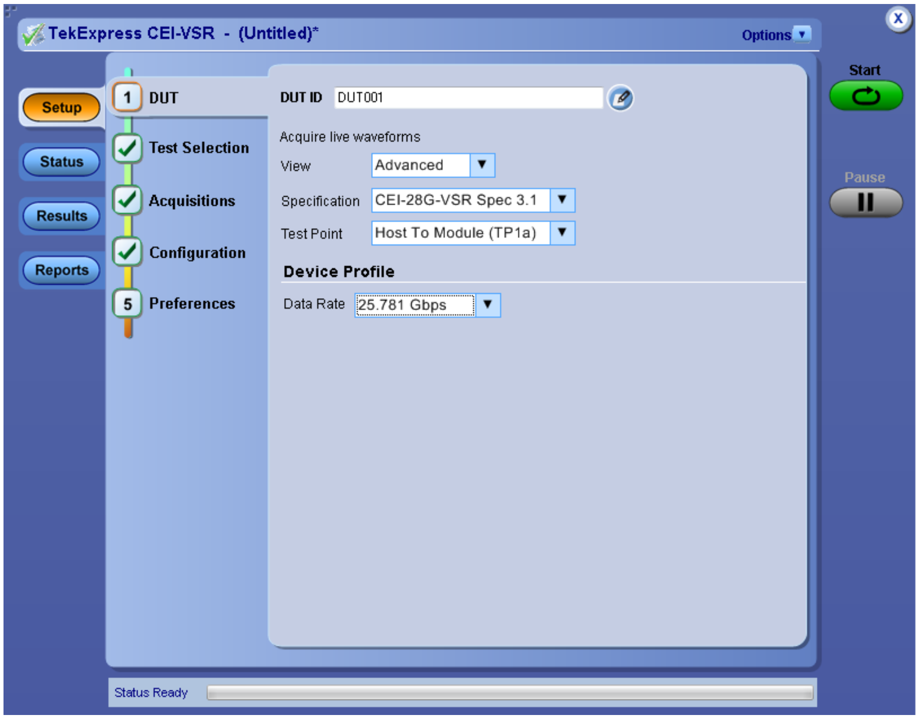
Operating basics

Run the application

To launch the TekExpress CEI-VSR application, do either of the following:

- Select **Application > CEI-VSR** from the TekScope menu.
- Double-click any saved TekExpress CEI-VSR session file (<file name>.TekX).

The oscilloscope opens the TekExpress CEI-VSR application:



DUT ID DUT001

Acquire live waveforms

View: Advanced

Specification: CEI-28G-VSR Spec 3.1

Test Point: Host To Module (TP1a)

Device Profile

Data Rate: 25.781 Gbps

Status Ready

When you first run the application after installation, the application checks for a file called `Resources.xml` located in the `C:\Users\<username>\My TekExpress\CEI-VSR` folder. The Resources.xml file gets mapped to the `X:` drive when the application launches. Session files are then stored inside the `X:\CEI-VSR` folder.

The Resources.xml file contains information about available network-connected instruments. If this file is not found, the application runs an instrument discovery program to detect connected instruments before launching CEI-VSR.

NOTE. *Do the steps in the Required My TekExpress folder settings topic before running tests with the CEI-VSR application for the first time.*

To keep the CEI-VSR application window on top, select **Keep On Top** from the CEI-VSR *Options menu*. If the application goes behind the oscilloscope application, click **Application > CEI-VSR** to move the application to be in front.

See also

Required My TekExpress folder settings

Activate the license

Exit the application

Application controls

Application panel overview

Exit the application

To exit the application, click X on the application title bar. Follow on-screen prompts to save any unsaved session, save test setup files, or exit the application.

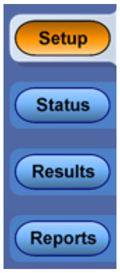
NOTE. *Using other methods to exit the application can result in abnormal termination of the application.*

Application controls and menus

Global application controls

Application controls.

Table 6: Application controls descriptions

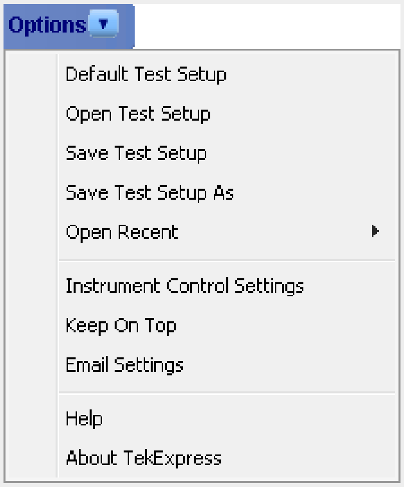
Item	Description
Options menu 	Menu to display global application controls.
Test Panel buttons  	Controls that open panels for configuring test settings and options.
Start / Stop button 	Use the Start button to start the test run of the measurements in the selected order. If prior acquired measurements have not been cleared, the 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.
Pause / Continue button 	Use the Pause button to temporarily interrupt the current acquisition. When a test is paused, the button name changes to “Continue.”
Clear button 	Use the Clear button to clear all existing measurement results. Adding or deleting a measurement, or changing a configuration parameter of an existing measurement, also clears measurements. This is to prevent the accumulation of measurement statistics or sets of statistics that are not coherent. This button is available only on the *Results panel*.
Application window move icon 	Place the cursor over the three-dot pattern in the upper left corner of the application window. When the cursor changes to a hand, drag the window to the desired location.

See also. *Application panel overview*

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

The Options menu has the following selections:

Options menu



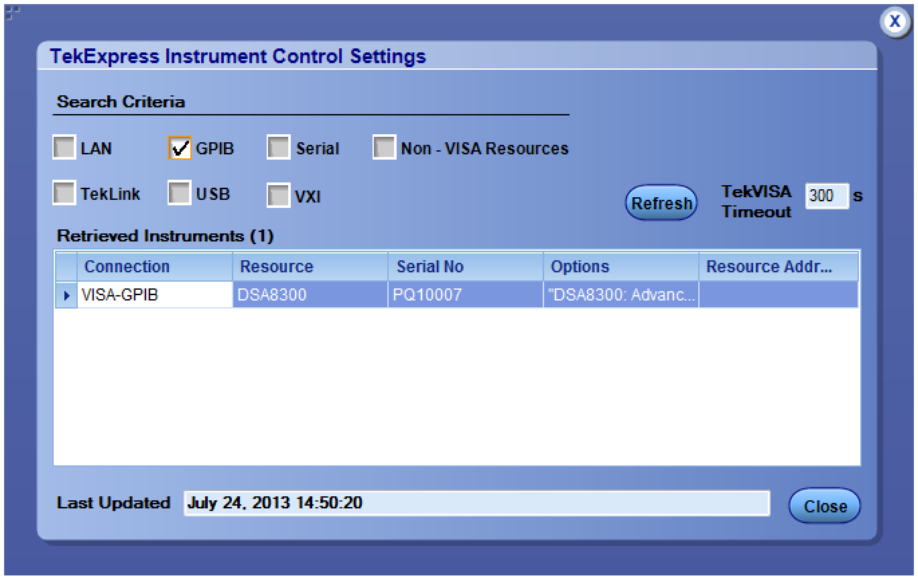
Menu	Function
Default Test Setup	Opens an untitled test setup with defaults selected
Open Test Setup	Opens a saved test setup
Save Test Setup	Saves the current test setup selections
Save Test Setup As	Creates a new test setup based on an existing one
Open Recent	Displays a menu of recently opened test setups to select from
Instrument Control Settings	Detects, lists, and refreshes the connected instruments found on specified connections (LAN, GPIB, USB, and so on)
Keep On Top	Keeps the TekExpress CEI-VSR application on top of other open windows on the desktop
Email Settings	Use to configure email options for test run and results notifications
Help	Displays the TekExpress CEI-VSR help
About TekExpress	■ Displays application details such as software name, version number, and copyright ■ Provides access to *license information* for your CEI-VSR installation ■ Provides a link to the Tektronix Web site

See also. *Application controls*

TekExpress instrument control settings.

Instrument control settings. Use the TekExpress Instrument Control Settings dialog box to search for and list the connected resources (instruments) found on specified connections (LAN, GPIB, USB, and so on), and each instruments connection information.

Access this dialog box from the **Options** menu.



TekExpress Instrument Control Settings

Search Criteria

□ LAN ☑ GPIB □ Serial □ Non - VISA Resources

□ TekLink □ USB □ VXI

TekVISA Timeout 300 s

Retrieved Instruments (1)

Connection	Resource	Serial No	Options	Resource Addr...
VISA-GPIB	DSA8300	PQ10007	"DSA8300: Advanc...	

Last Updated July 24, 2013 14:50:20

Use the Instrument Control Settings feature to *search for connected instruments* and view instrument connection details. Connected instruments displayed here can be selected for use under Global Settings in the test configuration section.

See also. *Options menu overview*

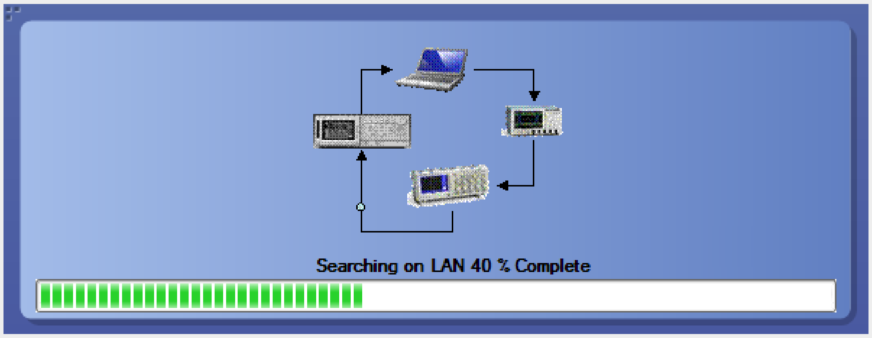
View connected instruments. Use the Instrument Control Settings dialog box to view or search for connected instruments required for the tests. The application uses TekVISA to discover the connected instruments on all selected connection types.

NOTE. *The correct instruments for the current test setup must be connected and recognized by CEI-VSR before running tests.*

To refresh the list of connected instruments:

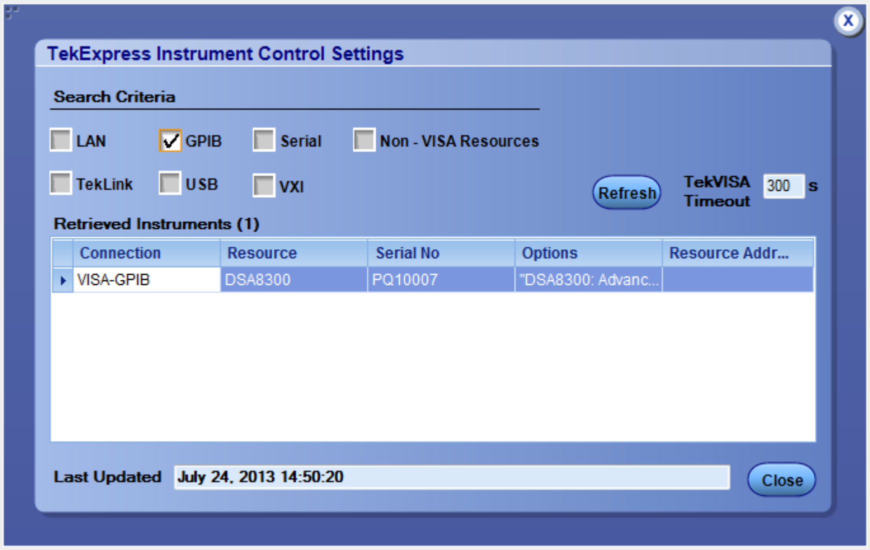
1. From the Options menu, select **Instrument Control Settings**.
2. In the **Search Criteria** section of the Instrument Control Settings dialog box, select the connection types of the instruments for which to search.

   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.



Searching on LAN 40 % Complete

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



TekExpress Instrument Control Settings

Search Criteria

□ LAN ☑ GPIB □ Serial □ Non - VISA Resources

□ TekLink □ USB □ VXI

TekVISA Timeout 300 s

Retrieved Instruments (1)

Connection	Resource	Serial No	Options	Resource Addr...
VISA-GPIB	DSA8300	PQ10007	"DSA8300: Advanc...	

Last Updated July 24, 2013 14:50:20

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.

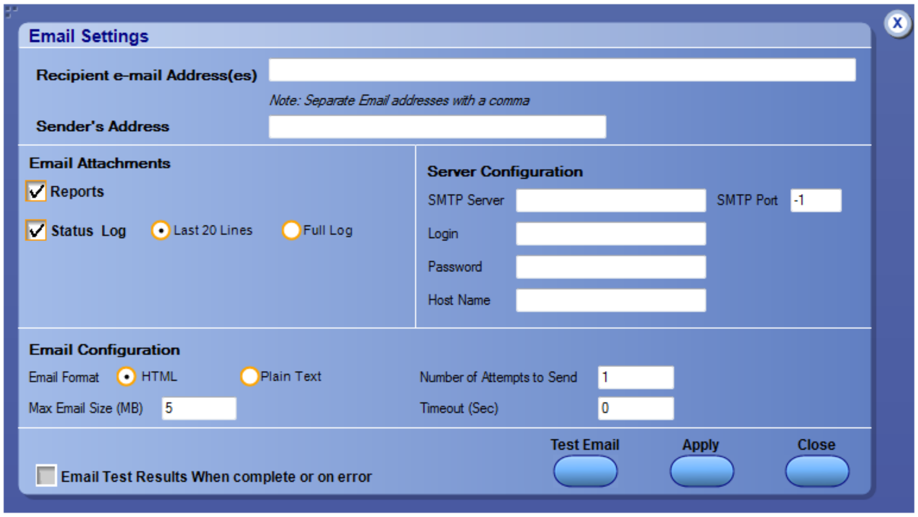
See also. *Configuration test parameters*

Equipment connection setup

Email settings. Use the Email Settings utility to *configure email notifications* to receive notifications when a test completes, produces an error, or fails. Select the type of test session information to include in the notification, such as test reports and test logs, the email message format, and the email message size limit.

Select **Options > Email Settings** to open this dialog box.

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



Email Settings

Recipient e-mail Address(es)

Note: Separate Email addresses with a comma

Sender's Address

Email Attachments

☑ Reports

☑ Status Log ● Last 20 Lines ○ Full Log

Server Configuration

SMTP Server

SMTP Port -1

Login

Password

Host Name

Email Configuration

Email Format ● HTML ○ Plain Text

Max Email Size (MB) 5

Number of Attempts to Send 1

Timeout (Sec) 0

□ Email Test Results When complete or on error

See also. *Configure email settings*

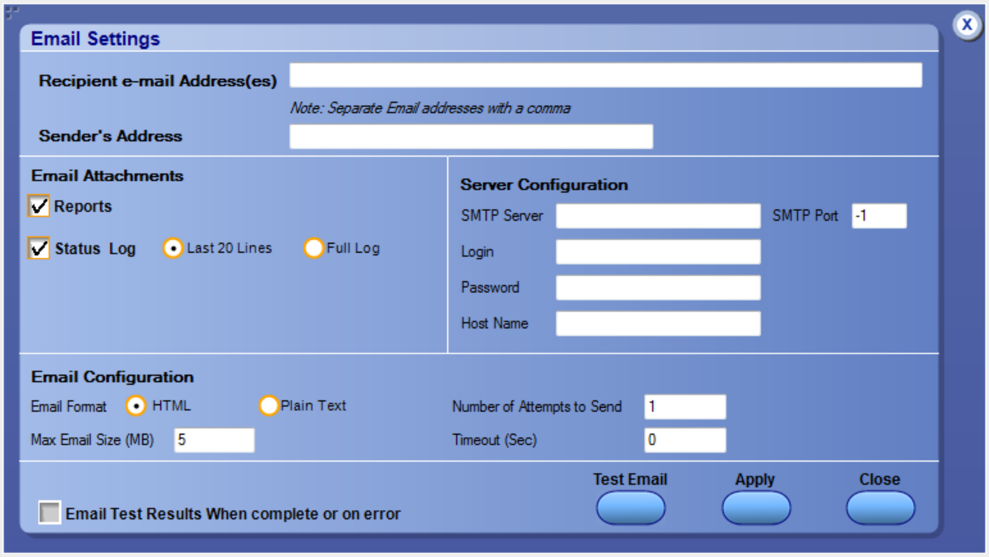
Options menu

Select test notification preferences

Configure email settings. Use the Email Settings dialog box to be notified by email when a test completes, fails, or produces an error:

1. Select **Options > Email Settings** to open the Email Settings dialog box.

Email settings



Email Settings

Recipient e-mail Address(es)

Note: Separate Email addresses with a comma

Sender's Address

Email Attachments

☑ Reports

☑ Status Log ● Last 20 Lines ○ Full Log

Server Configuration

SMTP Server

SMTP Port -1

Login

Password

Host Name

Email Configuration

Email Format ● HTML ○ Plain Text

Max Email Size (MB) 5

Number of Attempts to Send 1

Timeout (Sec) 0

☐ Email Test Results When complete or on error

2. (Required) For Recipient email Address(es), enter one or more email addresses to which to send the test notification. To include multiple addresses, separate the addresses with commas.

3. (Required) For Sender's Address, enter the email address used by the instrument. This address consists of the instrument name followed by an underscore followed by the instrument serial number, then the @ symbol and the email server used. 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:

   - Select the message file format to send: HTML (the default) or plain text.
   - Enter a maximum file size for the email message. Messages with attachments larger than this limit will not be sent. The default is 5 MB.
   - Enter the number in the Number of Attempts to Send field, to limit the number of attempts that the system makes to send a notification. The default is 1. You can also specify a timeout period.

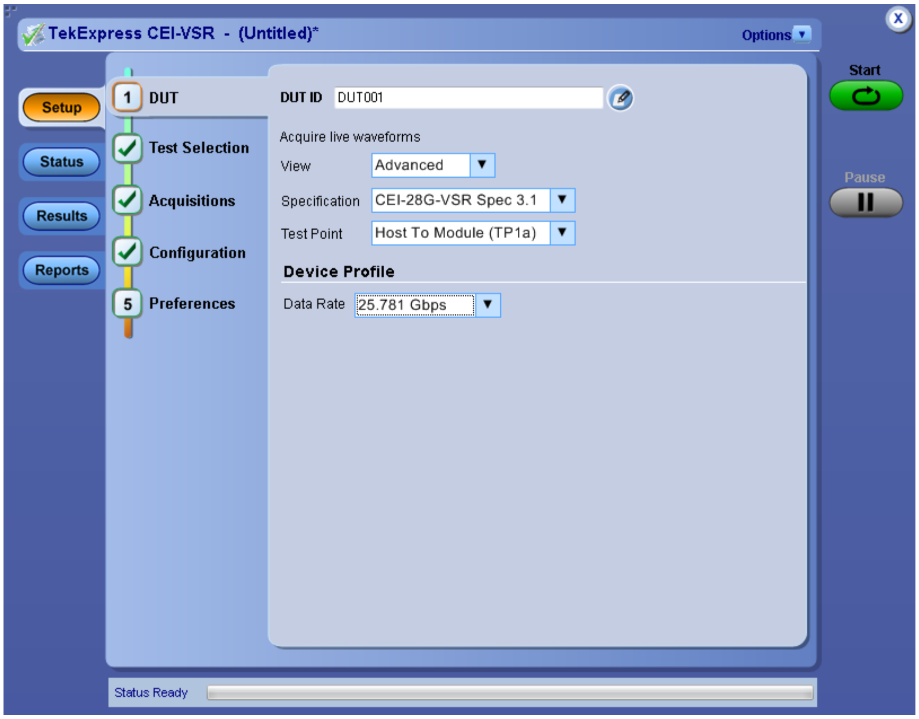
7. Select the **Email Test Results When complete or on error** check box. Use this check box to quickly enable or disable email notifications.

8. To test your email settings, click **Test Email**.

9. To apply your settings, click **Apply**.
10. Click **Close** when finished.

Application test panels

Application panels overview. TekExpress CEI-VSR uses panels to group related configuration, test, and results settings. Click on a 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.



TekExpress CEI-VSR - (Untitled)*

DUT ID DUT001

Acquire live waveforms

View: Advanced

Specification: CEI-28G-VSR Spec 3.1

Test Point: Host To Module (TP1a)

Device Profile

Data Rate: 25.781 Gbps

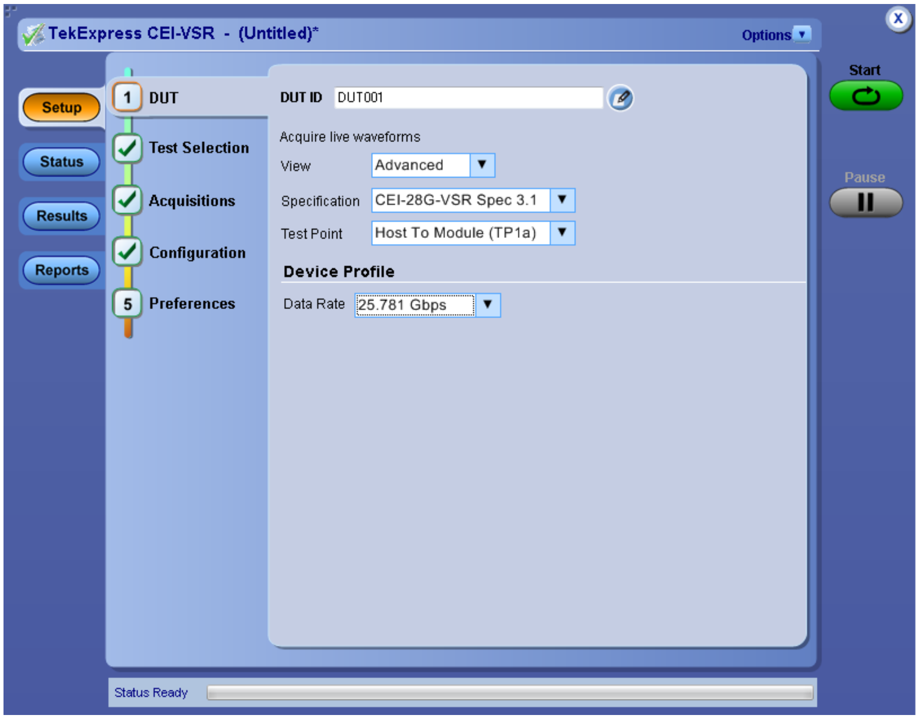
Status Ready

Table 7: Application panels overview

Panel Name	Purpose
Setup	The Setup panel shows the test setup controls. Click the **Setup** button to open this panel. Use this panel to: ■ *Select DUT parameters*. ■ *Select the test(s)*. ■ *Set acquisitions parameters* for selected tests. ■ *Configuration test parameters* ■ *Select test notification preferences*.
Status	View the progress and analysis status of the selected tests, and view test logs.
Results	View a summary of test results and select result viewing preferences.
Reports	Browse for reports, save reports 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*

Setup control overview. The Setup panel contains sequentially ordered tabs that help guide you through a typical test setup and execution process. Click on a tab to open the associated controls.



DUT ID DUT001

Acquire live waveforms

View: Advanced

Specification: CEI-28G-VSR Spec 3.1

Test Point: Host To Module (TP1a)

Device Profile

Data Rate: 25.781 Gbps

Status Ready

The tabs on this panel are:

DUT: *Set the DUT parameters*

Test Selection: *Select test(s)*

Configuration: *Set the configuration tab parameters*

Acquisitions: *Select acquisition parameters*

Preferences: *Select test fail notification preferences*

Set DUT parameters. Use the DUT tab to select parameters for the device under test. The settings are global and apply to all tests for the current session. DUT settings also affect the list of available tests in the Test Selection tab.

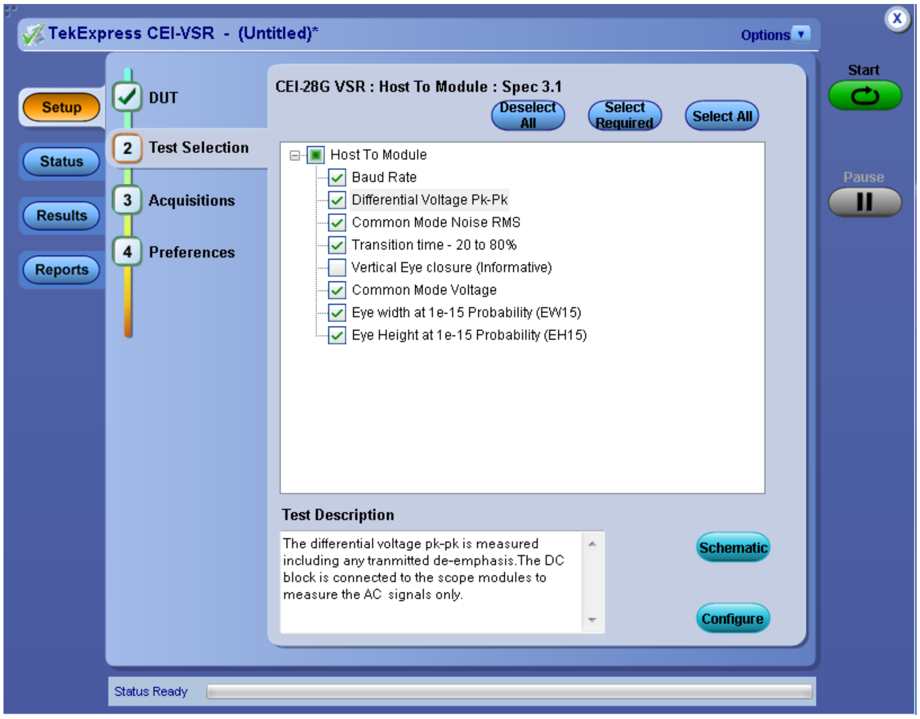
Click **Setup > DUT** to access the DUT parameters:

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 a 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 *Select report options*.)
Acquire live waveforms	Acquire active signals from the DUT for measurement and analysis.
View	Select Compliance or Advanced from the drop-down list.
Specification	Lists the supported CEI-VSR test specification(s).
Test Point	Sets the DUT device type to test (Device to Module or Host to Module).
Device Profile	
Data Rate	Select the data rate of the DUT to be tested from the drop-down list. ■ If custom is selected, enter the data rate (supported range is 14 to 28.6 Gbps)

See also. *Select a test*

Select tests. Use the **Test Selection** tab to select CEI-VSR tests.



CEI-28G VSR : Host To Module : Spec 3.1

- ☑ Host To Module
  - ☑ Baud Rate
  - ☑ Differential Voltage Pk-Pk
  - ☑ Common Mode Noise RMS
  - ☑ Transition time - 20 to 80%
  - □ Vertical Eye closure (Informative)
  - ☑ Common Mode Voltage
  - ☑ Eye width at 1e-15 Probability (EW15)
  - ☑ Eye Height at 1e-15 Probability (EH15)

Test Description

The differential voltage pk-pk is measured including any tranmitted de-emphasis.The DC block is connected to the scope modules to measure the AC signals only.

Table 9: Test Selection tab settings

Setting	Description
Deselect All, **Select Required**,**Select All** buttons	Deselect or select tests in the list. The Select Required button selects all required tests when in Compliance test mode.
Tests	Click on a test to select or unselect. Highlight a test to show details in the Test Description pane. All required tests are selected when in Compliance test mode. Measurements are grouped according to standard specifications such as Host-to-Module and Module-to-Host.
Test Description	Shows a brief description of the highlighted test in the Test field.
Schematic button	Shows an equipment and test fixture setup schematic for the selected test. Use to set up the equipment and fixtures or to verify the setup before running the test.

NOTE. *All informative tests except Vertical Eye Closure (VEC) are selected by default.*

Host-to-Module

Includes Baud Rate, Differential Voltage pk-pk, Common Mode Noise rms, Transition Time 20/80%, Common Mode Voltage, Vertical Eye Closure (VEC), Eye width at 1e–15 probability (EW15) and Eye height at 10e–15 probability (EH15).

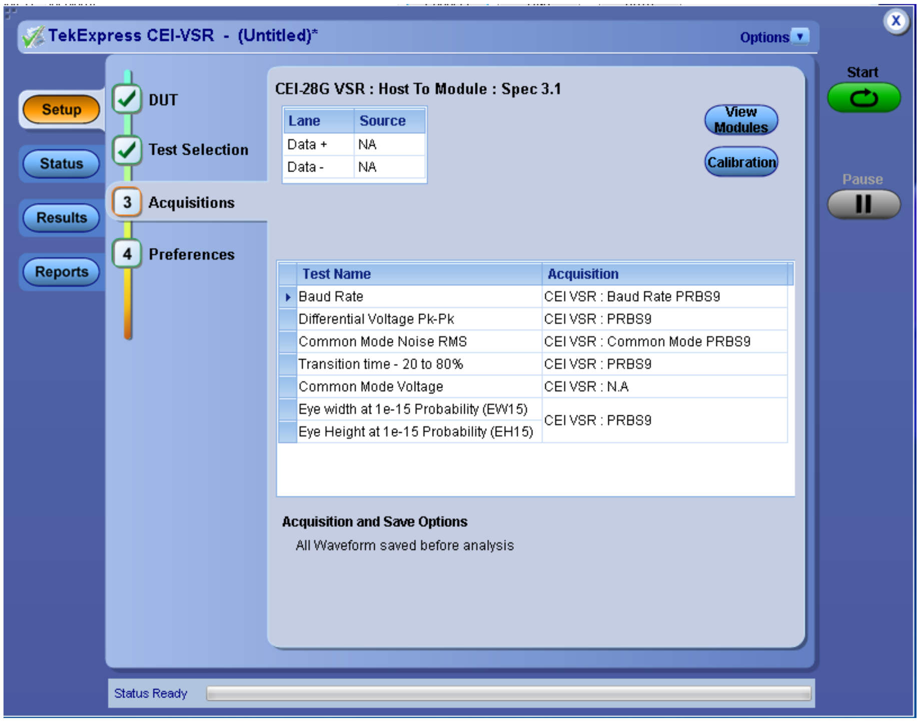
Module-to-Host

Includes Baud Rate, Differential Voltage pk-pk, Common Mode Noise rms, Transition Time 20/80%, Vertical Eye Closure (VEC), Eye width at 1e–15 probability (EW15) and Eye height at 10e–15 probability (EH15).

See also. *Set acquisition parameters*

Set acquisition parameters. Use the **Acquisition** tab in the Setup panel to view test acquisition parameters. You also use this tab to load prerecorded (saved) test session waveform files on which to run tests.

Contents displayed on this tab depend on the DUT type and selected tests.



CEI-28G VSR : Host To Module : Spec 3.1

Lane	Source
Data +	NA
Data -	NA

Test Name	Acquisition
Baud Rate	CEI VSR : Baud Rate PRBS9
Differential Voltage Pk-Pk	CEI VSR : PRBS9
Common Mode Noise RMS	CEI VSR : Common Mode PRBS9
Transition time - 20 to 80%	CEI VSR : PRBS9
Common Mode Voltage	CEI VSR : N.A
Eye width at 1e-15 Probability (EW15)	CEI VSR : PRBS9
Eye Height at 1e-15 Probability (EH15)	

Acquisition and Save Options

All Waveform saved before analysis

Status Ready

***NOTE.** CEI-VSR acquires all waveforms required by each test group before performing analysis.*

Table 10: Acquisitions tab settings

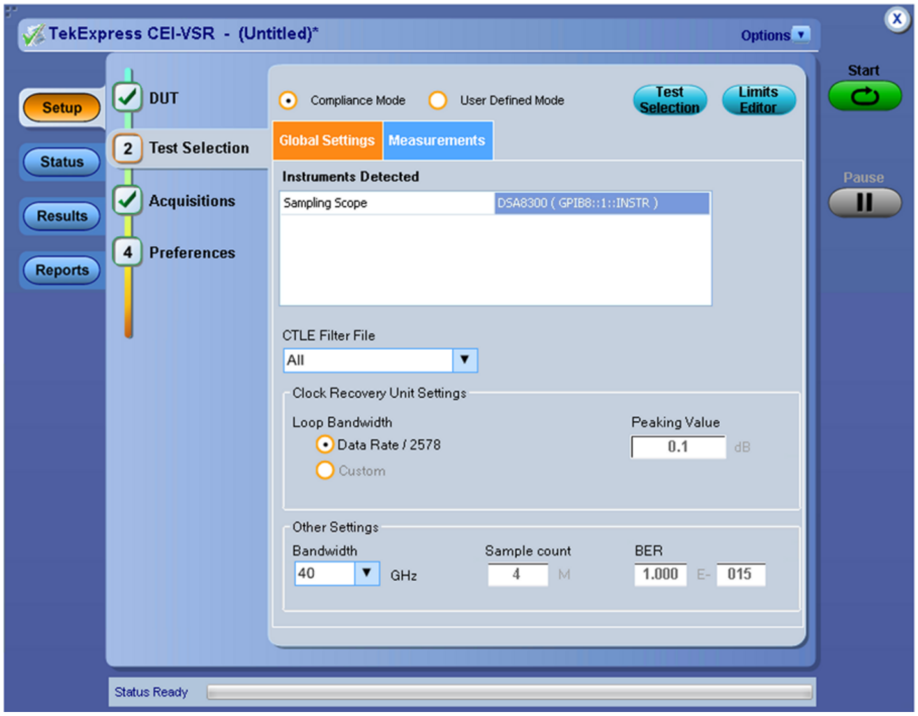
Setting	Description
View Modules button	Shows the detected modules that are installed in the instrument.
Calibration button	Shows the results of the most recent instrument calibration. Use the Calibrations dialog box to view the status of Deskew, external attenuation, scope calibration and instrumentation noise. Update these parameters by clicking the associated Refresh or Measure button. *Calibration guidelines*

CEI-VSR saves all acquisition waveforms to files by default. Waveforms are saved to a folder that is unique to each session (a session starts when you click the Start button). The folder path is `X:\CEI-VSR\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.

When the session is saved, content is moved to that session folder and the “Untitled Session” name is replaced by the session name.

Set the configuration tab parameters. Use the **Configuration** tab to set and view global instrument parameters for the selected tests. Which fields are available to edit depends on the selected test mode (Compliance or User Defined) as set in this tab or the DUT tab.

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



● Compliance Mode ○ User Defined Mode

Instruments Detected

Sampling Scope	DSA8300 (GPIB8::1::INSTR)

CTLE Filter File: All

Clock Recovery Unit Settings

Loop Bandwidth: ● Data Rate / 2578 ○ Custom

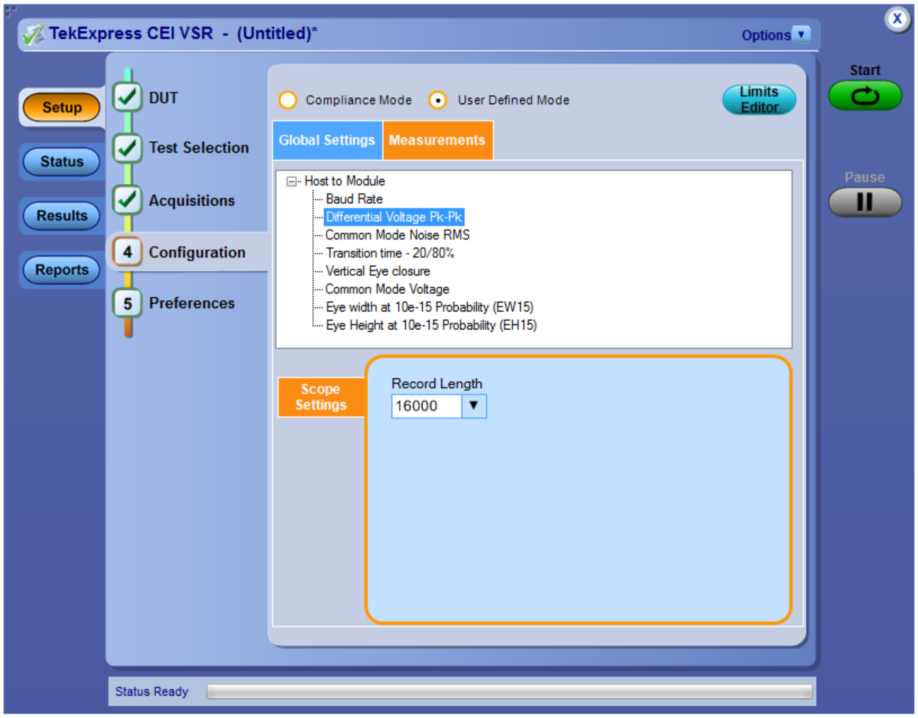
Peaking Value: 0.1 dB

Other Settings

Bandwidth: 40 GHz

Sample count: 4 M

BER: 1.000 E- 015



○ Compliance Mode ● User Defined Mode

- Host to Module
  - Baud Rate
  - Differential Voltage Pk-Pk
  - Common Mode Noise RMS
  - Transition time - 20/80%
  - Vertical Eye closure
  - Common Mode Voltage
  - Eye width at 10e-15 Probability (EW15)
  - Eye Height at 10e-15 Probability (EH15)

Record Length 16000

Status Ready

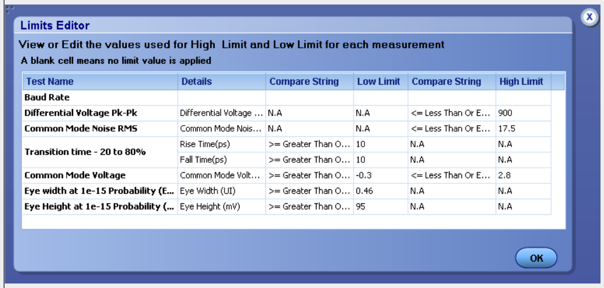
Figure 1: Configuration tab: Measurements

See also. *Configuration tab: Global Settings parameters*

Configuration tab: Measurements parameters

Configuration tab: Global settings parameters. The following table lists the Configuration tab settings and parameters. Fields shown on this tab can change depending on selected items.

Table 11: Configuration tab Global Settings

<table>
<tr><th>Control</th><th>Description</th></tr>
<tr><td>Test Mode</td><td>Determines whether test parameters are in compliance or can be edited.
- Compliance: Most test parameter values cannot be edited. Select this for compliance mode testing.
- User Defined: Enables editing of parameters.</td></tr>
<tr><td><b>Limits Editor</b> button</td><td>Opens the Limits Editor dialog box.
In User Defined Mode, use the Limits Editor to edit individual test limit settings.

Limits Editor
View or Edit the values used for High Limit and Low Limit for each measurement
A blank cell means no limit value is applied
<table>
<tr><th>Test Name</th><th>Details</th><th>Compare String</th><th>Low Limit</th><th>Compare String</th><th>High Limit</th></tr>
<tr><td>Baud Rate</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Differential Voltage Pk-Pk</td><td>Differential Voltage ...</td><td>N.A</td><td>N.A</td><td>&lt;= Less Than Or E...</td><td>900</td></tr>
<tr><td>Common Mode Noise RMS</td><td>Common Mode Nois...</td><td>N.A</td><td>N.A</td><td>&lt;= Less Than Or E...</td><td>17.5</td></tr>
<tr><td>Transition time - 20 to 80%</td><td>Rise Time(ps)</td><td>&gt;= Greater Than O...</td><td>10</td><td>N.A</td><td>N.A</td></tr>
<tr><td></td><td>Fall Time(ps)</td><td>&gt;= Greater Than O...</td><td>10</td><td>N.A</td><td>N.A</td></tr>
<tr><td>Common Mode Voltage</td><td>Common Mode Volt...</td><td>&gt;= Greater Than O...</td><td>-0.3</td><td>&lt;= Less Than Or E...</td><td>2.8</td></tr>
<tr><td>Eye width at 1e-15 Probability (E...</td><td>Eye Width (UI)</td><td>&gt;= Greater Than O...</td><td>0.46</td><td>N.A</td><td>N.A</td></tr>
<tr><td>Eye Height at 1e-15 Probability (...</td><td>Eye Height (mV)</td><td>&gt;= Greater Than O...</td><td>95</td><td>N.A</td><td>N.A</td></tr>
</table>

To edit a value, click that field and either select from the displayed list or enter a new value. Use scroll bars to view all available fields.
In Compliance Mode, use the Limits Editor to view the measurement high and low limits used for selected tests. You cannot edit values while in Compliance mode.</td></tr>
<tr><td>Instruments Detected</td><td>Displays a list of the connected instruments found during the instrument discovery. Instrument types include equipment such as sampling oscilloscopes. Select <b>Options > Instrument Control Settings</b> to <i>refresh the connected instrument list</i>.</td></tr>
<tr><td>CTLE Filter File (1-8)</td><td>Select the CTLE filter appropriate for your measurement.
1. For data rates 25.78125 Gbps, 27.952 Gbps, 25.05 Gbps
- CTLE filters given in the specification are provided as part of application
- Create custom filters using SDLA visualizer and provide them as custom filters
2. For custom data rates
- Create custom filters using SDLA visualizer and provide them as custom filters</td></tr>
<tr><td>Clock Recovery Units Settings</td><td>Sets the Loop bandwidth and Peaking units for the Tektronix CR286A Clock Recovery module. Default values set to meet compliance testing.</td></tr>
</table>

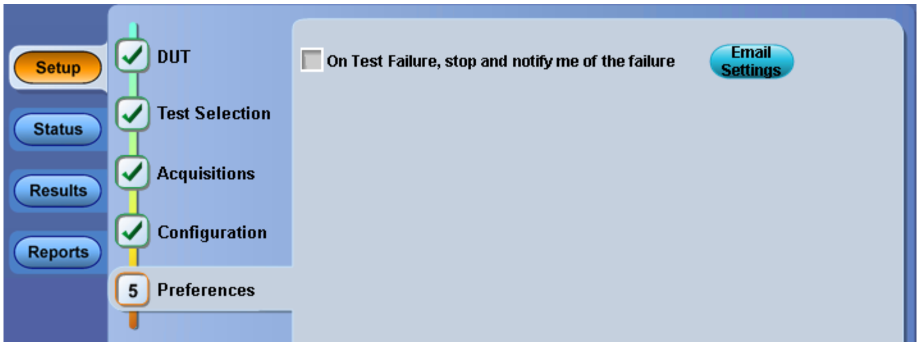
Control	Description
Other Settings	Bandwidth: Sets the oscilloscope bandwidth parameter. BER: Sets the BER parameter for the 80SJNB software. Default value is 1e-15. Sample count: Sets the number of samples to be acquired using the 80SJNB software. Maximum value is 30M and the default value is 4M.

See also. *About acquisitions*

Configuration tab: Measurement parameters. Lists all selected tests. All measurements have either pattern type or record length configuration parameters.

See also. *Set acquisition parameters*

Preferences tab. Use the Preferences tab to set the application action when a test measurement fails.



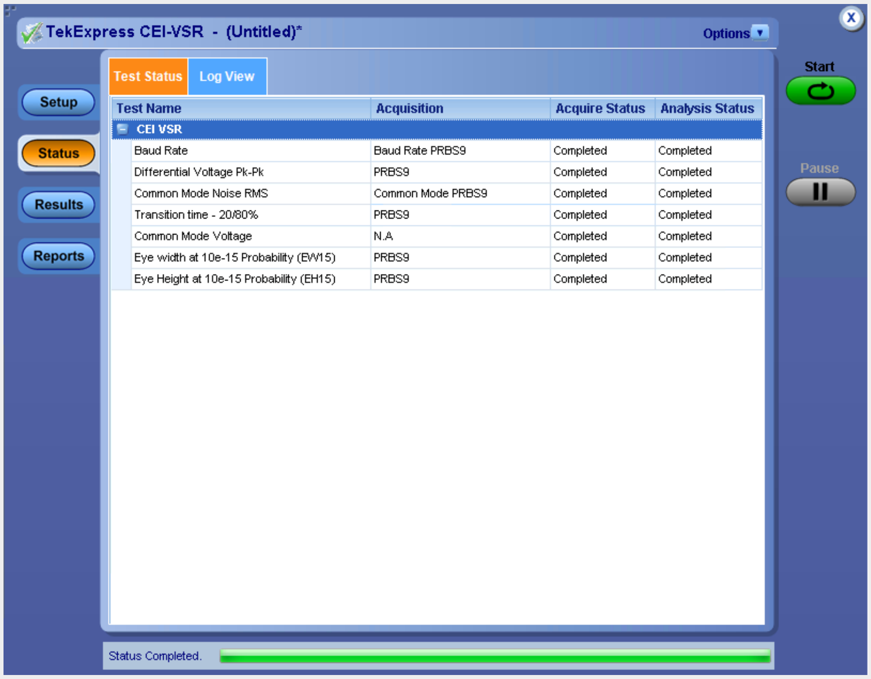
□ On Test Failure, stop and notify me of the failure

Table 12: Preferences tab settings

Setting	Description
On Test Failure, stop and notify me of the failure	Stops the test and sends an email when a test fails. Click Email Settings to verify that Email Test Results when complete or on error is selected, and to verify the address to which the email is sent.

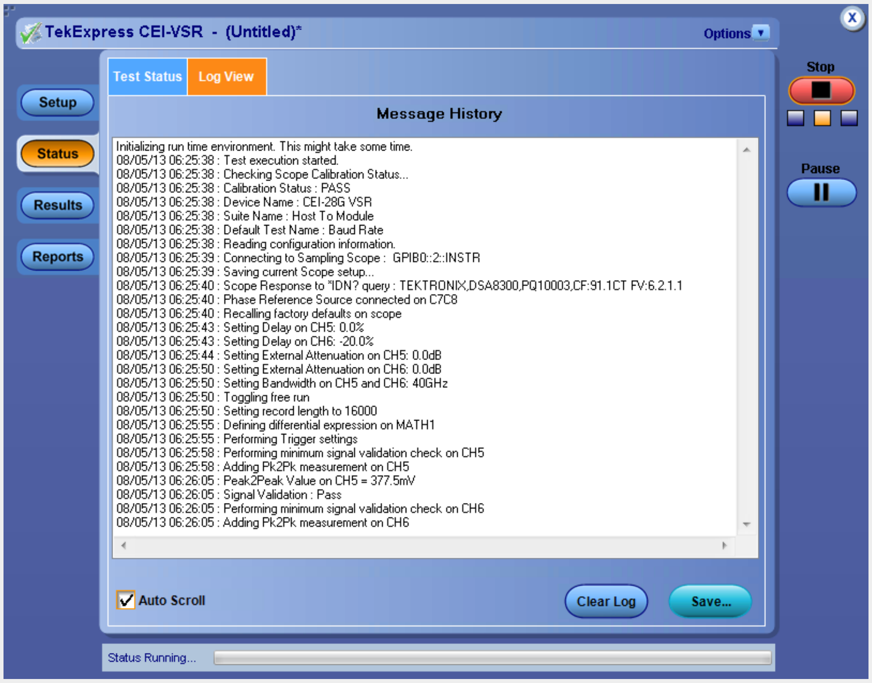
Status panel overview. The Status button accesses the Test Status and Log View tabs, which provide status on test acquisition and analysis (Test Status tab) and a listing of test tasks performed (Log View tab). The application opens the Test Status tab when you start a test run. You can select the Test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

Test status view



Test Name	Acquisition	Acquire Status	Analysis Status
CEI VSR			
Baud Rate	Baud Rate PRBS9	Completed	Completed
Differential Voltage Pk-Pk	PRBS9	Completed	Completed
Common Mode Noise RMS	Common Mode PRBS9	Completed	Completed
Transition time - 20/80%	PRBS9	Completed	Completed
Common Mode Voltage	N.A	Completed	Completed
Eye width at 10e-15 Probability (EW15)	PRBS9	Completed	Completed
Eye Height at 10e-15 Probability (EH15)	PRBS9	Completed	Completed

Log view



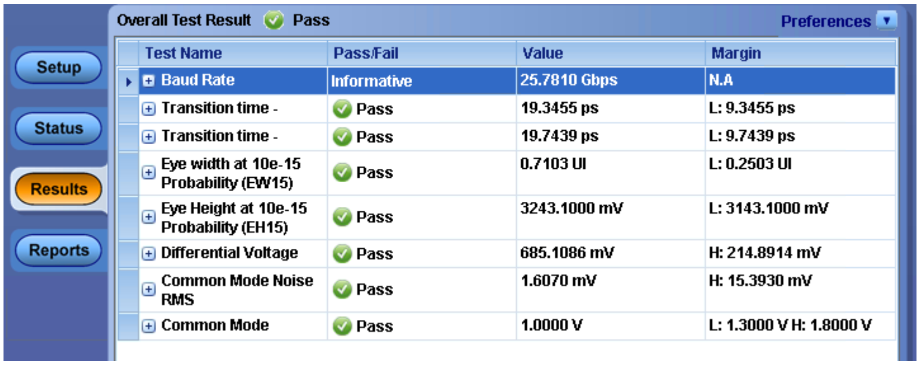
```
Initializing run time environment. This might take some time.
08/05/13 06:25:38 : Test execution started.
08/05/13 06:25:38 : Checking Scope Calibration Status...
08/05/13 06:25:38 : Calibration Status : PASS
08/05/13 06:25:38 : Device Name : CEI-28G VSR
08/05/13 06:25:38 : Suite Name : Host To Module
08/05/13 06:25:38 : Default Test Name : Baud Rate
08/05/13 06:25:38 : Reading configuration information.
08/05/13 06:25:39 : Connecting to Sampling Scope :  GPIB0::2::INSTR
08/05/13 06:25:39 : Saving current Scope setup...
08/05/13 06:25:40 : Scope Response to *IDN? query : TEKTRONIX,DSA8300,PQ10003,CF:91.1CT FV:6.2.1.1
08/05/13 06:25:40 : Phase Reference Source connected on C7C8
08/05/13 06:25:40 : Recalling factory defaults on scope
08/05/13 06:25:43 : Setting Delay on CH5: 0.0%
08/05/13 06:25:43 : Setting Delay on CH6: -20.0%
08/05/13 06:25:44 : Setting External Attenuation on CH5: 0.0dB
08/05/13 06:25:50 : Setting External Attenuation on CH6: 0.0dB
08/05/13 06:25:50 : Setting Bandwidth on CH5 and CH6: 40GHz
08/05/13 06:25:50 : Toggling free run
08/05/13 06:25:50 : Setting record length to 16000
08/05/13 06:25:55 : Defining differential expression on MATH1
08/05/13 06:25:55 : Performing Trigger settings
08/05/13 06:25:58 : Performing minimum signal validation check on CH5
08/05/13 06:25:58 : Adding Pk2Pk measurement on CH5
08/05/13 06:26:05 : Peak2Peak Value on CH5 = 377.5mV
08/05/13 06:26:05 : Signal Validation : Pass
08/05/13 06:26:05 : Performing minimum signal validation check on CH6
08/05/13 06:26:05 : Adding Pk2Pk measurement on CH6
```

Table 13: Status panel settings

Control	Description
Message History	Window that lists all executed test operations and timestamp information.
Show Detailed Log	Enables recording a more-detailed history of test execution. ***NOTE.*** *This must be selected before starting a measurement.*
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 overview. When a test finishes, the application automatically opens the **Results** panel to display a summary of signal and preset test results.



Overall Test Result ✓ Pass

Test Name	Pass/Fail	Value	Margin
Baud Rate	Informative	25.7810 Gbps	N.A
Transition time -	✓ Pass	19.3455 ps	L: 9.3455 ps
Transition time -	✓ Pass	19.7439 ps	L: 9.7439 ps
Eye width at 10e-15 Probability (EW15)	✓ Pass	0.7103 UI	L: 0.2503 UI
Eye Height at 10e-15 Probability (EH15)	✓ Pass	3243.1000 mV	L: 3143.1000 mV
Differential Voltage	✓ Pass	685.1086 mV	H: 214.8914 mV
Common Mode Noise RMS	✓ Pass	1.6070 mV	H: 15.3930 mV
Common Mode	✓ Pass	1.0000 V	L: 1.3000 V H: 1.8000 V

The Overall Test Result is displayed at the top left of the Results table. If all of the tests for the session pass, the overall test result is **Pass**. If one or more tests fail, the overall test result is **Fail**.

Set viewing preferences for this panel from the *Preferences menu* in the upper right corner. Viewing preferences include showing whether a test passed or failed, summary or detailed results, and enabling wordwrap.

Each test result occupies a row in the Results table. By default, results are displayed in summary format with the measurement details collapsed and with the Pass/Fail column visible. Change the view in the following ways:

- To expand and collapse tests to show more or less detail, click the plus and minus buttons in the table.
- To expand the width of a column, place the cursor over the vertical line that separates the column from the column to the right. When the cursor changes to a double-ended arrow, hold down the mouse button and drag the column to the desired width.
- To clear all test results displayed, click **Clear**.
- Use the *Preferences menu* to change how some items display in the Results panel.

See also. *View a report*

Application panels overview

Preferences menu. The Preferences menu is part of the Results panel display. Use the Preferences menu to change how some items display in the Results panel.

- To show or hide the Pass/Fail column, select **Preferences > Show Pass/Fail**.
- To collapse all expanded tests, select **Preferences > View Results Summary**.
- To expand all tests listed, select **Preferences > View Results Details**.
- To enable or disable the wordwrap feature, select **Preferences > Enable Wordwrap**.

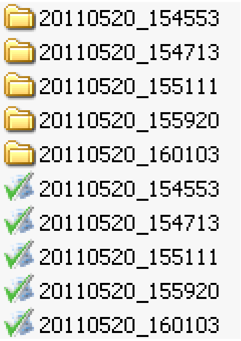
See also. *Results panel overview*

View test-related files. Files related to tests are stored in the `My TekExpress\CEI-VSR` folder. Each test setup in this folder has 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:



20110520_154553
20110520_154713
20110520_155111
20110520_155920
20110520_160103
20110520_154553
20110520_154713
20110520_155111
20110520_155920
20110520_160103

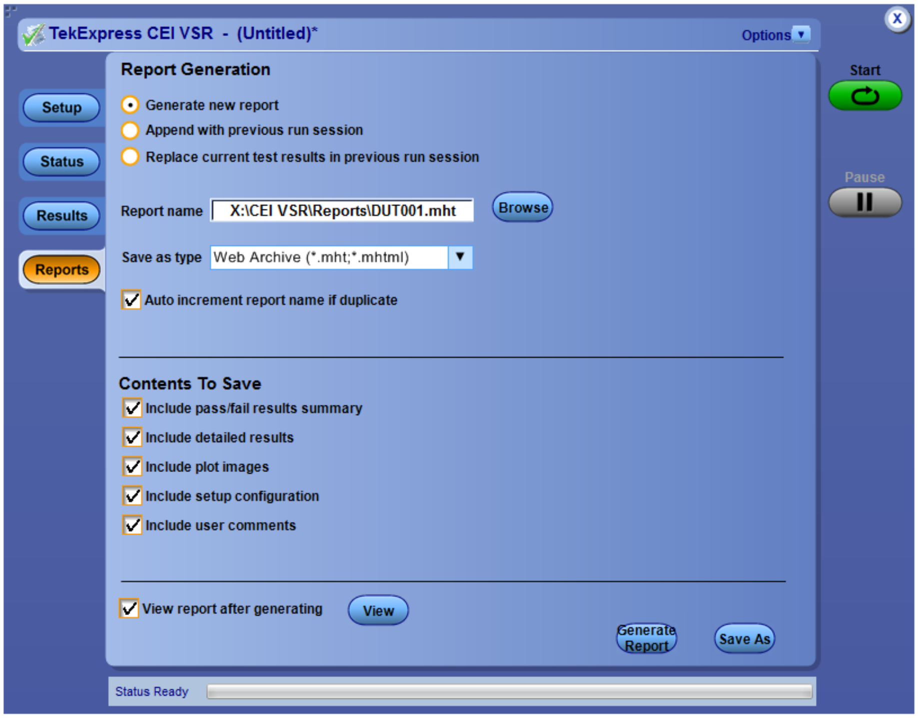
Each session folder contains image files of any plots generated from running the test session. If you selected to save all waveforms or ran tests using prerecorded waveform files, these are included here.

The first time you run a new, unsaved session, the session files are stored in the `Untitled Session` folder located at `..\My TekExpress\CEI-VSR`. 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 CEI-VSR application.

See also. *File name extensions*

Required My TekExpress folder settings

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.



TekExpress CEI VSR - (Untitled)*

Report Generation

- ● Generate new report
- ○ Append with previous run session
- ○ Replace current test results in previous run session

Report name: X:\CEI VSR\Reports\DUT001.mht

Save as type: Web Archive (*.mht;*.mhtml)

☑ Auto increment report name if duplicate

Contents To Save

- ☑ Include pass/fail results summary
- ☑ Include detailed results
- ☑ Include plot images
- ☑ Include setup configuration
- ☑ Include user comments

☑ View report after generating

Status Ready

For information on setting up reports, see *Select report options*. For information on viewing reports, see *View a Report*.

See also. *About panels*

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
Report Generation	
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.
Report name	Displays the name and location from which to open a report. The default location is at *\My TekExpress\CEI-VSR\Untitled Session*. The report file in this folder gets overwritten each time you run a test unless you specify a unique name or select to auto increment the report name. Change the report name or location. Do one of the following: ▪ In the Report Path field, type over the current folder path and name. ▪ Double-click in the Report Path field and then make selections from the popup keyboard and click the **Enter** button. 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\CEI-VSR\DUT001.mht. ***NOTE.*** *You cannot set the file location using the Browse button.* Open an existing report. Click **Browse**, locate and select the report file and then click **View** at the bottom of the panel.
Save as type	Saves a report in the specified file type. Lists supported file types to choose from. ***NOTE.*** *If you select a file type different from the default, be sure to change the report file name extension in the Report Name field to match.*
Auto increment report name if duplicate	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.
Contents To Save	
Include pass/fail results summary	Sets the application to include the color block labeled Test Result (indicating whether the test passed or failed) in the report. For details, see Report Contents in *View a report*.
Include detailed results	Sets the application to include parameter limits, execution time, and test-specific comments generated during the test.
Include plot images	Sets the application to include plotted diagrams such as Eye diagrams.

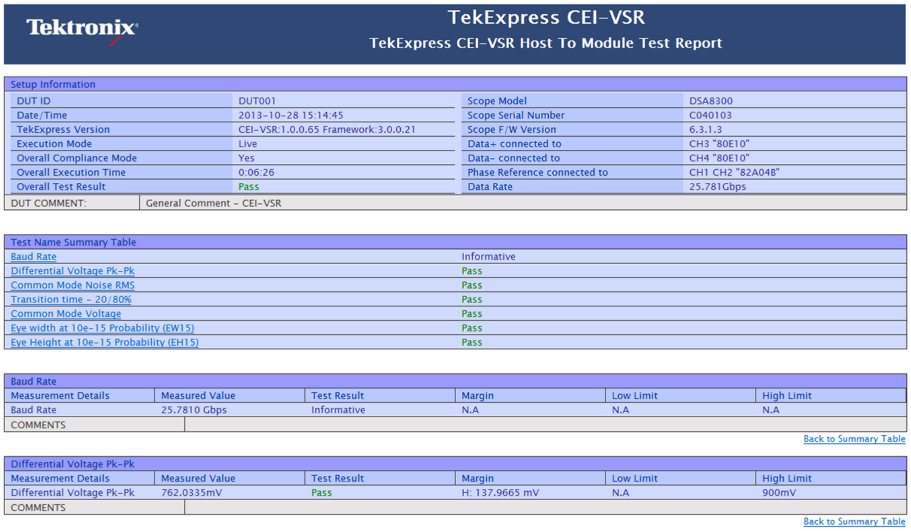
Setting	Description
Include setup configuration	Sets the application to include hardware and software information in the summary box at the top of the report. Information includes: the oscilloscope model and serial number, the oscilloscope firmware version, and software versions for applications used in the measurements.
Include user comments	Select to include any comments about the test that you or another user added in the DUT tab of the Setup panel. Comments appear in the Comments section, under the summary box at the beginning of each report.
View Report After Generating	Automatically opens the report in 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.



TekExpress CEI-VSR

TekExpress CEI-VSR Host To Module Test Report

Setup Information			
DUT ID	DUT001	Scope Model	DSA8300
Date/Time	2013-10-28 15:14:45	Scope Serial Number	C040103
TekExpress Version	CEI-VSR:1.0.0.65 Framework:3.0.0.21	Scope F/W Version	6.3.1.3
Execution Mode	Live	Data+ connected to	CH3 "80E10"
Overall Compliance Mode	Yes	Data- connected to	CH4 "80E10"
Overall Execution Time	0:06:26	Phase Reference connected to	CH1 CH2 "82A04B"
Overall Test Result	Pass	Data Rate	25.781Gbps
DUT COMMENT:	General Comment - CEI-VSR		

Test Name Summary Table	
Baud Rate	Informative
Differential Voltage Pk-Pk	Pass
Common Mode Noise RMS	Pass
Transition time - 20/80%	Pass
Common Mode Voltage	Pass
Eye width at 10e-15 Probability (EW15)	Pass
Eye Height at 10e-15 Probability (EH15)	Pass

Baud Rate					
Measurement Details	Measured Value	Test Result	Margin	Low Limit	High Limit
Baud Rate	25.7810 Gbps	Informative	N.A	N.A	N.A
COMMENTS					

Back to Summary Table

Differential Voltage Pk-Pk					
Measurement Details	Measured Value	Test Result	Margin	Low Limit	High Limit
Differential Voltage Pk-Pk	762.0335mV	Pass	H: 137.9665 mV	N.A	900mV
COMMENTS					

Back to Summary Table

Setup configuration information

Setup configuration information is listed in the summary box at the beginning of the report. This information includes the oscilloscope model and serial number, and software versions.

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.

Test result summary

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

See also. *Results panel overview*

View test-related files

Pre-measurement calibration procedures

Pre-measurement calibration guidelines

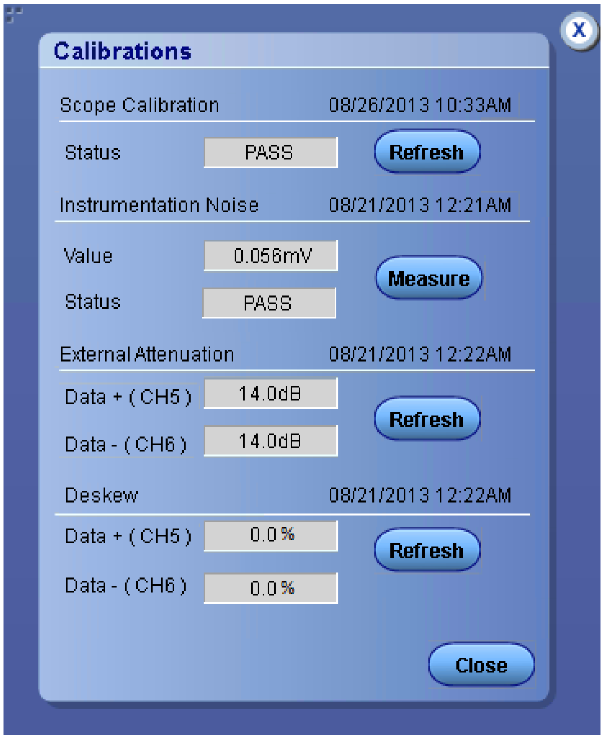
- You need to perform the following calibration procedures before starting a measurement session using the CEI-VSR software, and any time after that you make changes to the setup configuration, such as after installing or moving any sampling modules, cables, or connectors.
- The calibration procedures in this section require specific cables, connectors, and accessories to ensure measurement accuracy. See the *DSA8300 Digital Serial Analyzer Practices for Measurements on 25 Gb/s Signaling Application Note* (Tektronix part number 071-3207-XX) for information on where and how to obtain these parts.
- Perform the procedures in the following order:

  Instrument noise measurement

  Vertical gain calibration

  Deskew calibration (minimize common mode waveform method)

  Deskew calibration (minimize eye crossing method)



Calibrations

Scope Calibration 08/26/2013 10:33AM

Status: PASS

Instrumentation Noise 08/21/2013 12:21AM

Value: 0.056mV

Status: PASS

External Attenuation 08/21/2013 12:22AM

Data + (CH5): 14.0dB

Data - (CH6): 14.0dB

Deskew 08/21/2013 12:22AM

Data + (CH5): 0.0 %

Data - (CH6): 0.0 %

Instrument noise measurement

NOTE. *The following instrument noise measurement procedure assumes that the DUT Data+ and Data– lanes are connected to oscilloscope channels 5 and 6, respectively (80E09/B or 80E10/B Modules). Adjust the procedure accordingly if you connect the Data lanes to other channels for your measurements. This procedure is performed automatically when you click the Measure button under the Calibration Panel in the Acquisition menu.*

Instrumentation noise calibration

1. Disconnect all of the signals that are connected to the sampling oscilloscope.

2. Select **Setup > Vert > waveform C5 and C6** to **On**.

3. Set the Ch 5 and Ch 6 Bandwidth to **40 GHz**.

4. Set the minimum vertical scale per division to **1 mV/div** for Ch 5 and Ch 6.

5. Set the Trigger Source to **Free Run**.

6. Select measurement **Setup > Meas > Meas 1 > Pulse Amplitude: AC RMS**.

7. Set **Setup > Meas > Signal Type: Pulse**.

8. Set **Setups > Meas > Source: C5**.

9. Uncheck the **Use Wfm Database** control for the measurement.

10. Record the Ch 5 RMS value.

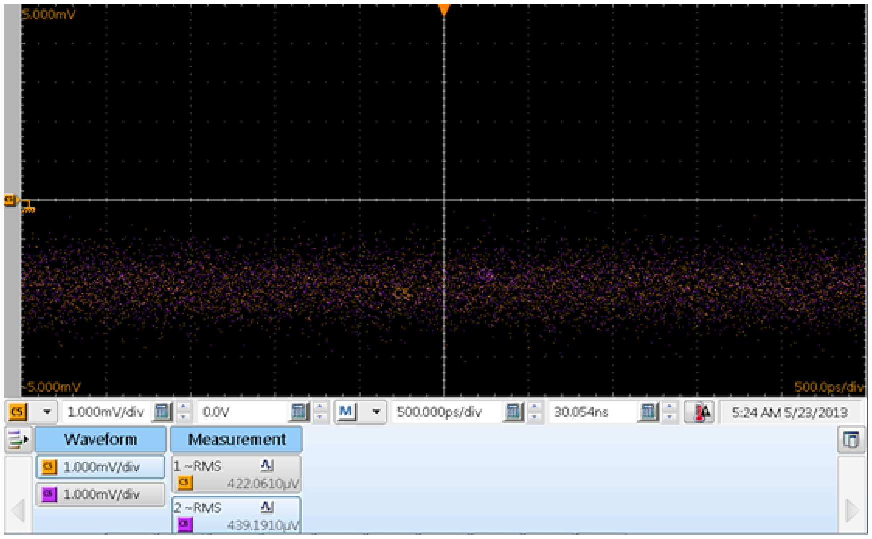
11. Select measurement **Setup > Meas > Meas 2 > Pulse Amplitude: AC RMS**.

12. Set **Setup > Meas > Signal Type: Pulse**.

13. Set **Setup > Meas > Source: C6**.

14. Uncheck the **Use Wfm Database** control for the measurement.

15. Record the Ch 6 RMS value as reported in the measurement readout.



C5 1.000mV/div 0.0V M 500.000ps/div 30.054ns 5:24 AM 5/23/2013

C5 1.000mV/div

C6 1.000mV/div

1 ~RMS C5 422.0610µV

2 ~RMS C6 439.1910µV

16. Use the following formula to calculate noise:

SQRT ((AC_RMS (C5)^2 + AC_RMS (C6) ^{2}))

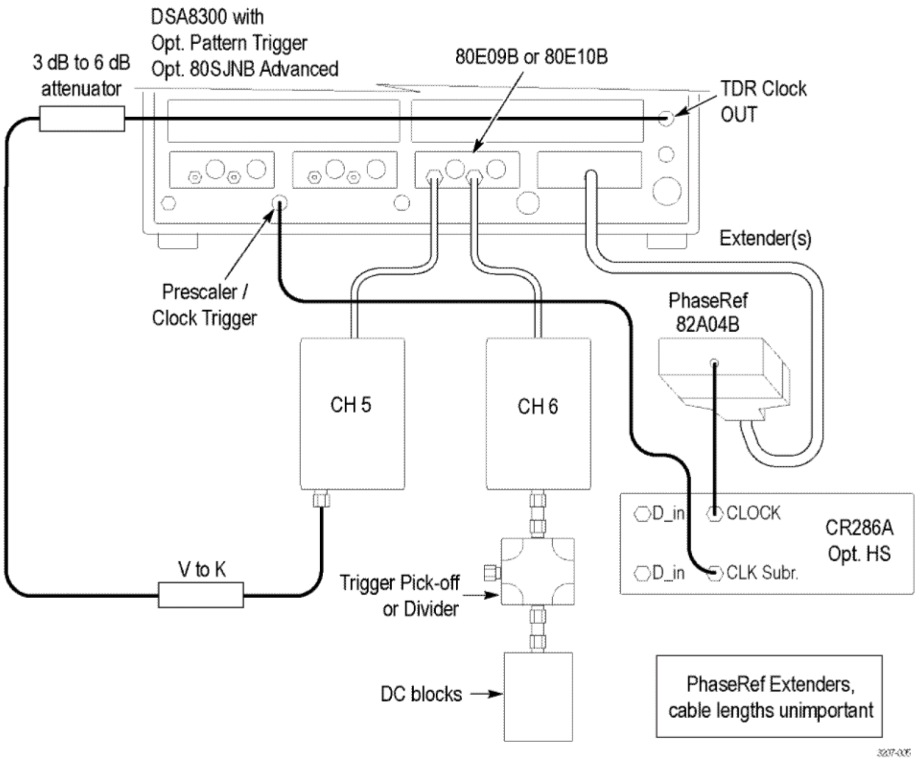
Noise level measurement should be in the range of **200 µV – 1 mV**.

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.

Vertical gain calibration

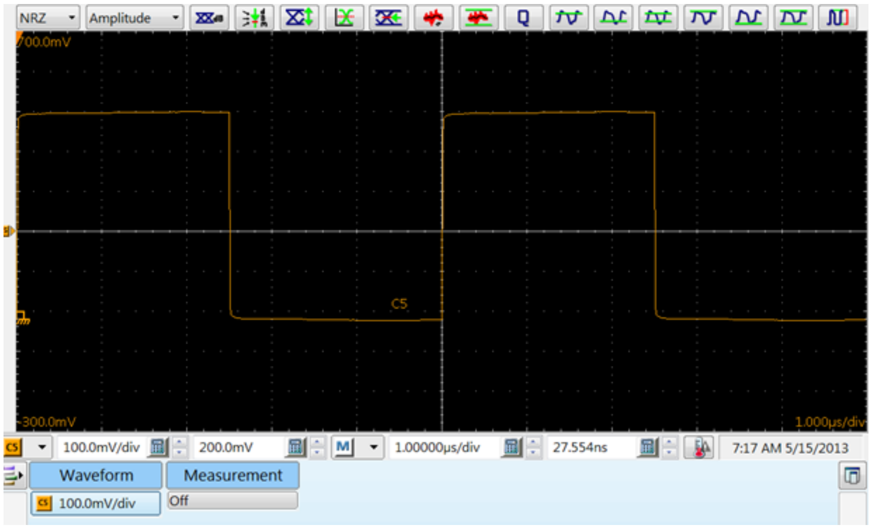
Use the following procedure to calculate the test configuration Vertical Gain:

1. Connect the instrument as shown in the following setup diagram:



2. Push **Default Setup**.
3. Set **Setup > Mode/Trigger > Trigger Source: TDR**.
4. Set **Setup > Vert > waveform C5** to **On**.
5. Set **Horizontal Scale time/div** to **1 us/div**.
6. Set **Setup > Horz > Record Length > 1000**(samples).
7. Set **Setup > Disp > Style: Show Vectors**.
8. Set oscilloscope Run/Stop state to **Run**.
9. Set **Setup > Acq > Acquisition Mode: Average (16 samples)**.

10. Set **Setup > Vertical > Channel: Offset** (on C5) to**200 mV** to the waveform within the dynamic range.



11. Add Amplitude measurement and configure the following settings:

    a. **Setup > Meas > Signal Type: Pulse**

    b. **Setup > Meas > Source: C5**

    c. **Setup > Meas > Pulse Amplitude: Amplitude**

    d. **Setup > Meas > Meas1**: select **On** (The oscilloscope creates this as **Meas1**)

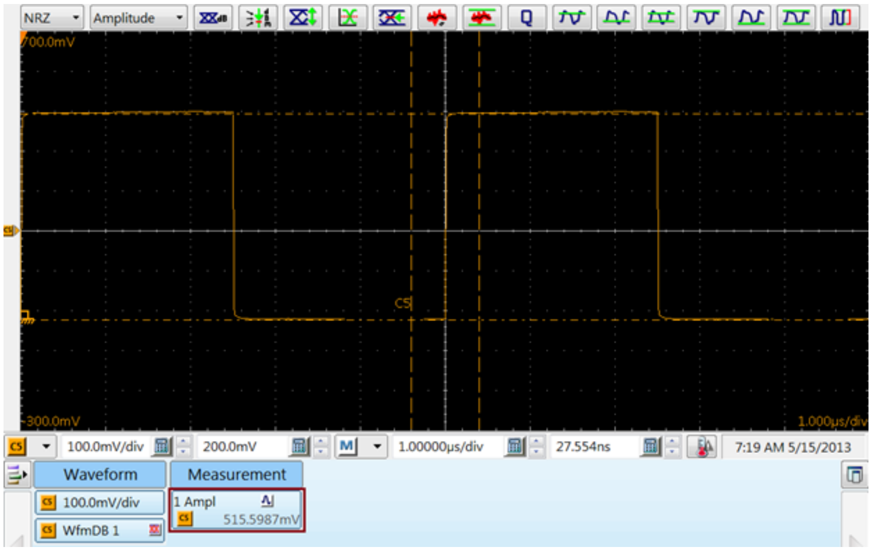
    e. **Setup > Meas > Region** to **On**

    f. **Setup > Meas > Region**: Gates G1 to **46%**

    g. **Setup > Meas > Region: Gates G1: 54%**

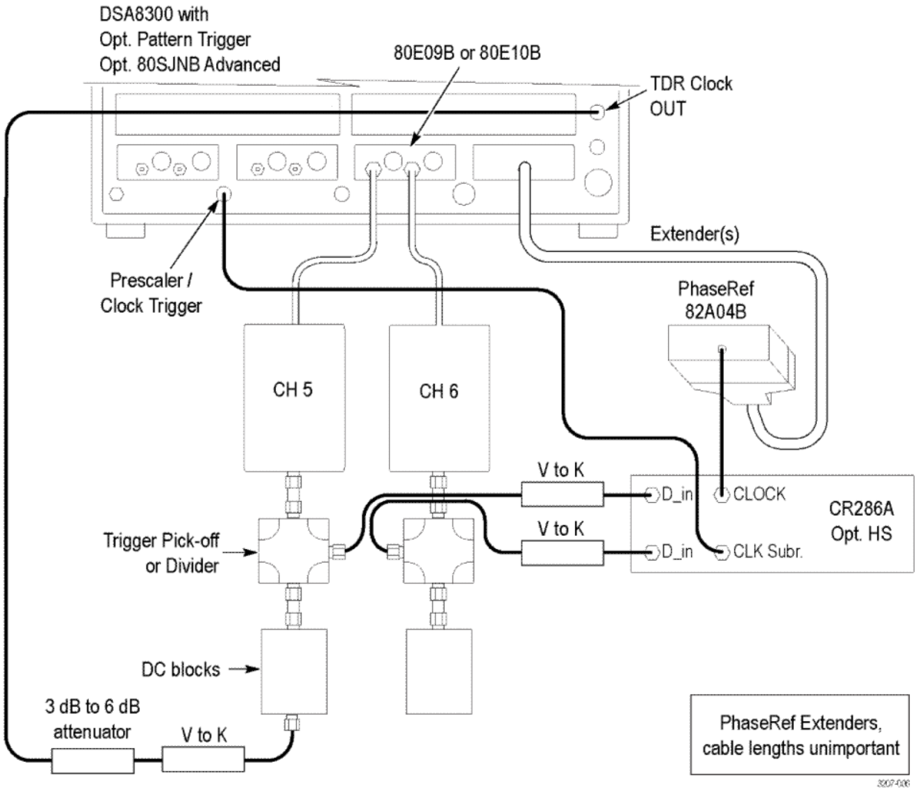
    h. **Setup > Meas> Annotations: On**

    i. Measure the Amplitude Referenced as shown in the following screen shot.



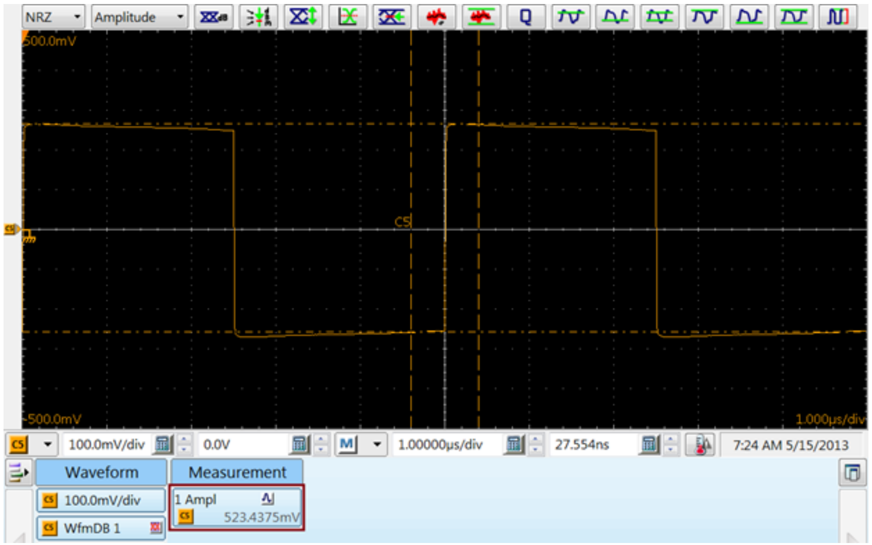
1 Ampl
C5 515.5987mV

12. Change the instrument connections as shown in the following figure (connect DC block and 6 dB attenuator to Ch 5 and other end to TDR Clock).



13. After making the connections shown in the above figure, measure the amplitude again.

14. Measure the Amplitude Apparent as shown in the following screen shot.



C5 100.0mV/div 0.0V M 1.00000µs/div 27.554ns 7:24 AM 5/15/2013

C5 100.0mV/div

C5 WfmDB 1

1 Ampl C5 523.4375mV

15. Calculate the Gain Correction factor for Channel 5:

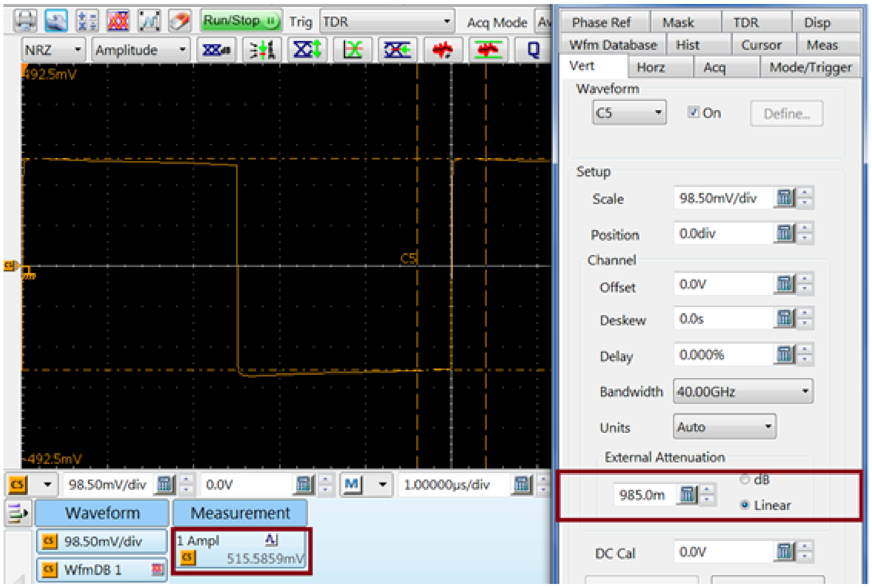
Channel 5 Gain correction factor = Amplitude_Referenced ÷ Amplitude_Apparent

16. Enter this correction factor into the instrument:

a. Setup > Vert: set waveform to C5

b. Set External Attenuation to Linear and

c. Enter the Gain correction factor for Channel 5 into the **External Attenuation** field as shown in the following image.



Waveform: C5 ☑ On

Scale: 98.50mV/div

Position: 0.0div

Offset: 0.0V

Deskew: 0.0s

Delay: 0.000%

Bandwidth: 40.00GHz

Units: Auto

External Attenuation: 985.0m ○ dB ● Linear

DC Cal: 0.0V

C5 98.50mV/div 0.0V M 1.00000µs/div

C5 98.50mV/div

1 Ampl C5 515.5859mV

17. Repeat steps 2 through 16, using Channel 6 instead of Channel 5, to calculate and enter the gain correction factor for Channel 6.

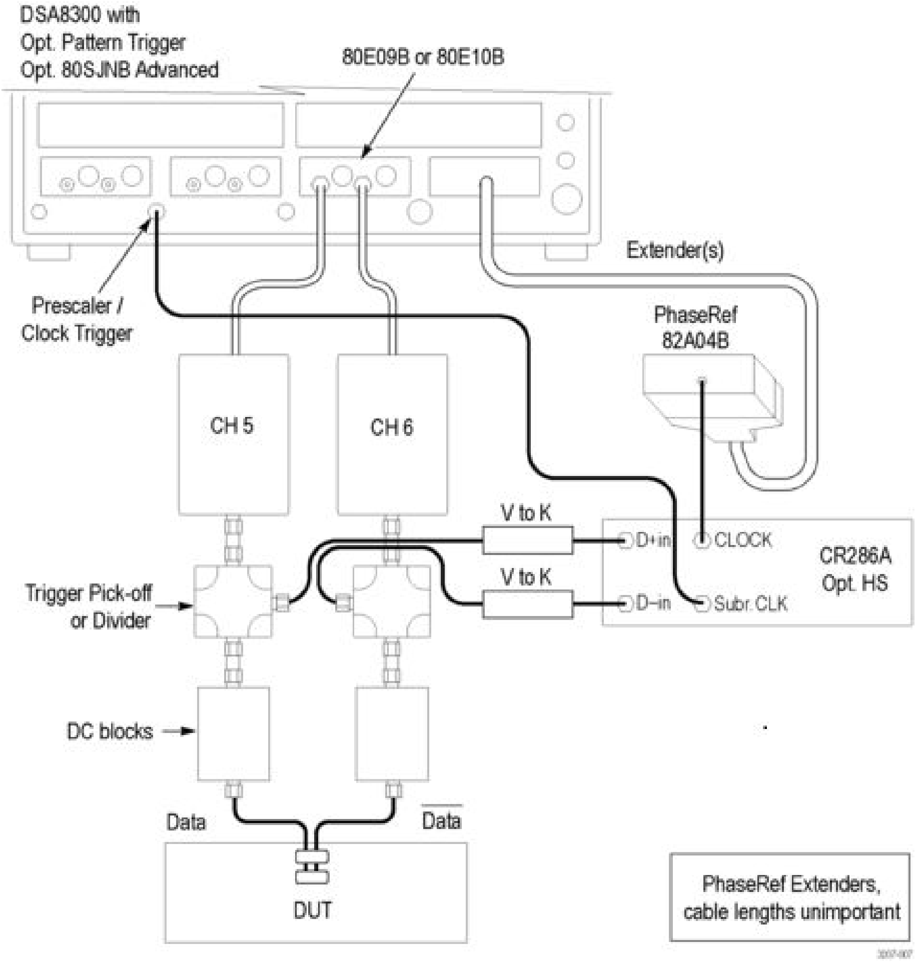
Deskew calibration (minimize common mode waveform method)

NOTE. *This procedure achieves deskew by minimizing the energy of a common mode waveform. This method is less sensitive to large skews, but can provide multiple minima.*

Another method is to minimize the eye-crossing to eye-crossing. The minimize eye crossing method fails for large initial skew, but if the initial skew is less than ½ UI it provides the best result.

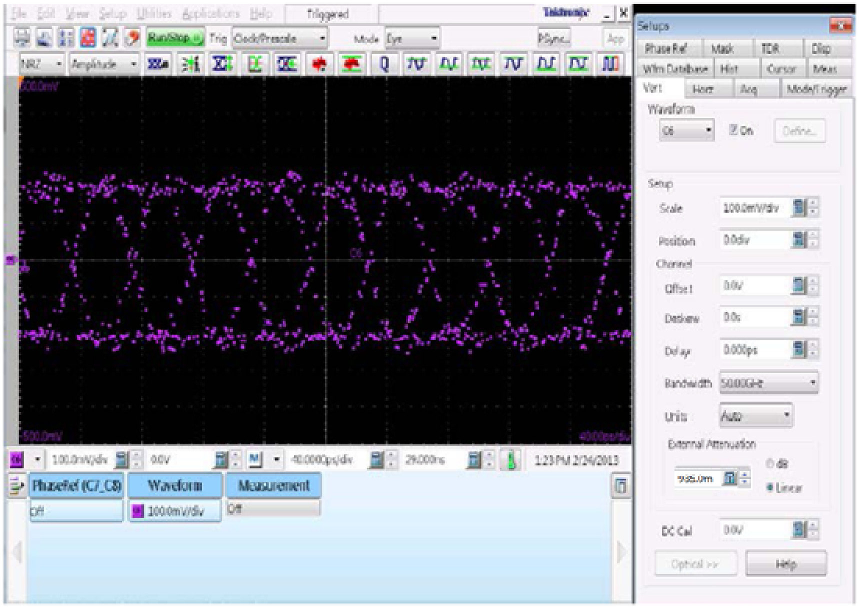
Thus the best result is obtained by following the two procedures in the order given here. The user can select just one or the other, depending on the need.

1. Connect the instrument as shown in the following setup diagram:



2. Configure the DUT settings:
   - **a.** Set the DUT output for standard operation
   - **b.** Set the DUT to generate a **PRBS9** pattern
3. Configure the oscilloscope channel settings:
   - **a.** **Setup > Mode/Trigger > Trigger Source** to **Clock/Prescale**
   - **b.** Select (enable) **C6** (Channel 6); turn OFF any other channel
   - **c.** **Setup > Acq > Acquisition Mode** to **Sample**
   - **d.** **Setup > Disp > Style**: uncheck **Show Vectors**
   - **e.** **Setup > Meas**: unselect (clear) On for all measurements
   - **f.** Set **Horizontal time/div** to approximately 1 UI/div (for example, 40 ps for 25 Gb/s)
   - **g.** Set **Setup > Horz > Record Length > 1000 [(Samples)]**
   - **h.** Select **Utilities > Autoset Properties**: clear (uncheck) Options: Horizontal, click Autoset
   - **i.** Close Autoset Properties
   - **j.** Set the oscilloscope Run/Stop state to **Run**

4. Observe that dimly visible eye diagrams are visible on the screen. If not, manually set the channel 6 V/div, Vertical Position, and Vertical Offset controls to position the waveform in the middle of the screen, as shown in the following figure:



Waveform: C6 ☑ On

Scale: 100.0mV/div

Position: 0.0div

Offset: 0.0V

Deskew: 0.0s

Delay: 0.000ps

Bandwidth: 50.00GHz

Units: Auto

External Attenuation: 935.0m ○ dB ● Linear

DC Cal: 0.0V

100.0mV/div 0.0V 40.0000ps/div 29.000ns 1:23 PM 2/24/2013

PhaseRef (C7_C8): Off

Waveform: 100.0mV/div

Measurement: Off

Eye diagrams Autoset, Vectors off. (sample dots enhanced in this picture)

5. Set**Setup > Vert > Waveform** to **C5**

6. **Set Setup > Vert > C6 Bandwidth** to **40 GHz**

7. **Set Setup > Vert > C5 Bandwidth** to **40 GHz**

8. Verify that both C5 and C6 have the External Attenuation values entered that were determined from the *Vertical gain calibration procedure*.

Pattern trigger settings:

Select **Setup > Mode/Trigger**: click **Pattern Sync/Framescan Setup**

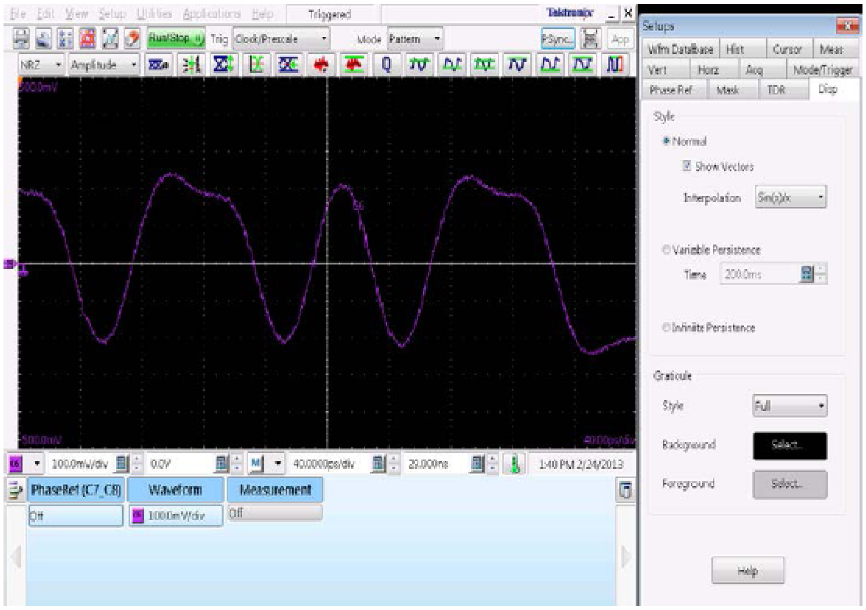
Clear the value in the **Data Rate** field and enter the correct Data Rate value (for example, 25.781Gb/s).

Set the **Pattern Length** field to **511** bits.

Click **AutoSync to selected waveform**.

Click **Close** to exit the **Pattern Sync/Framescan Setup**dialog box.

Select **Setup > Disp**: set Style to **Show Vectors**.



Select **Setups > Vert**: enable channel 6 waveform.

Set channel 6 **Vert Bandwidth** to **40 GHz**.

NOTE. *Observe both C5 and C6 displayed mid-screen, w/o clipping. Both signals should be of similar amplitude – if not, troubleshoot the interconnect to the DUT.*

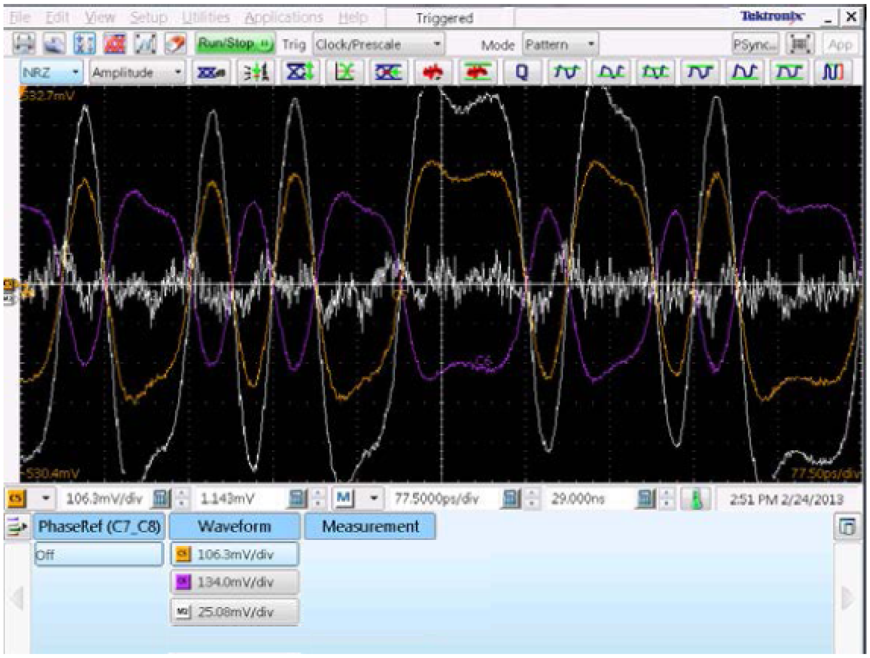
NOTE. *Position the screen such that multiple zero-crossings are seen. An alternative is to slow down time/div such that the longest run-length in the pattern would be no more than 1/3 of the screen. For example, if the pattern is PRBS9, the longest run-length is 9 bits; if the UI is 40 ps, then the duration of the longest RL is 40*9 = 360 ps. Set the time/div to 3*360/10 (for example, approx. 110 ps/div).*

Math measurement M2 settings:

Push the **Math** front-panel button

Define the math waveform **M2** to be **C5+C6**. Click **OK**.

Observe the common-mode waveform as the white trace as shown in the following figure.



Deskew of channel 6 to channel 5:

Select **Setup > Meas** and set the following parameters for the AC RMS measurement:

Set **Meas1** to **On**

Set**Signal Type** to **Pulse**

Set **Source** to **M2**

Set **Pulse Amplitude** to **AC RMS**

Set **Meas1** to **Select**

Setup > Vert > Waveform: Use the Front Panel **Fine** button and the Front Panel knob to set the C6 Adjust Channel: Delay to minimize the size of the M2 (white trace), or type values into the **Delay** window.

Math measurement M1 settings:

Push the **Math** front-panel button

Define the math waveform **M1** to be **C5-C6**. Click **OK**.

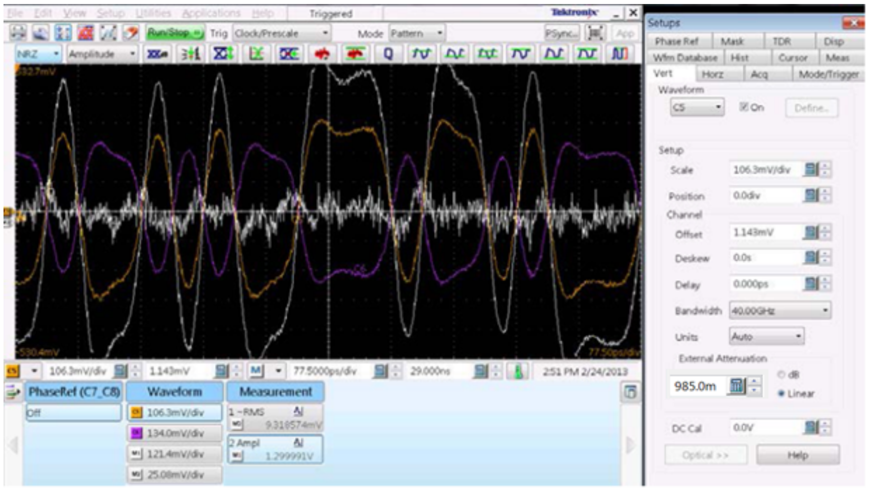
Observe the deskewed differential signal. Adjust M1 V/div if desired. If desired, enable diff. signal Amplitude measurement:

Setup > Meas: set **Signal Type** to **Pulse**

Setup > Meas: set **Source** to **M1**

Setup > Meas: set **Pulse Amplitude** to **Amplitude**

Setup > Meas: set **Meas2** to **On**



Waveform: C5 ☒ On

Setup

Scale 106.3mV/div

Position 0.0div

Channel

Offset 1.143mV

Deskew 0.0s

Delay 0.000ps

Bandwidth 40.00GHz

Units Auto

External Attenuation 985.0m ○ dB ● Linear

DC Cal 0.0V

106.3mV/div 1.143mV 77.5000ps/div 29.000ns 2:51 PM 2/24/2013

PhaseRef (C7_C8): Off

Waveform: 106.3mV/div, 134.0mV/div, 121.4mV/div, 25.08mV/div

Measurement: 1 ~RMS 9.318574mV; 2 Ampl 1.299991V

NOTE. *External Attenuation and Delay values are in the Vert tab fields.*

• End of Deskew calibration (minimize common mode waveform method) procedure •

• Go to *Deskew calibration (minimize eye crossing method)* procedure •

Deskew calibration (minimize eye crossing method)

NOTE. *This procedure achieves deskew by minimizing the waveform eye-crossing to eye-crossing. The eye crossing method fails for large initial skew, but if the initial skew is less than ½ UI it provides the best result.*

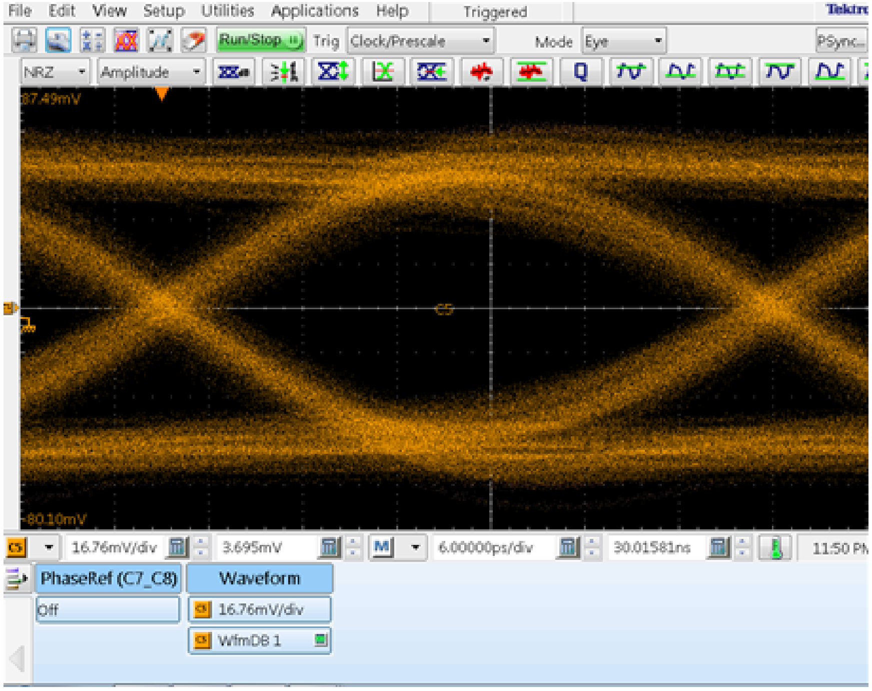
Another method is to minimize the energy of a common mode waveform. The common mode waveform method is less sensitive to large skews, but can provide multiple minima.

The best result is obtained by following the two procedures in the order given (minimize common mode waveform, minimize eye crossing). The user can select just one or the other, depending on the need.

This procedure uses the same connection setup as in the common mode procedure. To fine tune the deskew values by minimizing the interval between eye crossings:

1. Select C5 on front panel
2. Setup > Vert: set waveform-Ch5 to **On**.
3. Set the BW to **40 GHz**.
4. Setup > Horz: set the Bit Rate to the DUT's bit rate (for example, 25.781 Gb/s).
5. Setup > Horz: set the Record Length to any value above **1000** (1000 is the minimum recommended record length. Your measurement requirements may need more than 1000 records).
6. Setup > Horz: set the Horizontal Reference to **0%**.
7. Setup > Mode/Trigger: set the Scope Mode to **Eye**.
8. Setup > Wfm Database (Wfm DB1): select Source as C5; enable (check) **Display**; set Persistence to **Variable**; set Waveforms to **500**; set Display Options to **Intensity**.
9. Set the oscilloscope Run/Stop mode to **Run**.

10. Press **Autoset** front-panel button.



Trig Clock/Prescale

Mode Eye

C5 16.76mV/div 3.695mV M 6.00000ps/div 30.01581ns

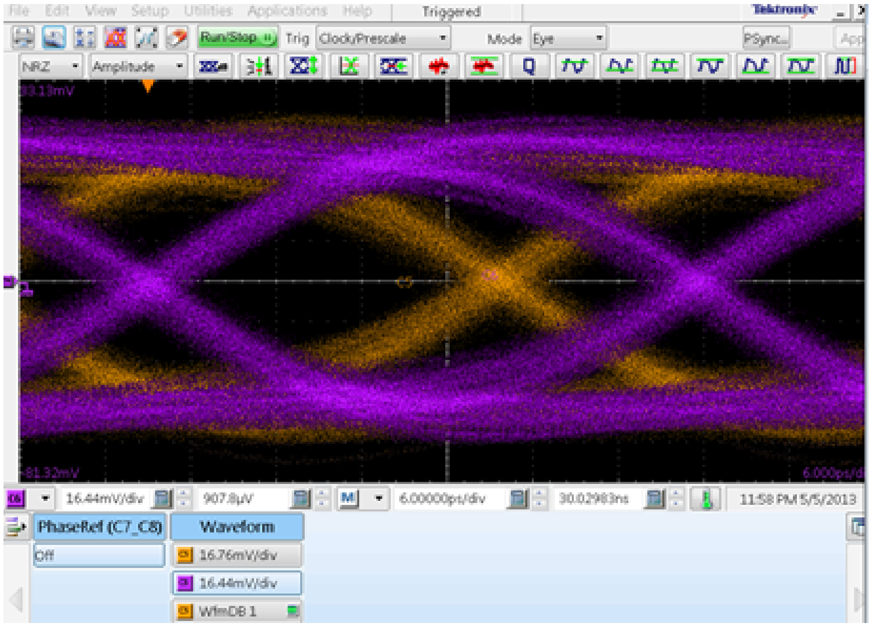
PhaseRef (C7_C8): Off

Waveform: C5 16.76mV/div; C5 WfmDB 1

11. Select (enable) **C6** (Channel 6) front panel button.

12. Setup > Wfm Database (Wfm DB2): select Source as **C6**; enable (check) **Display**; set Persistence to **Variable**; set Waveforms to **500**

13. Press **Autoset** front-panel button.



Trig Clock/Prescale

Mode Eye

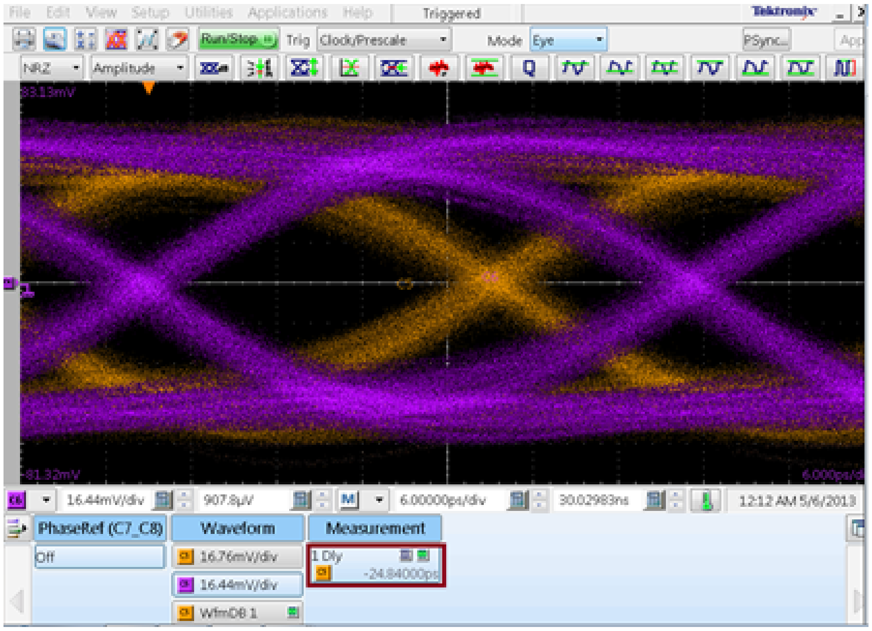
C6 16.44mV/div 907.8µV M 6.00000ps/div 30.02983ns

PhaseRef (C7_C8): Off

Waveform: C5 16.76mV/div; C6 16.44mV/div; C5 WfmDB 1

14. Set up a delay measurement between the C5 eye crossing and C6 eye crossing in the Setup > Meas tab:

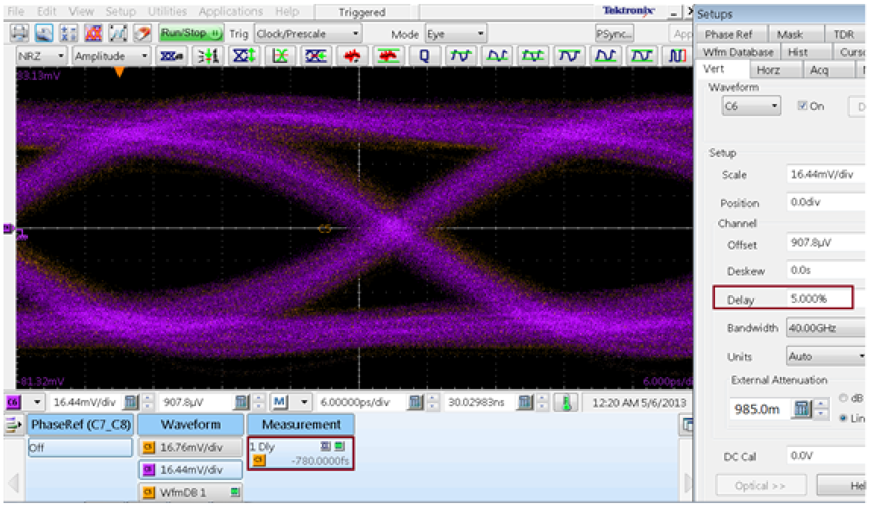
- **a.** **Setup > Meas > Select Meas > NRZ Timing > Delay**.
- **b.** Setup > Meas: click **Source1** and set Source to **C5** on **Main**.
- **c.** Setup > Meas: set Source Signal Type to **NRZ**.
- **d.** Setup > Meas: set Meas1 to **On**.
- **e.** Setup > Meas: click **Source2**: set source to **C6** on **Main**.
- **f.** Setup > Meas: click **Source1**.



C6 16.44mV/div 907.8µV M 6.00000ps/div 30.02983ns 12:12 AM 5/6/2013

PhaseRef (C7_C8)	Waveform	Measurement
Off	C5 16.76mV/div	1 Dly C5 -24.84000ps
	C6 16.44mV/div	
	C5 WfmDB 1	

15. Setup > Vert: Adjust the **Delay** value to minimize the delay between Ch5 and Ch6 eye crossings. Adjust the C6 channel delay until the delay measurement value becomes less than ¼ UI, as shown in the following image.



C6 16.44mV/div 907.8µV M 6.00000ps/div 30.02983ns 12:20 AM 5/6/2013

PhaseRef (C7_C8)	Waveform	Measurement
Off	C5 16.76mV/div	1 Dly C5 -780.0000fs
	C6 16.44mV/div	
	C5 WfmDB 1	

Waveform: C6 ☑ On

Setup

Scale	16.44mV/div
Position	0.0div
Channel	
Offset	907.8µV
Deskew	0.0s
Delay	5.000%
Bandwidth	40.00GHz
Units	Auto

External Attenuation: 985.0m ○ dB ● Lin

DC Cal 0.0V

• End of Deskew calibration (minimize eye crossing method) procedure •

Transition time algorithm procedure

1. Input the differential waveform exported from the 80SJNB application.
2. Find the position of the rising and falling edges in the input waveform, depending on the signal type. If you are using a PRBS9 test pattern, search for a transition sequence of five zeros and four ones.
3. The vHigh and vLow is estimated by the average signal within windows, from -3 UI to -2 UI, and from 2 UI to 3 UI, relative to the edge for a PRBS9 test pattern. The average of central 1 UI is used for the 8180 pattern.
4. On each rising edge, traverse forward until the first point crossing the upper limit (80%) is found (=End), and then traverse backward until the first point crossing the lower limit (20%) is found (= Start).
5. On each falling edge, traverse backward until the first point crossing the upper limit (80%) is found (=Start), and then traverse forward until the first point crossing the lower limit (20%) is found (= End).
6. Transition time is computed as Transition time = (End – Start). This value is averaged across all edges.

Instrument and DUT connection setup

Click the **Setup > Test Selection > Schematic** button to open a PDF file that shows the compliance test setup diagram(s) (instrument, DUT, connections, and cabling) for supported compliance testing configurations.

See also

Minimum system requirements

View connected instruments

Running tests

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

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

The application displays a report when the tests are complete.

See also

Configuration tab parameters

Prerun checklist

Do the following before you click Start to run a test:

***NOTE.** If this is the first time you are running a test on the application, make sure that you have done the steps in Required My TekExpress folder settings and the Calibration procedures, before continuing.*

1. Make sure that all the required instruments are properly warmed up (approximately 20 minutes).
2. Perform compensation:
   a. On the oscilloscope main menu, select **Utilities > Instrument Compensation**.
   b. Click the **Help** button in the Compensation window for information on how to perform instrument compensation.

See also

Instrument and DUT connection setup

Saving and recalling test setup files

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) are all saved under the setup name at **X:\CEI-VSR**.

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

Recall a saved test setup

Save a test setup file

Save a test setup before or after running a test to save the test settings. Create a new test setup from any open setup or from the default setup. When you select the default test setup, all parameters are returned to the application's default values.

To immediately save the current setup session to the same setup name, select **Options > Save Test Setup**.

To immediately save the current setup session to a new setup name, select **Options > Save Test Setup As**.

To create and save a new setup from the default test setup:

1. Select **Options > Default Test Setup** to return the application to default test settings.
2. Click the application **Setup** button and use the setup tabs to set required options and parameters (DUT, Test Selection, and so on).
3. Click the application **Reports** button and set your *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, edit the parameters and repeat this step until the test runs to your satisfaction.

   Running the test helps verify that all parameters are set correctly, but it is not a necessary step.
5. Select **Options > Save Test Setup**. Enter the file name for the new setup file. The application saves the file to X:\CEI-VSR \<*session_name*.

See also

View test-related files

Configuration tab parameters

Open (load) a saved test setup file

These instructions are for recalling saved test setups.

1. Select **Options > Open Test Setup**.
2. Select the setup from the list and click **Open**. Setup files must be located at **X:\CEI-VSR**.

See also

About test setups

Create a new test setup based on an existing one

Create a new test setup file based on an existing one

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

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

See also

About test setups

Set DUT parameters

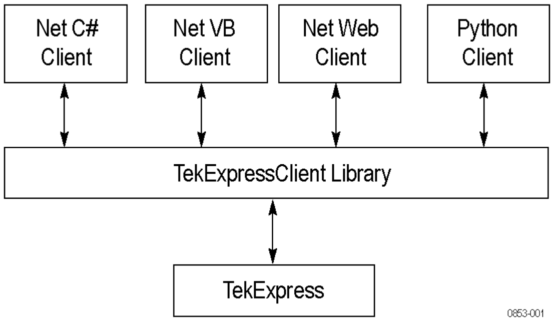
Configuration parameters

Select acquisitions

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

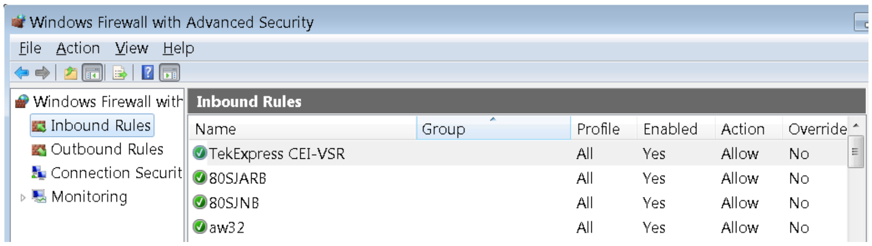
To enable remote access

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 CEI-VSR
   - TekExpress

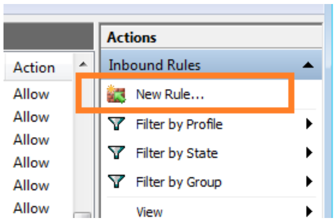


Name	Group	Profile	Enabled	Action	Override
TekExpress CEI-VSR		All	Yes	Allow	No
80SJARB		All	Yes	Allow	No
80SJNB		All	Yes	Allow	No
aw32		All	Yes	Allow	No

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

Run the New Inbound Rule Wizard

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

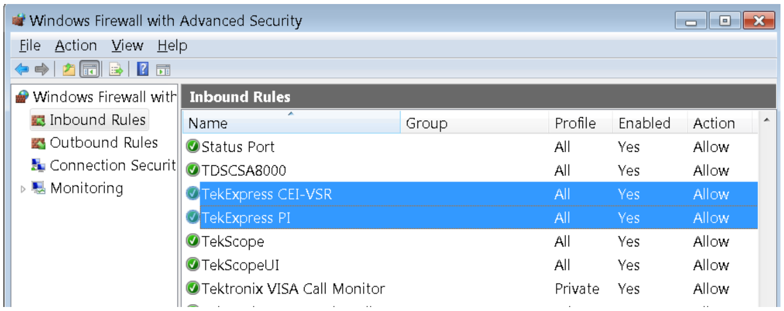


2. Verify that **Program** is selected in the Rule Type panel and click **Next**.
3. Click **Browse** in the Program panel and navigate to and select one of the following TekExpress applications (depending on the one for which you need to create a rule):
4. TekExpress CEI-VSR.exe
5. TekExpress.exe

NOTE. *See Application directories and content for the path to the application files.*

6. Click **Next**.
7. Verify that **Allow the connection** is selected in the Action panel and click **Next**.
8. Verify that all fields are selected (**Domain**, **Private**, and **Public**) in the Profile panel and click **Next**.
9. Use the fields in the Name panel to enter a name and optional description for the rule. For example, a name for the TekExpress CEI-VSR application could be **TekExpress CEI-VSR Application**. Add description text to further identify the rule.
10. Click **Finish** to return to the main Windows Firewall screen.

11. Scroll through the Inbound Rules list and verify that the list shows the rule that you just entered.



Name	Group	Profile	Enabled	Action
Status Port		All	Yes	Allow
TDSCSA8000		All	Yes	Allow
TekExpress CEI-VSR		All	Yes	Allow
TekExpress PI		All	Yes	Allow
TekScope		All	Yes	Allow
TekScopeUI		All	Yes	Allow
Tektronix VISA Call Monitor		Private	Yes	Allow

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

To use the remote access:

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

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

Requirements for developing TekExpress client

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

References required

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

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

Required steps for a client

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

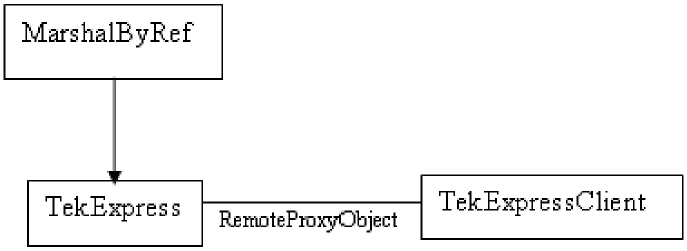
Develop a client UI to access the interfaces exposed through the server. This client loads `TekExpressClient.dll` to access the interfaces. After `TekExpressClient.dll` is loaded, the client UI can call the specific functions to run the operations requested by the client. When the client is up and running,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

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

   If any other client tries to access a server that is locked, it will receive a notification that the server is locked by another client.
3. When the client has connected to and locked the server, the client can access any of the programmatic controls needed to run the remote automations.
4. After the client operations finish, the client unlocks the server.

Remote proxy object

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



The following is an example:

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

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

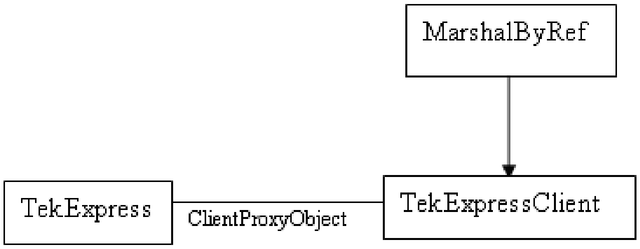
For example,

//Get a reference to the remote object

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

Client proxy object

Client exposes a proxy object to receive certain information.



For example,

//Register the client proxy object

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

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

```
RemotingConfiguration.RegisterWellKnownServiceType(typeof(ClientInterface), “Remote
Client Interface”, WellKnownObjectMode.Singleton);
```

//Expose the client proxy object through marshalling

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

The client proxy object is used for the following:

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

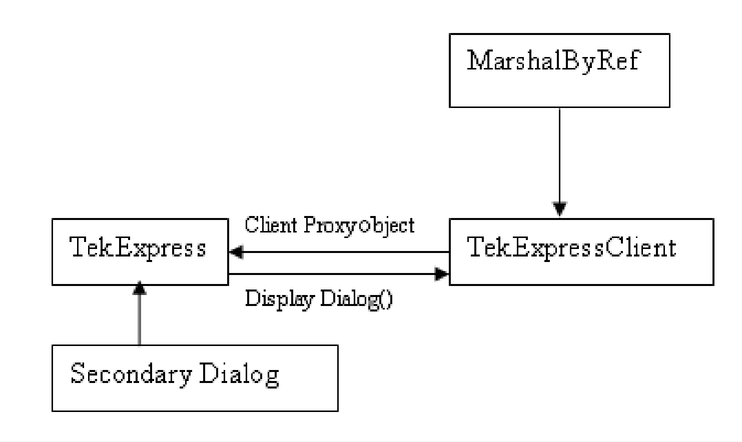
Examples

```
clientObject.clientIntf.DisplayDialog(caption, msg,iconType, btnType);

clientObject.clientIntf.TransferBytes(buffer, read, fileLength);
```

For more information, click the following links:

Secondary Dialog Message Handling



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

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

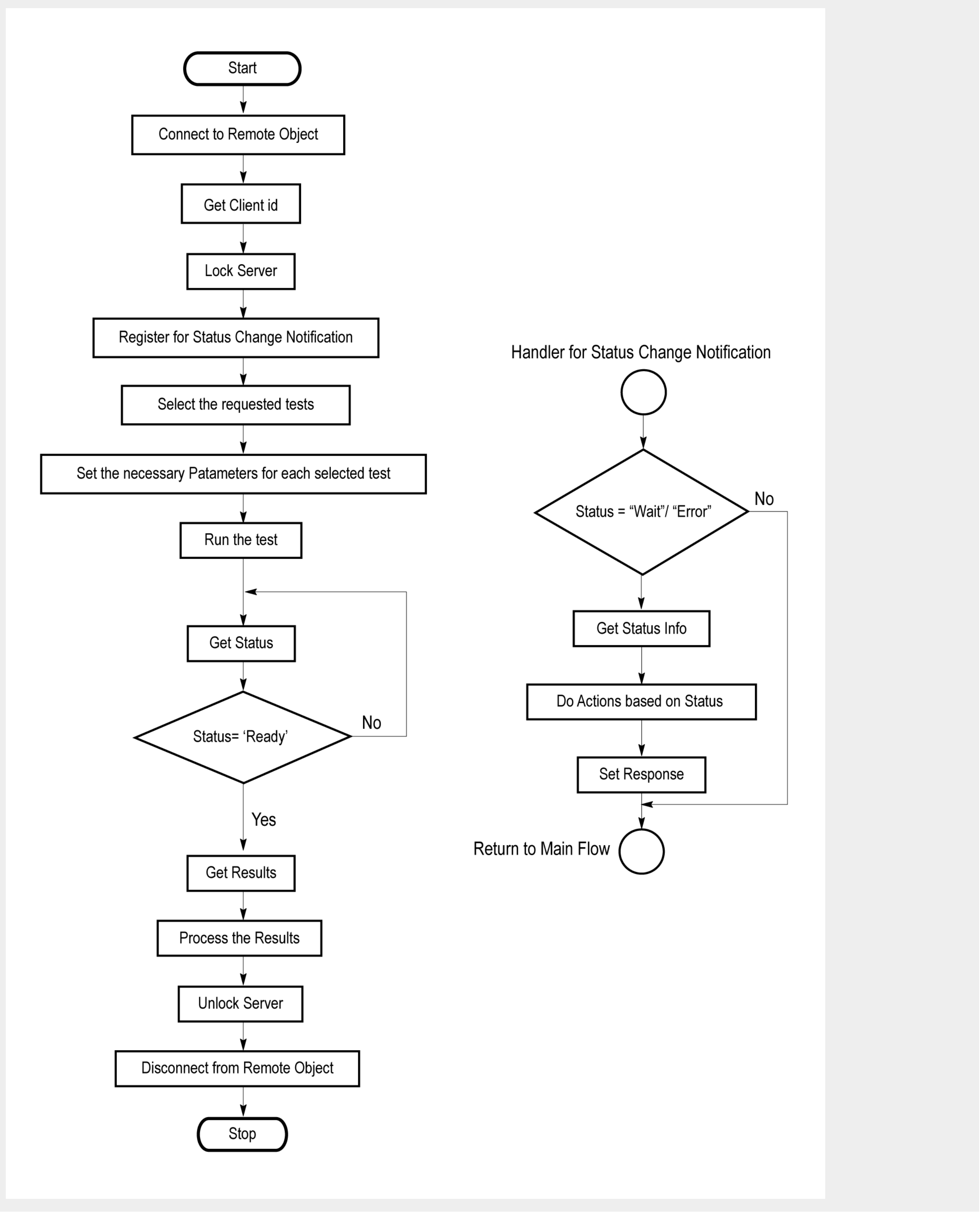
File Transfer Events

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

Client programmatic interface example

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

Process flowchart



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 CEI-VSR

Table 15: Remote access code example

Task	Code
Start the application	

Connect through an IP address.	'assigns client IP address to variable clientID; address valid until connection or measurement session ends (Disconnect). See Connect()
	`clientID = “ ”`
	`m_Client.Connect("localhost",out clientID)'True or False`

Lock the server	`m_Client.LockServer(clientID)`
Disable the Popups	`m_Client.SetVerboseMode(clientID, false)`
Set the DUT ID	`m_Client.SetDutId(clientID, "DUT_Name")`
Run with set configurations	`m_Client.Run(clientID)`

Wait for the test to complete.	`Do`
	`Thread.Sleep(500)`
	`m_Client.Application_Status(clientID)`
	`Select Case status`
	`Case "Wait"`

Get the current state information	`mClient.GetCurrentStateInfo(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxButtontexts)`

Send the response	`mClient.SendResponse(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxResponse)`
	`End Select`
	`Loop Until status = "Ready"`

Save results	'Save all results values from folder for current run `m_Client.TransferResult(clientID, logDirname)`
Unlock the server	`m_Client.UnlockServer(clientID)`
Disconnect from server	`m_Client.Disconnect()`
Exit the application	

CEI VSR programmer interface commands

ApplicationStatus()

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

Parameters.

Name	Type	Direction	Description

clientId	string	IN	Identifier of the client that is performing the remote function.
			clientId variable
			clientId variable
			clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable:
			`clientID = “ ”`
			`m_Client.Connect("localhost",out clientId)'True or False`
			The clientId variable is stored until you call the Disconnect command.

NOTE. *The Fail condition for 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
			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.

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
			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. 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	Identifier of the client that is performing the remote function.
			clientId variable
			clientId variable
			clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable:
			`clientID = “ ”`
			`m_Client.Connect("localhost",out clientId)'True or False`
			The clientId variable is stored until you call the Disconnect command.

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	Identifier of the client that is performing the remote function.
			clientId variable
			clientId variable
			clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable:
			`clientID = “ ”`
			`m_Client.Connect("localhost",out clientId)'True or False`
			The clientId variable is stored until you call the Disconnect command.

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

1: Success

–1: Failure

Example. `try`

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

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

Related command(s). *Connect*

GetCurrentStateInfo()

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

Except client ID, all the others are Out parameters.

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

Parameters.

Parameter	Type	Direction	Description

clientId	string	IN	Identifier of the client that is performing t
			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.
			he remote function.
			clientId variable
			clientId variable
			clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable:
			`clientID = “ ”`
			`m_Client.Connect("localhost",out clientId)'True or False`
			The clientId variable is stored until you call the Disconnect command.

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	Identifier of the client that is performing the remote function.
			clientId variable
			clientId variable
			clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable:
			`clientID = “ ”`
			`m_Client.Connect("localhost",out clientId)'True or False`
			The clientId variable is stored until you call the Disconnect command.

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
			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.

device	string	IN	Specifies the DUT type (**CEI-28G VSR**)
suite	string	IN	string with device connection type. Valid values are **Host To Module** and **Module to Host**.
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, “CEI-28G VSR”, “Host to Module”, test);

GetPassFailStatus(clientId, “Host”, “Module to Host”, 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	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.

device	string	IN	Specifies the DUT type (**CEI-28G VSR**).
suite	string	IN	string with device connection type. Valid values are **Host To Module** and **Module to Host**.
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 **“Scope Model”** , **“TekExpress Version”**, or **“Application Version”** 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 CEI-VSR 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
			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.

device	string	IN	Specifies the DUT type (**CEI-28G VSR**).
suite	string	IN	string with device connection type. Valid values are **Host To Module** and **Module to Host**.
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 **“Value”** 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	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 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	Identifier of the client that is performing the remote function.
			clientId variable
			clientId variable
			clientId is a user-defined variable that stores the client ID address information. Use the Connect() command to fill this variable:
			`clientID = “ ”`
			`m_Client.Connect("localhost",out clientId)'True or False`
			The clientId variable is stored until you call the Disconnect command.

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	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.

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
WaitingMsbBxCaption	string	OUT	The wait state or error state message sent to you
WaitingMsbBxMessage	string	OUT	The wait state/error state message sent to you
WaitingMsbBxButtontexts	string array	OUT	An array of strings containing the possible response types that you can send

clientId variable

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

```
clientID = “ ”
m_Client.Connect("localhost",out clientId)'True or False
```

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

NOTE. *The Fail condition for 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 WaitingMsbBxCaption, out WaitingMsbBxMessage, out WaitingMsbBxButtontexts)

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 (**CEI-28G VSR**)

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,"CEI-28G VSR")

SelectSuite()

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

Parameters.

Parameter	Type	Direction	Description
clientId	string	IN	Identifier of the client that is connected to the server clientId variable
device	string	IN	Specifies the DUT type (**CEI-28G VSR**)
suite	string	IN	string with device connection type. Valid values are **Host to Module** and **Module to Host**.

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,"CEI-28G VSR","Host to Module",true);

SelectSuite(clientId,"CEI-28G VSR","Module to Host",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
			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.

device	string	IN	Specifies the DUT type (CEI-28G VSR).
suite	string	IN	string with device connection type. Valid values are **Host to Module** and **Module to Host**
test	string	IN	Name of the CEI-VSR test.

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

Example. SelectTest(clientId, device, suite, "Differential Voltage Pk-Pk", 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
			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.

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 (**CEI-28G VSR**).
suite	string	IN	Valid values are **Host to Module** and **Module to Host**.
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.

Set Pattern Type: *Set pattern type* on page 84

Set Record Length Value: *Set record length value* on page 84

paramString values for SetAcquireParameter command

Set pattern type. Use this paramString value to set the pattern type for Transition time – 20/80% test. This is the same as selecting the Pattern Type control on the Configuration > Measurements > Transition Time – 20/80% tab.

The value in bold font is the default value.

Values:. For test Transition Time &#x2013; 20/80%.

- **PRBS9$AcquisitionIncluded$True**
- **8180$AcquisitionIncluded$False**
- PRBS9$AcquisitionIncluded$False
- 8180$AcquisitionIncluded$True

Example. SetAcquireParameter(clientId, “CEI-28G VSR”, “Host To Module”, "Transition time – 20/80%", " PRBS9$AcquisitionIncluded$False”);

SetAcquireParameter(clientId, “CEI-28G VSR”, “Host To Module”, "Transition time – 20/80%", " 8180$AcquisitionIncluded $True”);

Set record length value. Use this paramString value to set the record length Value. This is the same as selecting the Record Length control on the Configuration > Measurements > Selected test in tree view tab.

The value in bold font is the default value.

Values:. For test Common Mode Noise.

- RMS Common Mode Noise PRBS9$Record Length$4000
- Common Mode Noise PRBS9$Record Length$8000
- **Common Mode Noise PRBS9$Record Length$16000**

For test Differential Voltage Pk-Pk.

- PRBS9$Record Length$4000
- PRBS9$Record Length$8000
- **PRBS9$Record Length$16000**

Example. SetAcquireParameter(clientId, “CEI-28G VSR”, “Host To Module”, "Common Mode Noise RMS", "Common Mode Noise PRBS9$Record Length$4000”);

SetAcquireParameter(clientId, “CEI-28G VSR”, “Host To Module”, "Differential Voltage Pk-Pk", "PRBS9$Record Length$4000”);

The SetGeneralParameter command

SetGeneralParameter(). SetGeneralParameter(clientId, device, suite, "", paramString)ParametersReturn valueparamString argument values

This command sets the general parameter and its value based on the "paramString" argument values as listed.

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 (**CEI-28G VSR**).
suite	string	IN	Valid values are **Host To Module** and **Module to Host**.
test	string	IN	Specifies the name of the test for which to obtain the pass or fail status or a test result value. Enter a null value for this field ("").
parameterString	string	IN	Specifies the control to set. See the following links for argument values and examples for this field.

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.

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 Channel for Data– Signal: *Select channel for data- signal* on page 86

Select Channel for Data+ Signal: *Select channel for data+ signal* on page 86

Select Data Rate: *Select data rate* on page 86

Set Bandwidth of scope: *Set bandwidth of oscilloscope* on page 87

Set BER Value: *Set BER value* on page 87

Set CRU Loop Bandwidth Custom Mode value: *Set CRU loop bandwidth custom mode value* on page 87

Set CRU Loop Bandwidth Mode: *Set CRU loop bandwidth mode* on page 87

Set CRU Peaking Value: *Set CRU peaking value* on page 87

Set CTLE Filter file: *Set sample count value* on page 89

Select channel for data- signal. Use this paramString value to select the channel where Data– signal is connected used by the application. This is the same as using the **Lane Channel Selection Table** controls on the **Acquisition** tab.

The value in bold font is the default value.

Values:. Lane0 Connected to:Data –:Single Ended$<Channel><Channel> corresponds to one of the following values CH1,CH2,CH3,CH4,CH5,CH6,CH7,CH8

Example. SetGeneralParameter(clientId, , “CEI-28G VSR”, “Host To Module”, "", " Lane0 Connected to:Data –:Single Ended $CH1")

Select channel for data+ signal. Use this paramString value to select the channel where Data+ signal is connected used by the application. This is the same as using the **Lane Channel Selection Table** controls on the **Acquisition** tab.

The value in bold font is the default value.

Values:. Lane0 Connected to:Data +:Single Ended$<Channel><Channel> corresponds to one of the following values CH1,CH2,CH3,CH4,CH5,CH6,CH7,CH8

Example. SetGeneralParameter(clientId, , “CEI-28G VSR”, “Host To Module”, "", " Lane0 Connected to:Data +:Single Ended $CH1")

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, “CEI-28G VSR”, “Host To Module”, "", " Data Rate$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.

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*

Set bandwidth of oscilloscope. Use this paramString value to set the oscilloscope bandwidth. This is the same as selecting the Other Settings>**Bandwidth** control on the **Configuration** > **Global Settings** tab.

The value in bold font is the default value.

Values:.

- **BandWidth$40**
- BandWidth$50

Example. SetGeneralParameter(clientId, “CEI-28G VSR”, “Host To Module”, "", "BandWidth$50”);

Set BER value. Use this paramString value to set the BER Value. This is the same as selecting the BER control on the Configuration > Global Settings tab.

The value in bold font is the default value.

Values:. BER$1E-15

Example. SetGeneralParameter(clientId, “CEI-28G VSR”, “Host To Module”, "", "BER$1E-15”);

Set CRU loop bandwidth custom mode value. Use this paramString value to set the CRU Loop Bandwidth Mode. This is the same as selecting the **CRU Loop Bandwidth Custom Mode value** control on the **Configuration** > **Global Settings** tab.

The value in bold font is the default value.

Values:.

- **CRU Loop BandWidth Custom$10**

  The value should be between 10 and 12.
- CRU Loop BandWidth Mode$Custom

Example. SetGeneralParameter(clientId, “CEI-28G VSR”, “Host To Module”, "", "CRU Loop BandWidth Custom$10.5”);

Set CRU loop bandwidth mode. Use this paramString value to set the CRU Loop Bandwidth Mode. This is the same as selecting the **CRU Loop Bandwidth** control on the **Configuration** > **Global Settings** tab.

The value in bold font is the default value.

Values:.

- **CRU Loop BandWidth Mode$DataRate / 2578**
- CRU Loop BandWidth Mode$Custom

Example. SetGeneralParameter(clientId, “CEI-28G VSR”, “Host To Module”, "", "CRU Loop BandWidth Mode$ Custom”);

Set CRU peaking value. Use this paramString value to set the CRU Peaking Value. This is the same as selecting the Peaking Value control on the Configuration > Global Settings tab.

The value in bold font is the default value.

Values:. CRU Peaking Value$0.1 The value should be between 0.0 to 1.0

Example. SetGeneralParameter(clientId, “CEI-28G VSR”, “Host To Module”, "", "CRU Peaking Value$0.2”);

Set CTLE filter file. Use this paramString value to set the CTLE filter file. This is the same as selecting the **CTLE Filter File** control on the**Configuration > Global Settings** tab.

The value in bold font is the default value.

Values:. For Host To Module

- **CTLE FilterFile$All**
- CTLE FilterFile $ sdlaCtle_JNB_1dB.flt
- CTLE FilterFile $ sdlaCtle_JNB_2dB.flt
- CTLE FilterFile $ sdlaCtle_JNB_3dB.flt
- CTLE FilterFile $ sdlaCtle_JNB_4dB.flt
- CTLE FilterFile $ sdlaCtle_JNB_5dB.flt
- CTLE FilterFile $ sdlaCtle_JNB_6dB.flt
- CTLE FilterFile $ sdlaCtle_JNB_7dB.flt
- CTLE FilterFile$Custom

NOTE. *To run this command following pre-requisite command should be run prior*

- *SetComplianceMode(clientID, User-Defined)*

For Module To Host

- CTLE FilterFile $ sdlaCtle_JNB_8dB.flt
- CTLE FilterFile$All CTLE FilterFile $ sdlaCtle_JNB_1dB.flt
- CTLE FilterFile $ sdlaCtle_JNB_2dB.flt
- CTLE FilterFile$Custom

NOTE. *To run this command following pre-requisite command should be run prior*

- *SetComplianceMode(clientID, User-Defined)*

Example. SetGeneralParameter(clientId, , “CEI-28G VSR”, “Host To Module”, "", "CTLE FilterFile $ sdlaCtle_JNB_1dB.flt ");

SetGeneralParameter(clientId, , “CEI-28G VSR”, “Module To Host”, "", "CTLE FilterFile $ sdlaCtle_JNB_1dB.flt ");

Set custom filter file. Use this paramString value to set the custom CTLE filter file. This is the same as selecting the custom **CTLE Filter File** control on the **Configuration > Global Settings** tab.

Values:. For Host To Module

- **Custom Filter File$<filterfilePath>**

For Module To Host

- **Custom Filter File$<filterfilePath>**

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

- *SetComplianceMode(clientID, User-Defined)*
- *SetGeneralParameter(clientID,,"CEI-28G VSR", "Host To Module", "", "CTLE FilterFile$Custom")*

Example. Custom Filter File$C:\Program Files\Tektronix\TekExpress\TekExpress CEI-VSR\Lib\CTLE Filters \sdlaCtle_JNB_1dB.flt

Set sample count value. Use this paramString value to set the sample count value. This is the same as setting the **Sample Count Value** on the **Configuration > Global Settings** tab.

The value in bold font is the default value.

Values:. SAMPLE COUNT$4

Example. SetGeneralParameter(clientId, , “CEI-28G VSR”, “Host To Module”, "", "Sample Count$4");

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
			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.

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	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.

filePath	string	IN	The location where the screen shots must be saved in the client. ***NOTE.*** *If the client does not provide the location to save the report, the report is saved at C:\ProgramFiles.*

NOTE. *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
			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.

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*

Index