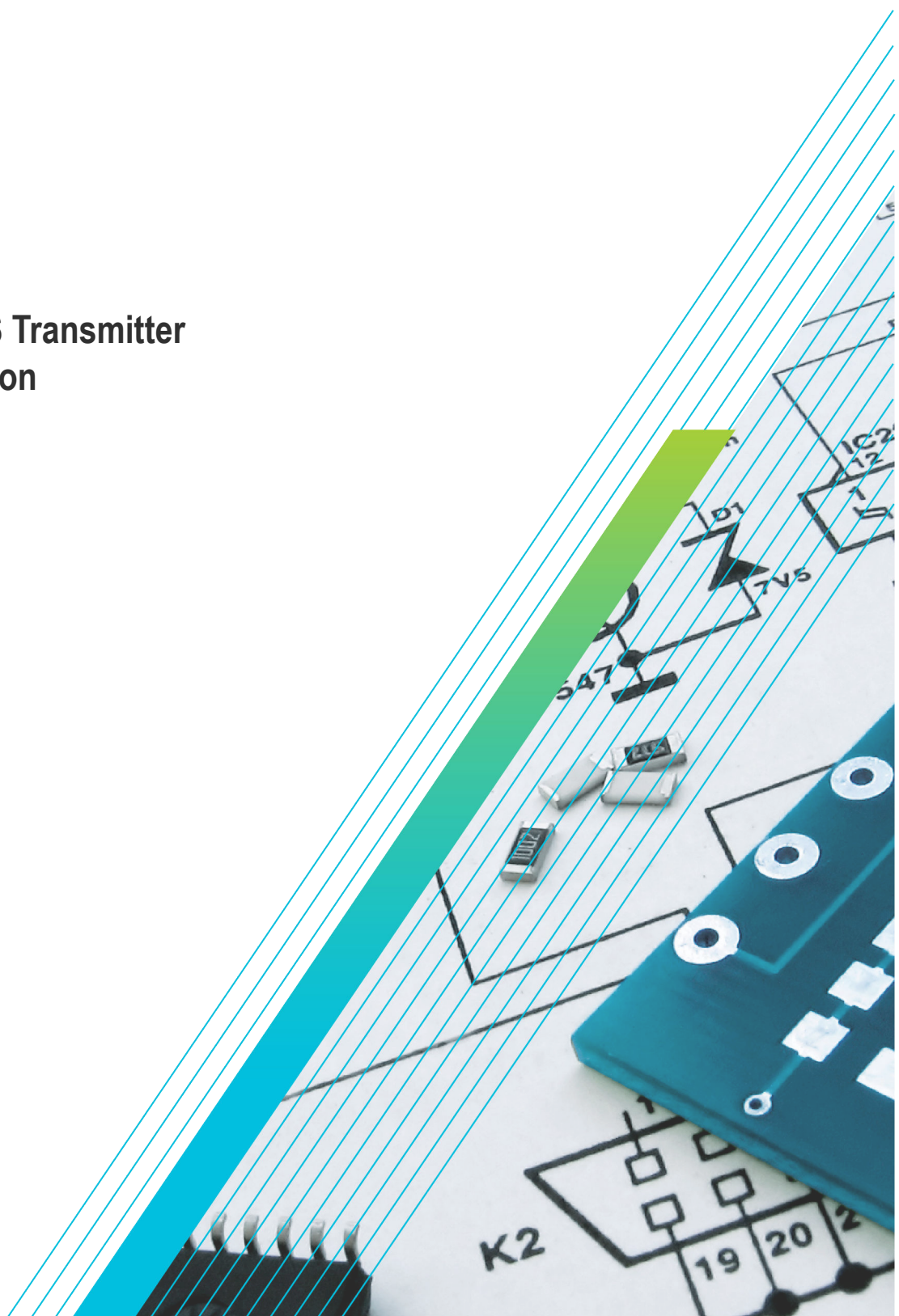




**TekExpress® SAS Transmitter**  
**Automated Solution**  
**Application Help**



077-0849-01





**TekExpress® SAS Transmitter  
Automated Solution  
Application Help**

**Register now!**  
**Click the following link to protect your product.**  
**[www.tek.com/register](http://www.tek.com/register)**



077-0849-01

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

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

## **Contacting Tektronix**

Tektronix, Inc.

14150 SW Karl Braun Drive

P.O. Box 500

Beaverton, OR 97077

USA

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

- In North America, call 1-800-833-9200.
- Worldwide, visit to [www.tek.com](http://www.tek.com) find contacts in your area.

# Table of Contents

|  |    |
|--|----|
| Welcome.....   | 9  |
| Getting help and support.....                            | 10 |
| Product documents.....                                   | 10 |
| Conventions.....   | 10 |
| Technical support.....                                   | 11 |
| Getting started.....                                     | 12 |
| Hardware requirements.....                               | 12 |
| Minimum system requirements.....                         | 12 |
| Supported oscilloscope models.....                       | 12 |
| Software requirements.....                               | 13 |
| Downloading and installing the software.....             | 13 |
| Activate the license.....                                | 13 |
| View software version and license key details.....       | 14 |
| Setting up the test environment.....                     | 15 |
| Connection diagram.....                                  | 15 |
| Search instruments connected to the application.....     | 15 |
| After the test is complete.....                          | 16 |
| Before you click Start.....                              | 19 |
| Pre-run checklist.....                                   | 20 |
| Starting the application.....                            | 22 |
| Application controls.....                                | 23 |
| Options menu functions.....                              | 24 |
| Configure email settings.....                            | 26 |
| Deskew.....  | 27 |
| TekExpress instrument control settings.....              | 28 |
| Setup panel: Configure the test setup.....               | 29 |
| DUT: Set DUT settings.....                               | 30 |
| Multiple-session run .....                               | 32 |
| Test Selection: Select the tests.....                    | 33 |
| Acquisitions: Set waveform acquisition settings.....     | 34 |
| Configuration: Set measurement limits for tests.....     | 36 |
| Preferences: Set the test run preferences.....           | 40 |
| Status panel: View the test execution status.....        | 41 |
| View test execution status.....                          | 41 |
| View test execution logs.....                            | 42 |
| Results panel: View summary of test results.....         | 43 |
| Filter the test results.....                             | 43 |
| Reports panel: Configure report generation settings..... | 44 |
| Select report generation options.....                    | 44 |
| Configure report view settings.....                      | 47 |
| View a generated report.....                             | 49 |
| Saving and recalling test setup.....                     | 50 |
| Overview.....  | 50 |
| Save the configured test setup.....                      | 50 |

---

|   |    |
|---|----|
| Load a saved test setup.....  | 51 |
| Perform a test using pre-run session files.....   | 52 |
| Save the test setup with a different name.....  | 56 |
| SCPI Commands.....  | 57 |
| About SCPI command.....   | 57 |
| Socket configuration for SCPI commands.....   | 57 |
| Set or query the device name of application.....  | 63 |
| Set or query the suite name of the application.....                                       | 63 |
| Set or query the test name of the application.....  | 64 |
| Set or query the version name of the application.....                                     | 64 |
| Set or query the general parameter values.....  | 65 |
| Set or query the acquire parameter values.....  | 76 |
| Set or query the analyze parameter values.....  | 76 |
| Query the available devices in the DUT panel of the application.....                      | 76 |
| Query the available suites for the selected device.....                                   | 77 |
| Query the list of available tests of the application.....                                 | 77 |
| Query the available version names of the application.....                                 | 77 |
| Query the list of available instruments based on the specified instrument type.....       | 78 |
| Set or query the IP address of the instrument based on the specified instrument type..... | 78 |
| Query the information of the generated report file.....                                   | 79 |
| Query the information of the generated waveform files.....                                | 79 |
| Query the information of the generated image files.....                                   | 79 |
| Query the active TekExpress application name.....   | 80 |
| Set or query the DUTID of application.....  | 80 |
| Sets or query the acquire mode status.....  | 80 |
| Set or query the execution mode status.....   | 81 |
| Generate the report for the current session.....  | 81 |
| Query the value of specified report header field in the report.....                       | 82 |
| Query the value of specified result detail available in report summary/details table..... | 82 |
| Restore the setup to default settings.....  | 83 |
| Save the setup.....   | 83 |
| Save the settings to a specified session.....   | 83 |
| Open the setup from a specified session.....  | 84 |
| Query the current setup file name.....  | 84 |
| Run/stop/pause/resume the selected measurements execution in the application.....         | 84 |
| Query the current measurement execution status.....                                       | 85 |
| Query whether the current setup is saved or not saved.....                                | 85 |
| Exit or close the application.....  | 85 |
| Query the status of the previous command execution.....                                   | 86 |
| Query the last error occurred.....  | 86 |
| Set or query the popup details.....   | 86 |
| Sets or query the limit values in the limits editor window.....                           | 87 |
| Set or query the waveform file recalled for the specified test name and acquire type..... | 87 |
| Set or query the enable/disable status of Verbose function.....                           | 88 |
| Query the enable or disable status of Continuous run function.....                        | 88 |
| Set or query the enable/disable status of Continuous Run function.....                    | 89 |
| Set or query the continuous run duration time value.....                                  | 89 |
| Set or query the session create option in the continuous run function.....                | 90 |
| Set or query the View report after generating option status.....                          | 90 |

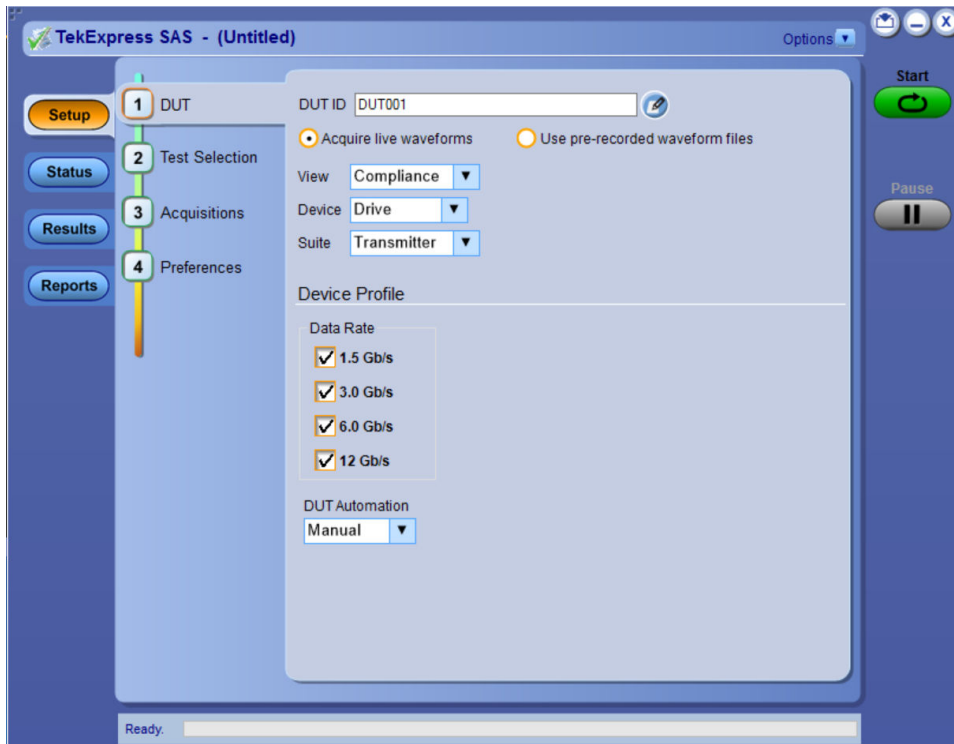
|   |     |
|---|-----|
| Returns the report as XML string.....   | 91  |
| Copies all the images from current run session to the given destination location..... | 91  |
| Selects the specified test(s) and deselect all other tests.....                       | 92  |
| Returns the complete information about the selected test.....                         | 92  |
| Set the default session.....  | 92  |
| Save the run/config sessions.....   | 92  |
| Load the run/config session.....  | 93  |
| Delete the run/config session.....  | 93  |
| Run the run/config saved session.....   | 93  |
| Query the available list in the run/config session.....                               | 94  |
| Query the current run/config session.....   | 94  |
| Override the run/config session.....  | 94  |
| About the programmatic interface.....   | 96  |
| Requirements for developing TekExpress client.....                                    | 96  |
| Remote proxy object.....  | 97  |
| Client proxy object.....  | 97  |
| Client programmatic interface overview.....   | 98  |
| Program remote access code example.....   | 100 |
| SAS application commands flow.....  | 101 |
| Connect through an IP address.....  | 105 |
| Lock the server.....  | 106 |
| Disable the popups.....   | 106 |
| Set or get the DUT ID.....  | 107 |
| Set the configuration parameters for a suite or measurement.....                      | 108 |
| Query the configuration parameters for a suite or measurement.....                    | 109 |
| Select a test.....  | 111 |
| Select a suite.....   | 111 |
| Select a channel.....   | 112 |
| Configure the selected measurement.....   | 113 |
| Run with set configurations or stop the run operation.....                            | 113 |
| Handle error codes.....   | 114 |
| Get or set the timeout value.....   | 115 |
| Wait for the test to complete.....  | 115 |
| After the test is complete.....   | 118 |
| Save, recall, or query a saved session.....   | 121 |
| Unlock the server.....  | 122 |
| Disconnect from the server.....   | 123 |
| SAS test definitions.....   | 124 |
| Maximum noise OOB IDLE.....   | 124 |
| OOB burst amplitude.....  | 124 |
| OOB offset delta.....   | 124 |
| OOB common mode delta.....  | 124 |
| SSC modulation type.....  | 125 |
| SSC modulation frequency.....   | 125 |
| SSC modulation deviation.....   | 125 |
| SSC modulation balance.....   | 125 |
| SSC DFDT.....   | 125 |
| Physical link rate long term stability.....   | 125 |
| Common mode RMS voltage.....  | 126 |

|   |     |
|---|-----|
| Common mode spectrum.....               | 126 |
| Peak-to-peak voltage.....               | 126 |
| VMA.....                                | 126 |
| Equalization.....                       | 126 |
| Rise time.....                          | 127 |
| Fall time.....                          | 127 |
| Random Jitter (RJ).....                 | 127 |
| Total Jitter (TJ).....                  | 127 |
| Waveform Distortion Penalty (WDP).....  | 127 |
| SAS3_EYEOPENING.....                    | 128 |
| Pre Cursor Equalization Ratio.....      | 128 |
| Post Cursor Equalization Ratio.....     | 128 |
| VHL (transition bit voltage Pk-Pk)..... | 128 |
| Unit Interval.....                      | 129 |
| References.....                         | 130 |
| Application directories.....            | 130 |
| File name extensions.....               | 130 |
| View test-related files.....            | 131 |
| Index.....                              | 132 |



# Welcome

Welcome to the TekExpress® SAS Transmitter Automated Solution application (Option SAS3-TSG). SAS TX provides an automated, simple, and efficient way to test SAS Transmitter interfaces and devices consistent to the requirements of the SAS Release Specification 05d. SAS is based on TekExpress version 3.0 Framework, the Tektronix Test Automation Framework developed to support your current and future test automation needs. TekExpress uses a highly modular architecture that lets you quickly deploy automated test solutions for various standards.



## Key Features

- Automated testing reduces the complexity of executing transmitter tests and enables you to test devices faster.
- Seamless debug allows pause on each test in automation, and switch to DPOJET analysis tool for detailed debug

# Getting help and support

## Product documents

Use the product documents for more information on the application functions, understand the theory of operation, how to remotely program or operate the application, and do other tasks.

**Table 1: TekExpress Application documents**




| To learn about                         | Use this document  |
|--|--|
| How to use the application             | TekExpress SAS Transmitter Automated Solution Help   |
| How to remotely control the instrument | PDF version of this document can be downloaded from <a href="http://www.tek.com/downloads">www.tek.com/downloads</a><br>Compiled HTML (CHM) version is integrated with the application. Press <b>F1</b> key from the keyboard to start the help.<br>Tektronix Part Number: 077-xxxx-xx |

## Conventions

This application help uses the following conventions:

- The term "Application," and "Software" refers to the TekExpress SAS Transmitter Automated Solution application.
- The term "DUT" is an abbreviation for Device Under Test.
- The term "select" is a generic term that applies to the two methods of choosing a screen item (button control, list item): using a mouse or using the touch screen.
- A **Note** identifies important information.

**Table 2: Icons used in the help**

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

## Technical support

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

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

### General information

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

### Application specific information

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

# Getting started

## Hardware requirements

### Minimum system requirements

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

**Table 3: System requirements**

| Instruments      | Description   |
|------------------|---|
| Oscilloscope     | Refer to <a href="#">Supported oscilloscopes models</a>   |
| Processor        | Same as the oscilloscope  |
| Operating System | Same as the oscilloscope  |
| Memory           | Same as the oscilloscope  |
| Hard Disk        | Same as the oscilloscope  |
| Display          | Same as the oscilloscope <sup>1</sup>   |
| Firmware         | TekScope v6.8.1.3 and above (Windows 7)   |
| Software         | <ul style="list-style-type: none"> <li>• Matlab Runtime engine v8.0 (Windows 7)</li> <li>• IronPython 2.7.3</li> <li>• PyVisa 1.0.0.25</li> <li>• Microsoft .NET 4.0 Framework</li> <li>• Microsoft Internet Explorer 6.0 SP1 or later</li> <li>• Adobe Reader 7.0 or equivalent software for viewing portable document format (PDF) files</li> </ul> |
| DPOJET           | v6.0.2 and above  |
| Other Devices    | <ul style="list-style-type: none"> <li>• Matched pair of SMA cables, one-set minimum for single lane</li> <li>• Microsoft compatible mouse or compatible pointing device</li> </ul>   |

### Supported oscilloscope models

The TekExpress SAS application runs on the following Tektronix oscilloscopes:

- DPO/DSA/MSO72004/C
- DPO/DSA/MSO71604/C
- DPO/DSA/MSO71254C
- DPO/MSO73304DX
- DPO/MSO72504DX
- DPO/MSO72304DX
- DPO/MSO72004DX
- DPO/MSO71604DX

<sup>1</sup> If TekExpress is running on an instrument having a video resolution lower than 800x600 (for example, sampling oscilloscope), it is recommended that you connect a secondary monitor. The secondary monitor must be configured and active before launching the application.

- DPO/MSO71254DX
- DPO/DSA73304D
- DPO/DSA72504D
- DPS77004SX
- DPS75904SX
- DPS75004SX
- DPO77002SX
- DPO75902SX
- DPO75002SX
- DPO73304SX
- DPO72504SX
- DPO72304SX
- DPO72004SX



**Note:** SAS Gen 3 (12 Gb/s) requires 20 GHz and above.

#### See also

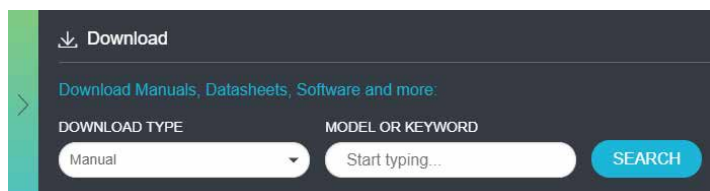
[Minimum system requirements](#)

## Software requirements

### Downloading and installing the software

Complete the following steps to download and install the latest TekExpress SAS Transmitter Automated Solution application.

1. Go to [www.tek.com](http://www.tek.com).
2. Click **Downloads**. In the Downloads menu, select DOWNLOAD TYPE as Software and enter the application name in the MODEL OR KEYWORD field and click **SEARCH**.



3. Select the latest version of software and follow the instructions to download the software. Copy the executable file into the oscilloscope.
4. Double-click the executable and follow the on-screen instructions.

The software is installed at C:\Program Files\Tektronix\TekExpress\TekExpress SAS Transmitter Automated Solution.

5. Select **Application > TekExpress SAS Transmitter Automated Solution** from the Oscilloscope menu, to open the application.

### 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.

## View software version and license key details

To view version information of the application, click **Options > About TekExpress**.



# Setting up the test environment

## Connection diagram

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

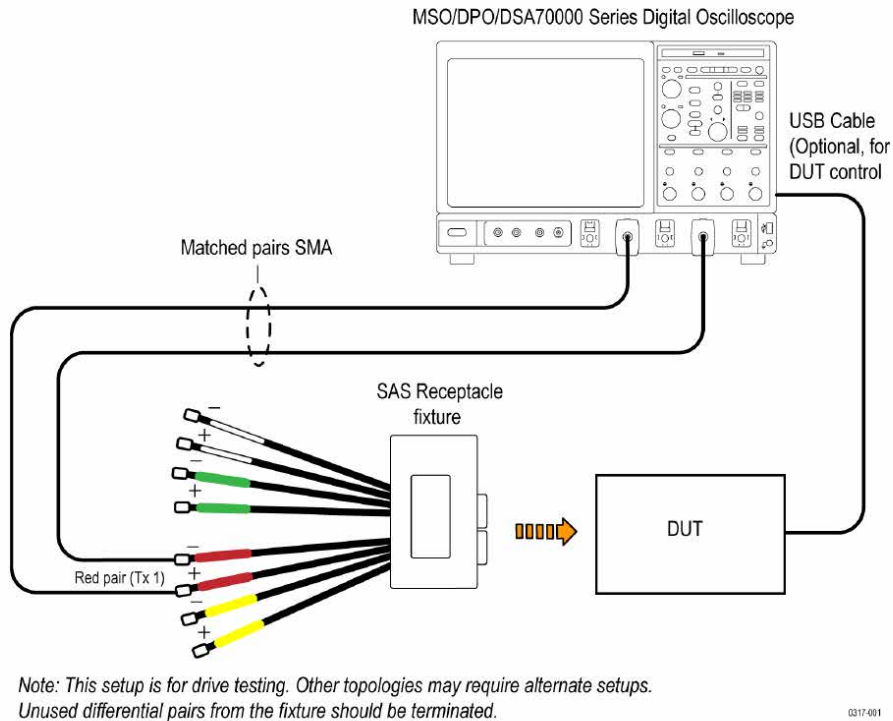


Figure 1: Device connection diagram

## Search instruments connected to the application

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

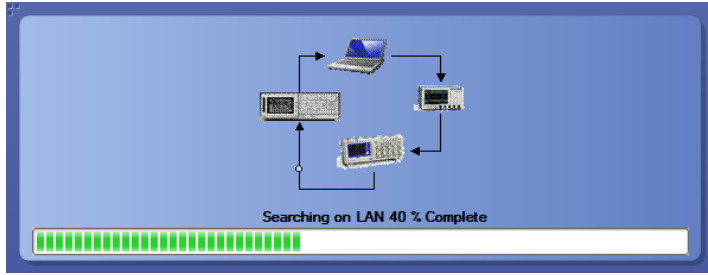


**Note:** The instruments required for the test setup must be connected and detected by the application, before running the test.

To refresh the list of connected instruments:

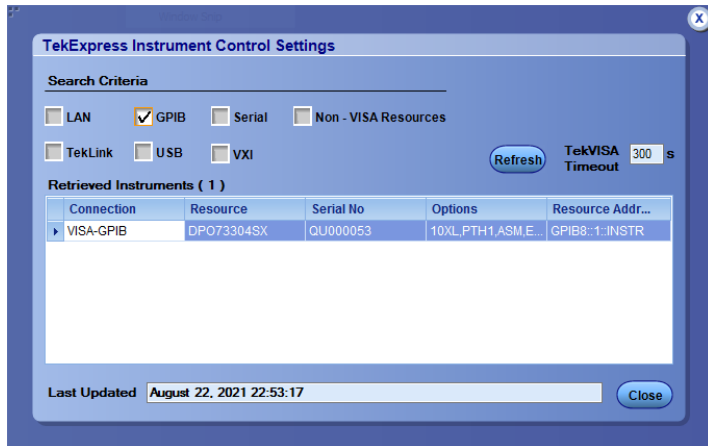
1. Select **Options > Instrument Control Settings**.
2. In the **Search Criteria** section of the **Instrument Control Settings** dialog box, select the connection types of the instruments to search. Instrument search is based on the VISA layer, but different connections determine the resource type, such as LAN, GPIB, and USB. For example, if you choose LAN, the search will include all the instruments supported by the TekExpress that are communicating over the LAN.
3. Click **Refresh**. The TekExpress application searches for the connected instruments.

**Search status of the instruments connected to LAN**



- When the search is complete, a dialog box lists the instrument-related details based on the search criteria. For example, for the Search Criteria as GPIB, the application displays all the GPIB instruments connected to the application.

**TekExpress Instrument Control Settings window.**



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.

**After the test is complete**


| Command name        | Parameters  | Description  | Return value   | Example   |
|---------------------|---|--|--|---|
| GetPassFailStatus() | <i>string clientID</i><br><i>string device</i><br><i>string suite</i><br><i>string test</i> | This method gets the pass or fail status of the measurement after test completion.<br><br><b>Note:</b><br> Execute this command after completing the measurement. | String that gives the status of the operation after it has been performed<br><br>Returns the pass or fail status in the form of a string | <pre> m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.GetPassFailStatus(clientID, device,suite, "5.3.9 - Random Jitter (RJ)")// Pass or Fail                     </pre> |

Table continued...




| Command name                        | Parameters   | Description  | Return value  | Example  |
|-------------------------------------|--|--|---|--|
| GetResultsValue()                   | <i>string clientID</i><br><i>string device</i><br><i>string suite</i><br><i>string test</i><br><i>string ParameterString</i>                     | This method gets the result values of the measurement after the run.             | String that gives the status of the operation after it has been performed<br><br>Returns the result value in the form of a string | <pre>m_Client = new Client()  //m_Client is a reference to the Client class in the Client DLL.  returnval as string  returnval=m_Client.GetPassFailStatus(clientID, device, suite, "5.3.9 - Random Jitter (RJ)")</pre> |
| GetResultsValueForSubMeasurements() | <i>string clientID</i><br><i>string device</i><br><i>string suite</i><br><i>string test</i><br><i>string ParameterString</i><br><i>int rowNr</i> | This method gets the result values for individual submeasurements after the run. | String that gives the status of the operation after it has been performed<br><br>Returns the result value in the form of a string | <pre>m_Client = new Client()  //m_Client is a reference to the Client class in the Client DLL.  returnval as string  <a href="#">Get results for a submeasurement example</a></pre>                                    |

Table continued...

| Command name         | Parameters  | Description  | Return value  | Example   |
|----------------------|---|--|---|---|
| GetReportParameter() | <p><i>string clientID</i></p> <p><i>string device</i></p> <p><i>string suite</i></p> <p><i>string test</i></p> <p><i>string ParameterString</i></p> | <p>This method gets the general report details such as oscilloscope model, TekExpress version, and SAS version.</p>  | <p>The return value is the oscilloscope model, TekExpress version, and SAS version.</p>   | <pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string</pre> <p><b>Oscilloscope Model</b></p> <pre>returnval=m_Client.GetReportParameter(clientID, "ScopeModel")</pre> <p><b>TekExpress Version</b></p> <pre>returnval=m_Client.GetReportParameter(clientID, "TekExpressVersion")</pre> <p><b>SAS Version</b></p> <pre>returnval=m_Client.GetReportParameter(clientID, "ApplicationVersion")</pre> |
| TransferReport()     | <p><i>string clientID</i></p> <p><i>string filePath</i></p>   | <p>This method transfers the report generated after the run.</p> <p>The report contains the summary of the run.</p> <p>The client must provide the location where the report is to be saved at the client-end.</p> | <p>String that gives the status of the operation after it has been performed</p> <p>Transfers all the result values in the form of a string</p> | <pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string</pre> <pre>returnval=m_Client.TransferReport(clientID, "C:\Report")</pre>   |

Table continued...

| Command name     | Parameters                                       | Description  | Return value   | Example   |
|------------------|--|--|--|---|
| TransferImages() | <i>string clientID</i><br><i>string filePath</i> | <p>This method transfers all the images (screenshots) from the specified client and folder for the current run (for a suite or measurement).</p> <p><b>Note:</b></p> <p>Every time you click Start, a folder is created in the X: drive. Transfer the waveforms before clicking Start.</p>  | <p>String that gives the status of the operation after it has been performed</p> <p>Transfers all the images in the form of a string</p> | <pre>m_Client = new Client()  //m_Client is a reference to the Client class in the Client DLL.  returnval as string  returnval=m_Client.TransferImages(clientID, "C:\Waveforms"</pre> |

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

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



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

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

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

**Table 4: string parameterString**

| Name            | Type   | Direction | Description   |
|-----------------|--------|-----------|---|
| parameterString | string | IN        | Specifies the oscilloscope model, TekExpress version, and SAS version |

### Get results for a submeasurement example

This example returns the specified submeasurement results for test 5.3.9 - Random Jitter (RJ).

```
returnval=m_Client.GetResultsValue( clientID,"Transmitter", "Drive", "5.3.9 - Random Jitter(RJ)", "Value", 0)
```

```
returnval=m_Client.GetResultsValue( clientID,"Transmitter", "Drive", "5.3.9 - Random Jitter(RJ)", "Value", 1)
```

### Before you click Start

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

1. Understand where your test files are stored on the instrument.

After you install and launch TekExpress SAS, it creates the following folders on the oscilloscope:

- \Program Files Tektronix\TekExpress\TekExpress SAS
- \My Documents\My TekExpress\SAS

- \My Documents\My TekExpress\SAS\Untitled Session

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



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

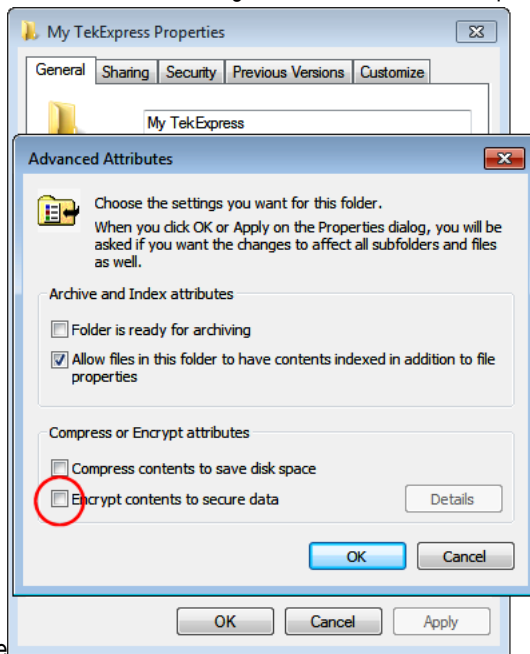
2. Map the shared My TekExpress folder as X: (X drive) on all instruments used in test setup running Microsoft Windows Operating System.

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



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

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



Example

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

## Pre-run checklist

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

On the oscilloscope:

- Make sure that all the required instruments are properly warmed up.
- Perform Signal Path Compensation (SPC).
  1. On the oscilloscope main menu, select the **Utilities** menu.

2. Select **Instrument Calibration** and run the **SPC utility**.

- Perform deskew on any cables.

In the SAS application:

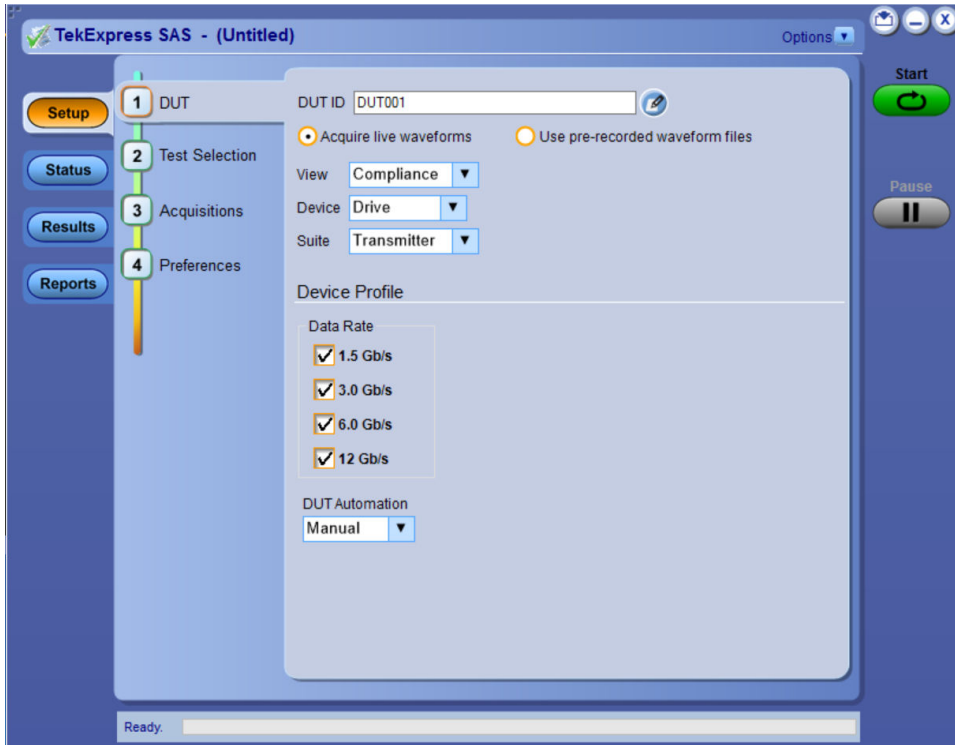
1. Verify that the application is able to find the DUT. If it cannot, perform a search for connected instruments.
  - a. Select **Setup > Test Selection**. Select any test and then click **Configure**.
  - b. In the Configuration section, click **Global Settings**.
  - c. In the **Instruments Detected**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

**See also**

[Equipment Connection Setup](#)

## Starting the application

To start the TekExpress SAS Transmitter, select from the oscilloscope menu bar **Applications > TekExpress SAS Transmitter Automated Solution**.



During start, a "My TekExpress" folder is created in the Documents folder of the current user and gets mapped to "X" drive. When the application is closed properly, the "X" drive gets unmapped. Session files are then stored inside the X:\SAS Transmitter folder. If this file is not found, the application runs an instrument discovery program to detect connected instruments before starting TekExpress SAS Transmitter Automated Solution.

To keep the TekExpress SAS Transmitter Automated Solution application on top of any application, select **Keep On Top** from the [options menu](#). If the application goes behind the oscilloscope application, select **Applications > TekExpress SAS Transmitter Automated Solution** to bring the application to the front.

## Application controls

This section describes the application controls with functionality and its details.

**Table 5: Application control description**


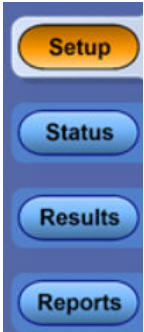










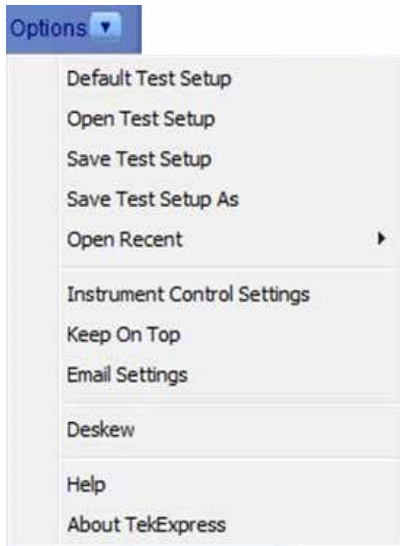
| Item  | Description   |
|---|---|
| <p><i>Options menu</i></p>             | Menu to display global application controls.  |
| <p>Test panel</p>                      | Controls that open tabs for configuring test settings and options.  |
| <p>Start / Stop button</p>            | Use the <b>Start</b> button to start the test run of the measurements in the selected order. If prior acquired measurements are not cleared, then new measurements are added to the existing set. The button toggles to the Stop mode while tests are running. Use the <b>Stop</b> button to abort the test.  |
| <p>Pause / Continue button</p>       | Use the <b>Pause</b> button to pause the acquisition. When a test is paused, this button changes as <b>Continue</b> .   |
| <p>Clear button</p>                  | <p>Use the <b>Clear</b> 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 <i>Results panel</i>.</p> <p> <b>Note:</b> This button is visible only when there are results data on the panel.</p> |
| <p>Application window move icon</p>  | Place the cursor over the top of the application window to move the application window to the desired location  |
| <p>Minimize icon</p>                 | Minimizes the application.  |
| <p>Close icon</p>                    | Close the application.  |

Table continued...

| Item  | Description   |
|---|---|
| Mini view / Normal view<br><br> | Mini view displays the run messages with the time stamp, progress bar, Start / Stop button, and Pause / Continue button. The application moves to mini view when you click the <b>Start</b> button. |

## Options menu functions

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



**Table 6: Options menu settings**

| Menu                               | Function   |
|------------------------------------|--|
| Default Test Setup                 | Opens a new test setup with default configurations.  |
| Open Test Setup                    | Opens a previously saved test setup. Displays the list of previously saved test setup file names. Make the selection and click <b>OK</b> to open the test setup. |
| Save Test Setup                    | Saves the current test configurations with the specified file name.  |
| Save Test Setup As                 | Saves the current test setup with a different file name or file type.  |
| Open Recent                        | Displays the recently opened test setup file names. Make the selection and click <b>OK</b> to open the test setup.   |
| <i>Instrument Control Settings</i> | Detects, lists, and refreshes the connected instruments found on the specified connections (LAN, GPIB, USB, Serial, Non-VISA Resources, TekLink, and VXI).       |
| Keep On Top                        | Always keeps the TekExpress SAS Transmitter application on top of all the applications.  |
| <i>Email Settings</i>              | Configures email options for test run and result notifications.  |
| Deskew                             | Loads oscilloscope channel deskew settings into the application.   |
| Help                               | Displays the TekExpress SAS Transmitter Automated Solution help.   |

Table continued...



---

| Menu             | Function   |
|------------------|--|
| About TekExpress | Displays the application name, version, and hyperlink to end the user license agreement. |

## Configure email settings

Use the **Email Settings** utility to get notified by email when a measurement completes or produces any error condition. Follow the steps to configure email settings:



Figure 2: Email settings window

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

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



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

5. In the **Email Attachments** section, select from the following options:
  - **Reports:** Select to receive the test report with the notification email.
  - **Status Log:** Select to receive the test status log with the notification email. If you select this option, then also select whether you want to receive the full log or just the last 20 lines.
6. In the **Email Configuration** section:
  - Enter a maximum file size for the email message. Messages with attachments larger than this limit will not be sent. The default is 0 MB.
  - Enter the number in the Number of Attempts to Send field, to limit the number of attempts that the system makes to send a notification. The default is 1. You can also specify a timeout period.
7. Select the **Email Test Results When complete or on error** check box. Use this check box to quickly enable or disable email notifications.
8. To test your email settings, click **Test Email**.
9. To apply your settings, click **Apply**.
10. Click **Close** when finished.

## Deskew

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



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



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



**Note:** SAS does not verify that the saved oscilloscope settings and deskew values are applicable to the current instrument attached to the application.

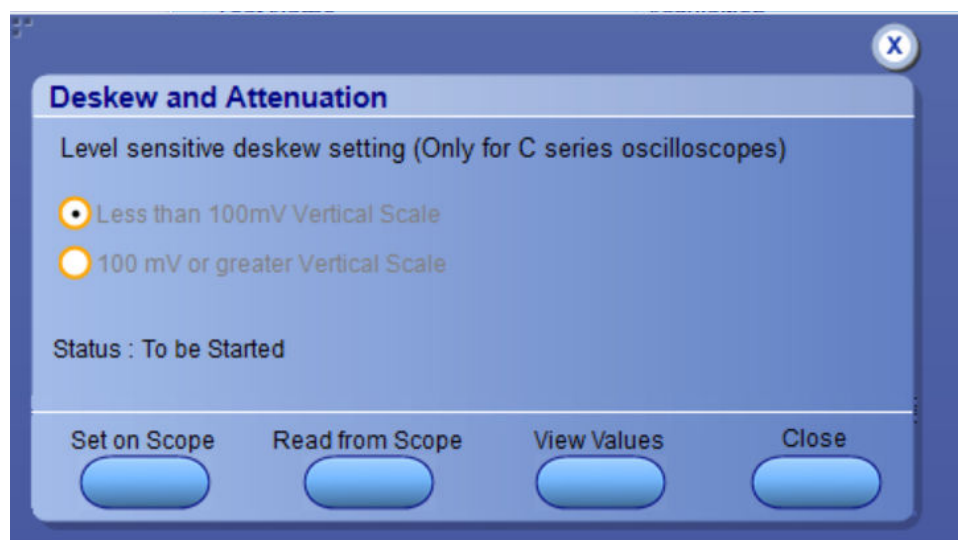


Figure 3: Deskew

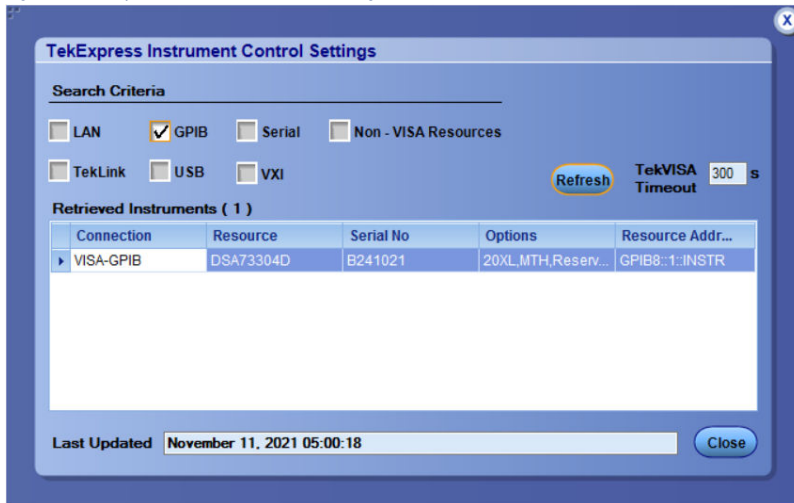
1. In the TekExpress SAS application, select **Options > Deskew**.
2. Select the level sensitive deskew setting:
  - **Less than 100 mV Vertical Scale:** Select this if the oscilloscope vertical setting is less than 100 mV/division for the signal you are measuring.
  - **100 mV or greater Vertical Scale:**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

## TekExpress instrument control settings

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

To access, click **Options > Instrument Control Settings**. Select **GPIB** as search criteria for TekExpress application and click **Refresh**. The connected instruments displayed in the Retrieved Instruments window and can be selected for use under Global Settings in the test configuration section.

Figure 4: TekExpress Instrument Control Settings window

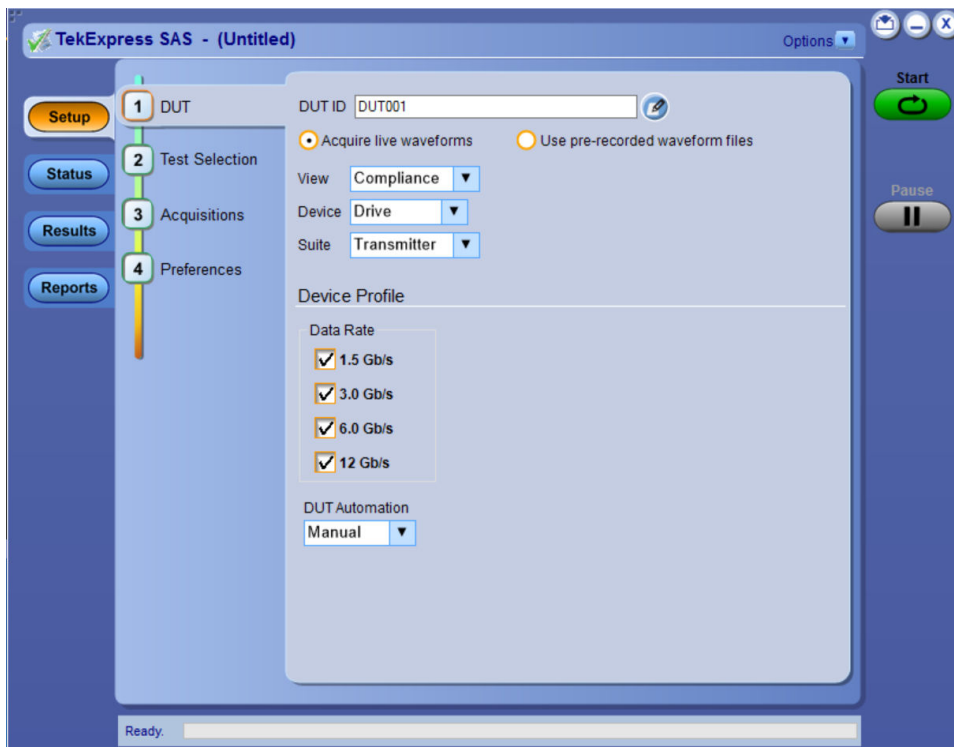


### See also


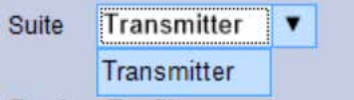

[Options menu functions](#) on page 24

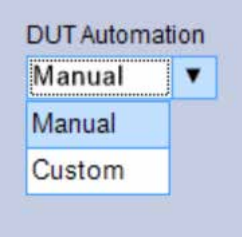
## Setup panel: Configure the test setup

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






| Setting        | Description   |
|----------------|---|
|                | <ul style="list-style-type: none"> <li>• <b>Compliance:</b> View configuration settings by clicking <b>Setup &gt; Test Selection &gt;Configure</b></li> <li>• <b>Advanced:</b> Enables the <b>Setup &gt; Configuration</b> tab in which to view configuration settings.</li> </ul>  |
| Device         | <p>There are two options to select the device.</p> <ul style="list-style-type: none"> <li>• Drive</li> <li>• Host</li> </ul>    |
| Suite          | <p>There is only one default option to select the suite name.</p> <ul style="list-style-type: none"> <li>• Transmitter</li> </ul>   |
| Data Rates     | <p>Select the data rates to include in the tests. Minimum one data rate should to selected to run any test.</p>   |
| DUT Automation | <p>The DUT Automation field displays the type of DUT control currently in use. DUT controls are detected by through the Instrument Control Settings feature.</p> <p>There are two options for DUT Automation:</p> <ul style="list-style-type: none"> <li>• <b>Manual:</b> This option is displayed if your setup does not have the capability to control the DUT automatically. You will be prompted to configure the DUT manually during the test. You can change this to Custom if needed.</li> <li>• <b>Automated:</b> Select this option if you are using the custom tool to manage the DUT.</li> </ul> |

| Setting | Description  |
|---------|--|
|         |  |

## Multiple-session run

Multiple-sessions run feature allows you to save multiple config sessions and run multiple config/run sessions together.



Click (  ) button in the DUT panel, displays the Run/Config Session window. The Run/Config session window provides the list of saved sessions and the ability to run selected sessions.

- Config session – Session saved by user manually from Run/Config Sessions window.
- Run session – Session created automatically after the test is executed.
- **Session Name:** Enter the name to save the config session. The maximum number of character supported is 40 and special characters (.,,.,.,.,.,\,/:?"<>|\*) are not supported.
- **Save:** Save current configuration as a session with the given session name.
- **Close:** Close the Run/Config Session window.
- **Default:** Sets the application configurations to default values.
- **Load:** Load the selected config/run session.
- **Delete:** Delete the selected config/run session.
- **Run Sessions:** Run the selected config/run session.

### Enable/ Disable the Multi Run session

By default the **Multi Run Session** is enabled in the application. Set the IsMultiSessionRunEnabled value to false to disable the **Multi Run Session** feature in the TekExpress.exe.Config file, which is downloaded along the application.



## Test Selection: Select the tests

Use the Test Selection tab to select the tests. The test measurements available depends on the settings selected in the DUT tab.

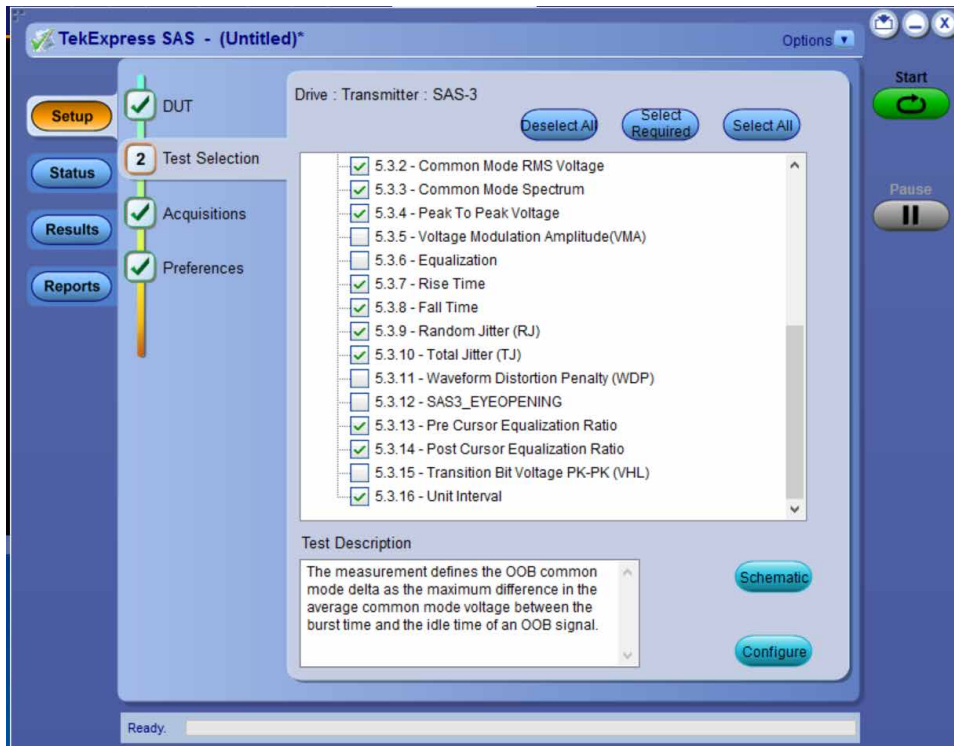


Figure 6: Test selection tab

Table 8: Test Selection tab configuration

| Setting          | Description  |
|------------------|--|
| Deselect All     | Deselect or select all tests in the list.  |
| Select All       |  |
| Select Required  |  |
| Tests            | Click on a test to select or unselect. Highlight a test to show details in the Test Description panel.   |
|                  | The application automatically selects all required tests when in Compliance mode.  |
| Test Description | Shows brief description of the highlighted test in the test tree.  |
| Schematic        | Shows an equipment and test fixture setup schematic (connection diagram) for the selected test. Use to set up the equipment and fixtures or to verify the setup before running the test.       |
| Configure        | When the View type is selected in the DUT tab is Compliance, this button opens the configuration section for the selected test. If the View type is Advanced, this button is not displayed. in |

## Acquisitions: Set waveform acquisition settings

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

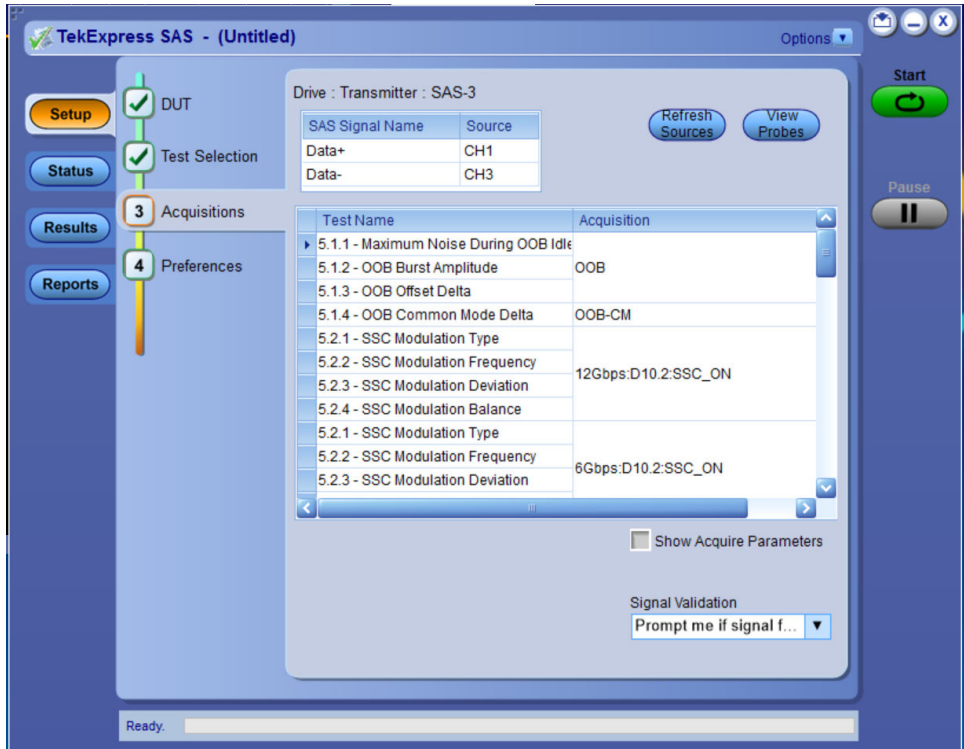
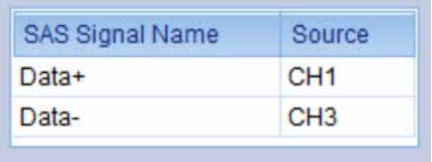


Figure 7: Acquisition tab

Table 9: Acquisitions tab configuration

| Setting                 | Description  |
|-------------------------|--|
| Source Selection        | Lists the signal type and input channel assigned to that type.<br>Click Source fields to assign a channel source to a signal type.<br> |
| Test Name               | Displays the name of the selected test to which the acquisitions apply. One or more tests can perform the same acquisitions.   |
| Acquisition             | The acquisition of the waveform for the corresponding test   |
| View Probes             | View the detected probe configuration. Use the View Probes dialog box to view the connected probes.  |
| Show Acquire Parameters | Select to view the acquisition parameters.   |

TekExpress SAS Transmitter Automated Solution application saves all acquisition waveforms to files by default. Waveforms are saved in a unique folder for each session (a session is started when you click the Start button). The folder path is X:\TekExpress SAS Transmitter Automated Solution\Untitled Session\<>dutid>\<date>\_<time>. Images created for each analysis, XML files with result values, reports, and other information specific to that particular execution are also saved in this folder.

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

## Configuration: Set measurement limits for tests

Use Configuration tab to view and configure the Global Settings and the measurement configurations. The measurement specific configurations available in this tab depends on the selections made in the DUT panel and Test Selection panel.

Table 10: Configuration tab: Common parameters

| Settings     | Description  |
|--------------|--|
| Limit Editor | Displays the upper and lower limits for the applicable measurement using different types of comparisons. |

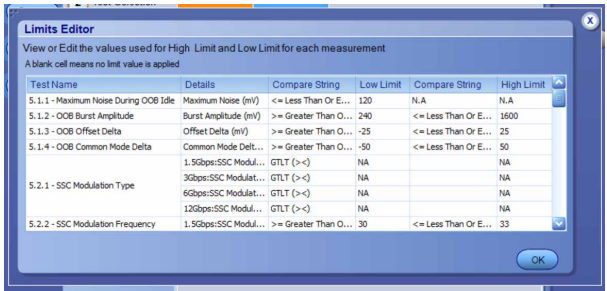


Figure 8: Configuration tab: Global Settings

Table 11: Configuration tab: Global Settings configuration

| Setting            | Description |
|--------------------|-------------|
| Global Settings    |             |
| Table continued... |             |


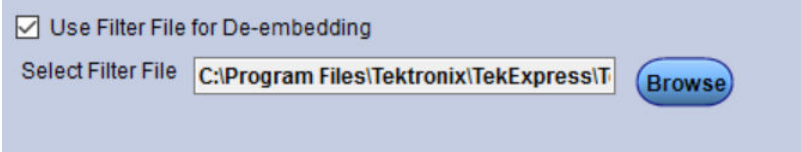
| Setting                          | Description  |
|----------------------------------|--|
| Compliance Mode                  | Select to use Compliance Mode values.<br>You cannot change most test parameter values in Compliance mode but you can view the parameters.  |
| User Defined Mode                | Select to run tests with custom parameter values.  |
| Instruments Detected             | Displays the instruments connected to this application. Click on the instrument name to open a list of available (detected) instruments. Select <b>Options &gt; Instrument Control Settings</b> to refresh the connected instrument list refer TekExpress instrument control settings. |
| <b>Device Control Script</b>     |  |
| Pattern                          | In the <b>Pattern</b> drop-down menu, you will notice different pattern types used in the application. Select each of the patterns and browse for the script file. The application will remember the path until you close the application.   |
| Script path                      | Browse and select the script path.<br>   |
| Use Filter file for De-embedding | Enable or disable the check box to browse and select de-embedding filter file path.  |
| Select Filter file               | Browse and select the filter file path.<br>  |



Figure 9: Configuration tab: Measurement

Table 12: Configuration tab: Measurements configuration

| Setting                   | Description   |
|---------------------------|---|
| <b>Measurements</b>       |   |
| Measurements              | Displays the measurement groups, that are selected in the Test Selection tab. Select the respective test group to view or modify the measurement configuration.   |
| <b>Scope Settings</b>     |   |
| Pattern                   | In the <b>Pattern</b> drop-down menu, you will notice different pattern types used in the application. Select each of the patterns and browse for the script file. The application will remember the path until you close the application.<br>Default Value:OOB |
| Horizontal scale (us/div) | Displays horizontal scale value.<br>Default Value: 10   |
| Trigger slope             | Select trigger slo<br>Default Value: Rise   |
| Resolution (ps/pt)        | Displays<br>Default Value: 20   |

Table continued...

---

| Setting                 | Description  |
|-------------------------|--|
| Vertical Position (Div) | Description<br>Default Value: 0                        |
| Sampling Rate(Gs/s)     | Description<br>Default Value: 50 Gs/s                  |
| Trigger Level(mV)       | Displays trigger level value<br>Default Value: -100 mV |

## Preferences: Set the test run preferences

Use **Preferences** tab to set the application action on completion of a measurement. The **Preferences** tab has the feature to enable or disable certain options related to the measurement execution.

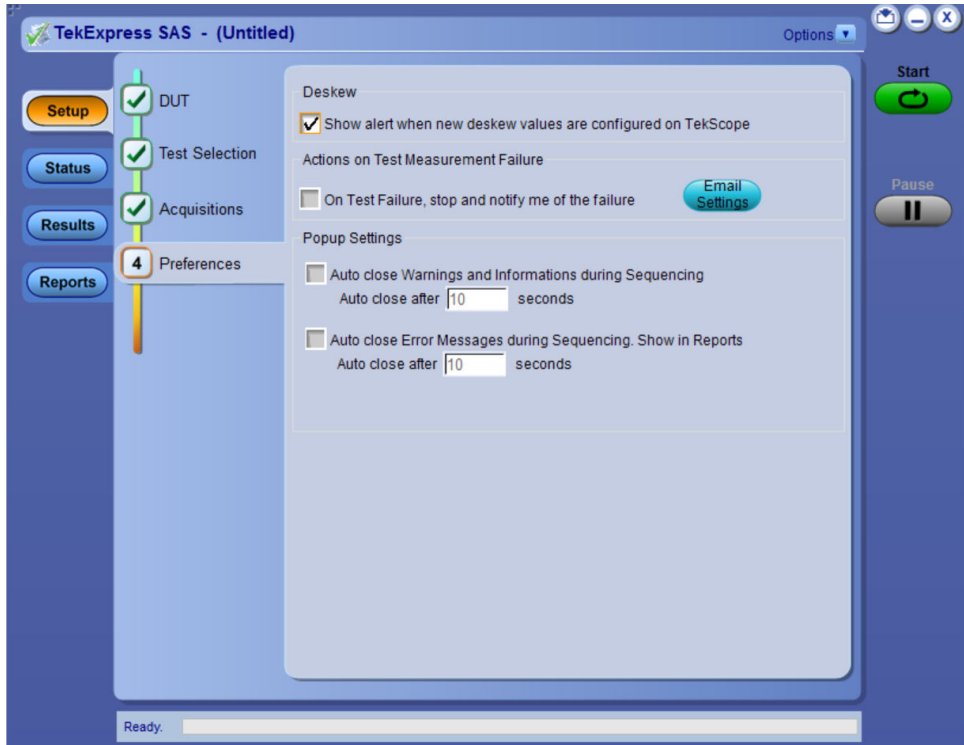


Figure 10: Preferences tab

Refer the below table for the options available in the **Preferences** tab:

**Table 13: Preferences tab settings**

| Setting   | Description   |
|---|---|
| <b>Deskew</b>   |   |
| Show alert when new deskew values are configured on TekScope                                  |   |
| <b>Actions on Test Measurement Failure</b>  |   |
| On Test Failure, stop and notify me of the failure  | Select to stop the test run on Test Failure, and to get notified via email. By default, it is unselected. Click <b>Email Settings</b> to configure the email settings to receive notifications. |
| <b>Popup Settings</b>   |   |
| Auto close Warnings and informations during Sequencing<br>Auto close after <no> seconds       | Select to close the warnings and information window automatically after the specified amount of time.<br>Specify the time in seconds using the edit box.  |
| Auto close Error Messages during Sequencing. Show in Reports<br>Auto close after <no> seconds | Select to close the error message window automatically after the specified amount of time.<br>Specify the time in seconds using the edit box.   |



# Status panel: View the test execution status

The Status panel contains the **Test Status** and **Log View** tabs, which provides status on the test acquisition and analysis (Test Status) and listing of test tasks performed (Log View tab). The application opens the **Test Status** tab when you start to execute the test. Select the **Test Status** or the **Log View** tab to view these items while the test execution is in progress.

## View test execution status

The tests are grouped and displayed based on the Clock and Data lane. It displays the tests along with the acquisition type, acquire, and analysis status of the tests. In pre-recorded mode, **Acquire Status** is not valid.

The **Test Status** tab presents a collapsible table with information about each test as it is running. Use the symbols to expand (+) and collapse (-) the table rows.

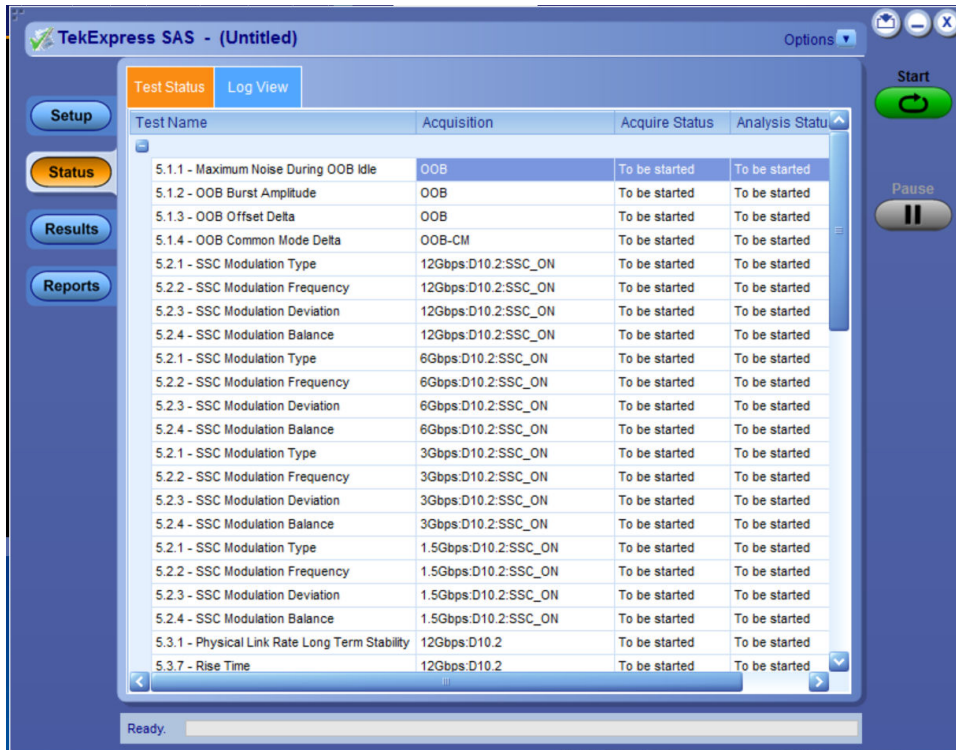


Figure 11: Test execution status view in Status panel

Table 14: Test execution status table headers

| Control        | Description   |
|----------------|---|
| Test Name      | Displays the measurement name.  |
| Acquisition    | Describes the type of data being acquired.  |
| Acquire Status | Displays the progress state of the acquisition: <ul style="list-style-type: none"> <li>To be started</li> <li>In Progress</li> <li>Testing</li> </ul> |

Table continued...

Status panel: View the test execution status

| Control         | Description  |
|-----------------|--|
| Analysis Status | <p>Displays the progress state of the analysis:</p> <ul style="list-style-type: none"> <li>• To be started</li> <li>• In Progress</li> <li>• Testing</li> <li>• Completed</li> </ul> |

## View test execution logs

The Test Status tab displays the detailed execution status of the tests. Also, displays each and every execution step in detail with its timestamp information. The log details can be used to troubleshoot and resolve any issue/bug which is blocking the test execution process.

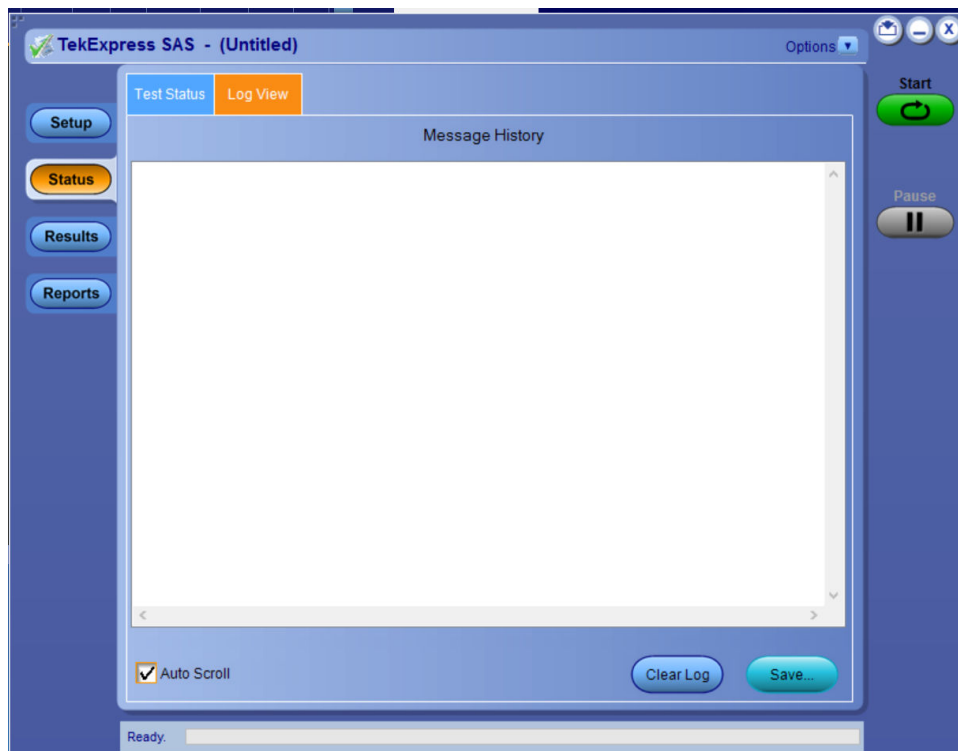


Figure 12: Log view in Status panel

Table 15: Status panel settings

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

## Results panel: View summary of test results

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

In the Results table, each test result occupies a row. By default, results are displayed in summary format with the measurement details collapsed and with the Pass/Fail column visible.

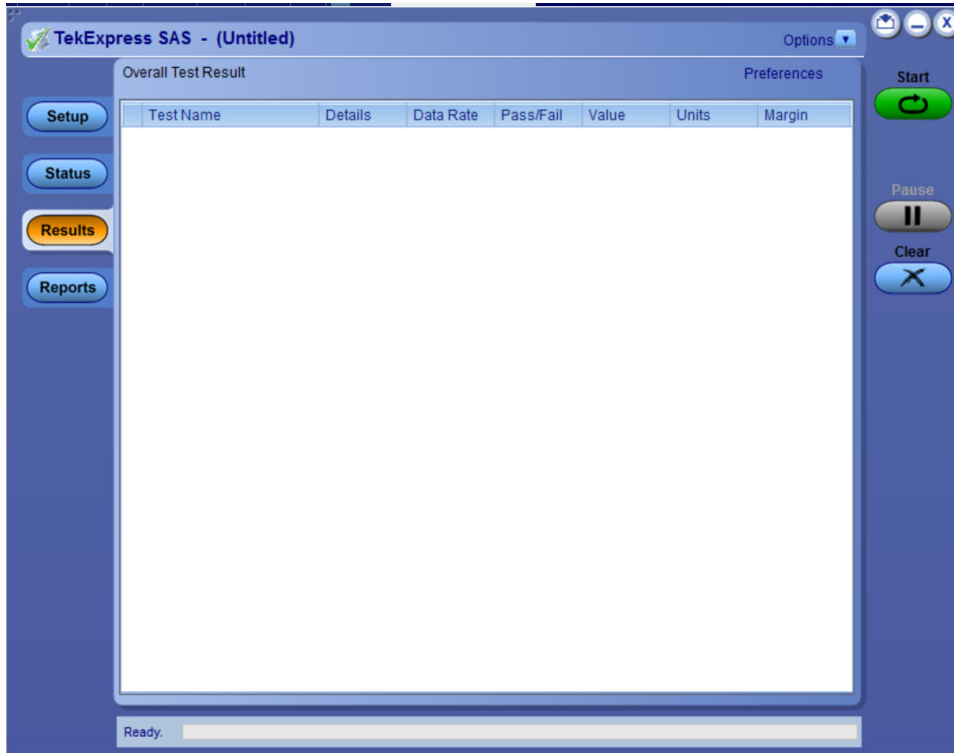



Figure 13: Results panel with measurement results

Click  icon on each measurement in the row to expand and to display the minimum and maximum parameter values of the measurement.

## Filter the test results

Each column in the result table can be customized and displayed by enabling or disabling any column as per your requirement. You can change the view in the following ways:

- To remove or restore the Pass/Fail column, select **Preferences > Show Pass/Fail**.
- To collapse all expanded tests, select **Preferences > View Results Summary**.
- To expand all the listed tests, select **View Results Details** from the **Preferences menu** in the upper right corner.
- To enable or disable the wordwrap feature, select **Preferences > Enable Wordwrap**.
- To view the results grouped by lane or test, select the corresponding item from the **Preferences menu**.
- 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**.

# Reports panel: Configure report generation settings

Click **Reports** panel to configure the report generation settings and select the test result information to include in the report. You can use the Reports panel to configure report generation settings, select test content to include in reports, generate the report, view the report, browse for reports, name and save reports, and select report viewing options.

## Select report generation options

This section describes the report generation settings you can configure in the Reports panel. Select report settings before running a test or when creating and saving test setups. Report settings configured are included in saved test setups.

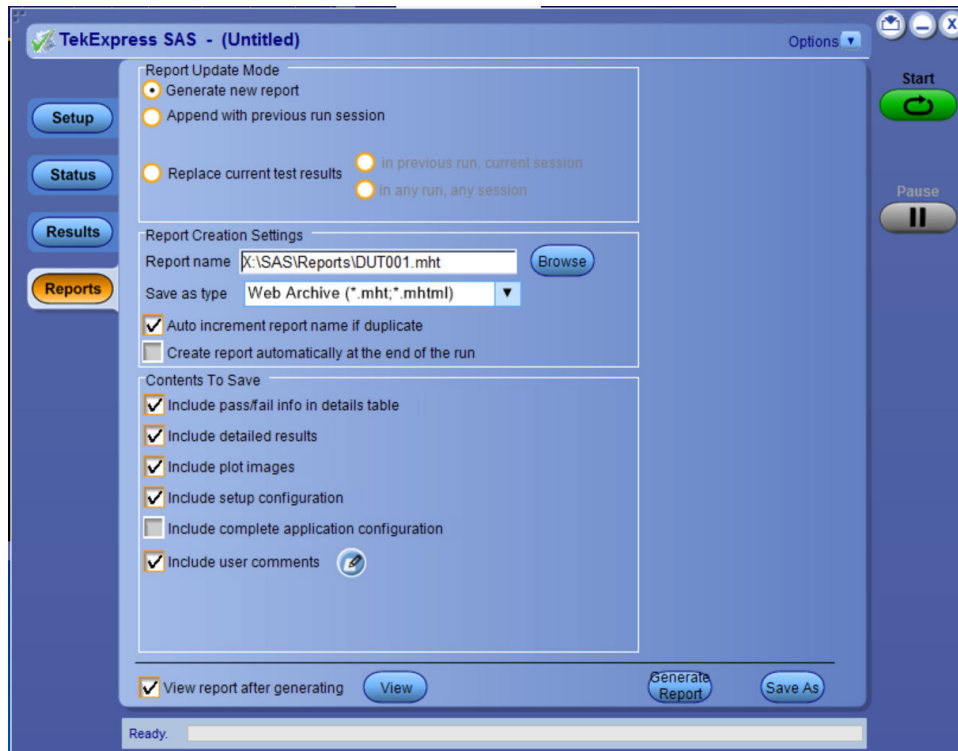




Figure 14: Reports panel

## Report Update Mode Settings

Table 16: Report Update Mode Settings

| Control                          | Description   |
|----------------------------------|---|
| Generate new report              | Each time when you click <b>Run</b> and when the test execution is complete, it will create a new report. The report can be in either .mht, .pdf, or .csv file formats.   |
| Append with previous run session | Appends the latest test results to the end of the current test results report. Each time when you click this option and run the tests, it will run the previously failed tests and replace the failed test result with the new pass test result in the same report. |
| Replace current test results     | Replaces the previous test results with the latest test results. Results from newly added tests are appended to the end of the report.  |

Table continued...

| Control                                 | Description   |
|---|---|
| In previous run, current session        | Select to replace current test results in the report with the test result(s) of previous run in the current session.  |
| In any run, any session                 | Select to replace current test results in the report with the test result(s) in the selected run session's report. Click and select test result of any other run session.   |
| <b>Report Creation Settings</b>         |   |
| Report name                             | <p>Displays the name and path of the TekExpress SAS Transmitter Automated Solution report. The default location is at <code>\My Documents&gt;\My TekExpress\SAS Transmitter\Reports</code>. The report file in this folder gets overwritten each time you run a test unless you specify a unique name or select to auto increment the report name.</p> <p><b>To change the report name or location, do one of the following:</b></p> <ul style="list-style-type: none"> <li>In the Report Path field, type the current folder path and name.</li> <li>Double-click in the Report Path field and then make selections from the popup keyboard and click <b>Enter</b>.</li> </ul> <p>Be sure to include the entire folder path, the file name, and the file extension. For example: <code>C:\Documents and Settings\your user name\My Documents\My TekExpress\SAS Transmitter\DUT001.mht</code>.</p> <p> <b>Note:</b> You cannot set the file location using the Browse button.</p> <p><b>Open an existing report</b></p> <p>Click <b>Browse</b>, locate and select the report file and then click <b>View</b> at the bottom of the panel.</p> |
| Save as type                            | <p>Saves a report in the specified file type, selected from the drop-down list. The report is saved in .csv, .pdf, or .mht.</p> <p><b>Note:</b></p> <p> If you select a file type different from the default, be sure to change the report file name extension in the Report Name field to match.</p>  |
| Auto increment report name if duplicate | Sets the application to automatically increment the name of the report file if the application finds a file with the same name as the one being generated. For example: DUT001, DUT002, DUT003. This option is enabled by default.  |
| <b>Contents To Save Settings</b>        |   |
| Include pass/fail info in details table | Select to include pass/fail information in the details table of the report.   |
| Include detailed results                | Select to include detailed results in the report.   |
| Include plot images                     | Select to include the plot images in the report.  |
| Table continued...                      |   |

| Control                               | 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 have added in the DUT tab of the Setup panel. Comments appear in the Comments section, below the summary box at the beginning of each report.   |
| <b>Other settings in report panel</b> |  |
| View report after generating          | Automatically opens the report in a Web browser when the test execution is complete. This option is selected by default.   |
| Generate Report                       | Generates a new report based on the current analysis results.  |
| <b>Group By</b>                       |  |
| Test Name                             | Select to group the test results based on the test name in the report.   |

## Configure report view settings

The **View Settings** tab describes the report view settings to configure the Reports panel. Select report view settings before running a test or when creating and saving test setups. Report settings configured are included in saved test setups.

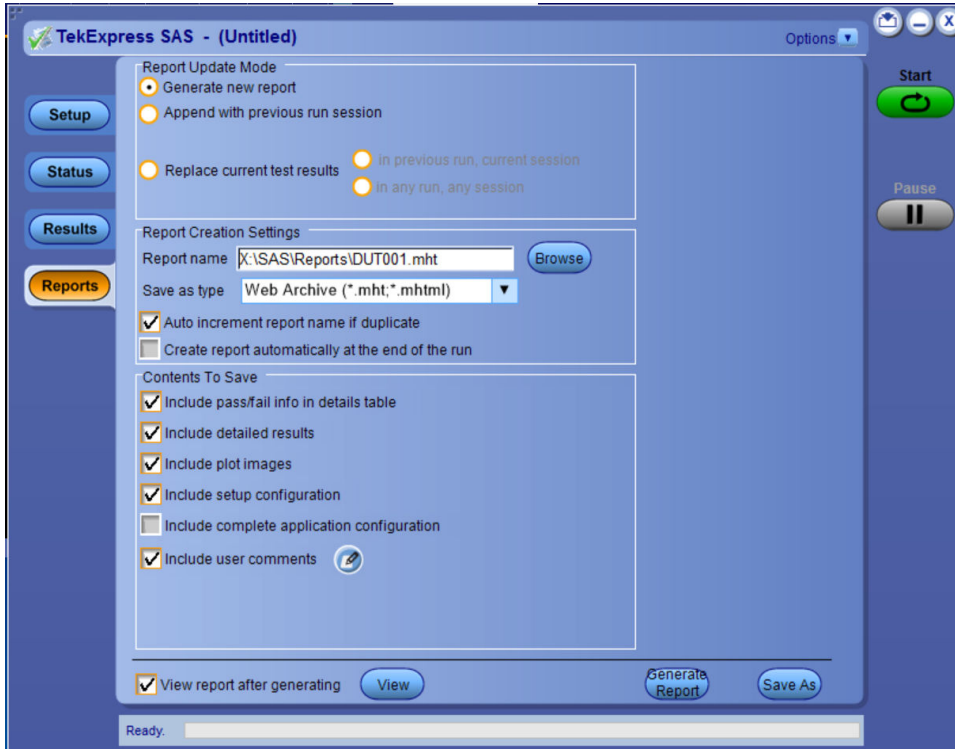


Figure 15: Report panel-View settings tab

Table 17: Report panel view settings

| Control                                 | Description  |
|---|--|
| <b>Contents To Save Settings</b>        |  |
| Include pass/fail info in details table | Select to include pass/fail information in the details table of the report.  |
| Include detailed results                | Select to include detailed results in the report.  |
| Include plot images                     | Select to include the plot images in the report.   |
| Include setup configuration             | Sets the application to include hardware and software information in the summary box at the top of the report. Information includes: the oscilloscope model and serial number, the oscilloscope firmware version, and software versions for applications used in the measurements. |
| Include user comments                   | Select to include any comments about the test that you or another user have added in the DUT tab of the Setup panel. Comments appear in the Comments section, below the summary box at the beginning of each report.   |
| Include switch matrix configuration     | Select to include the switch matrix configuration in the report.   |
| <b>Group Report By</b>                  |  |
| <b>Group Test Result By</b>             |  |
| Test Name                               | Select to group the test results based on the test name in the report..  |

Table continued...

Reports panel: Configure report generation settings

---

| Control           | Description  |
|-------------------|--|
| Lane              | Select to display the test results by lane.  |
| Equalization      | Select to display the test results by equalization.  |
| Pass/Fail Results | Select to display the test results by pass or fail results.  |
| Test Result       | Select to group the test results based on the test result in the report.   |
| <b>User logo</b>  |  |
| User logo         | Adds user logo into the generated report.  |
| Include user logo | Select to add your logo in the generated report. When selected, specify the logo file path in the Image file path option. Click <b>browse</b> and select the logo image. |



# View a generated report

## Sample report and its contents

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

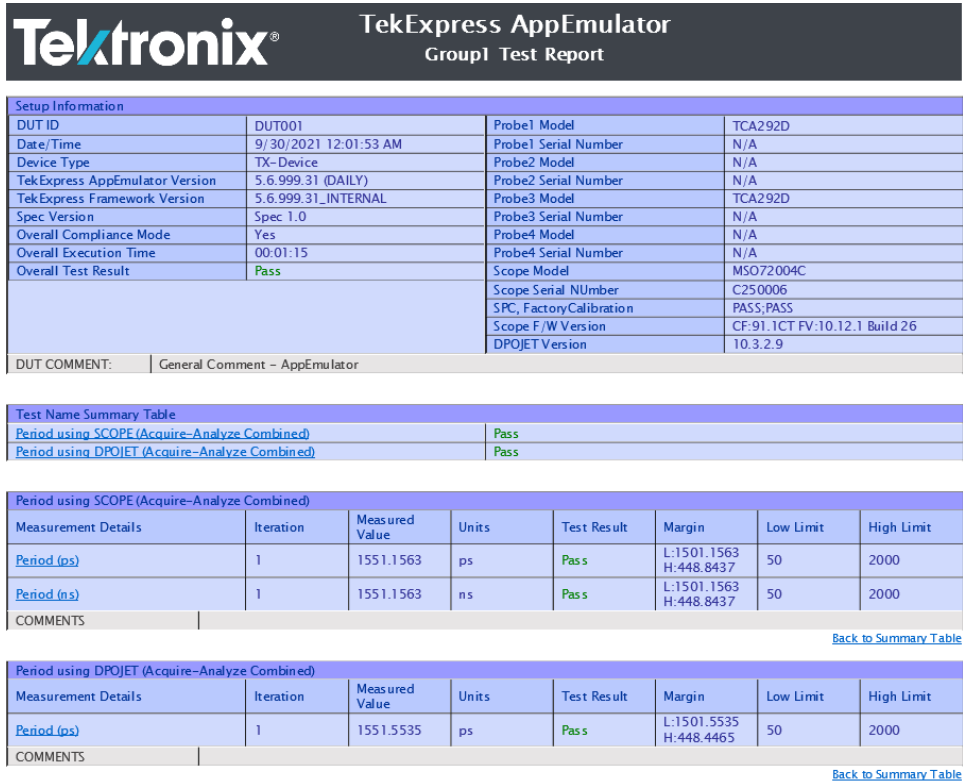


Figure 16: Report

### Setup Information

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

### Test Name Summary Table

The test summary table lists all the tests which are executed with its result status.

### Measurement

The measurement table displays the measurement related details with its parameter value.

### User comments

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

# Saving and recalling test setup

## Overview

You can save the test setup and recall it later for further analysis. Saved setup includes the selected oscilloscope, general parameters, acquisition parameters, measurement limits, waveforms (if applicable), and other configuration settings. The setup files are saved under the setup name at **X:\TekExpress SAS Transmitter Automated Solution**

| Name                 | Date modified      | Type        |
|----------------------|--------------------|-------------|
| 1-LP_20210331_210911 | 3/31/2021 9:06 PM  | File folder |
| 1-LP_20210331_220738 | 3/31/2021 10:05 PM | File folder |
| 1-LP_20210331_223715 | 3/31/2021 10:35 PM | File folder |
| 1-LP_20210331_224851 | 3/31/2021 10:48 PM | File folder |
| 1-LP_20210331_230337 | 3/31/2021 11:02 PM | File folder |
| 1-LP_20210331_230921 | 3/31/2021 11:08 PM | File folder |

Figure 17: Example of Test Setup File

Use test setups to:

- Recall a saved configuration.
- Run a new session or acquire live waveforms.
- Create a new test setup using 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.

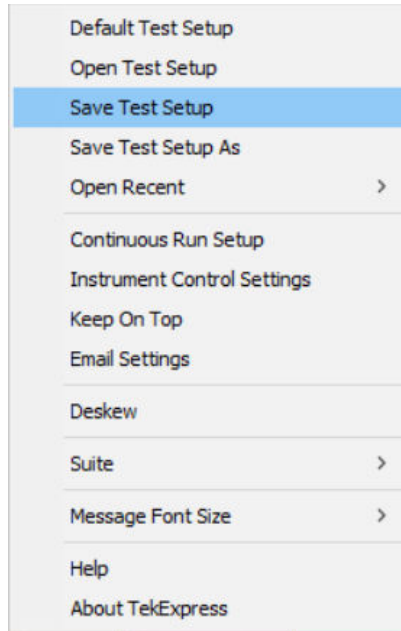


**Note:** Images that are shown in this Saving and recalling test setup chapter are for illustration purpose only and it may vary depending on the TekExpress application.

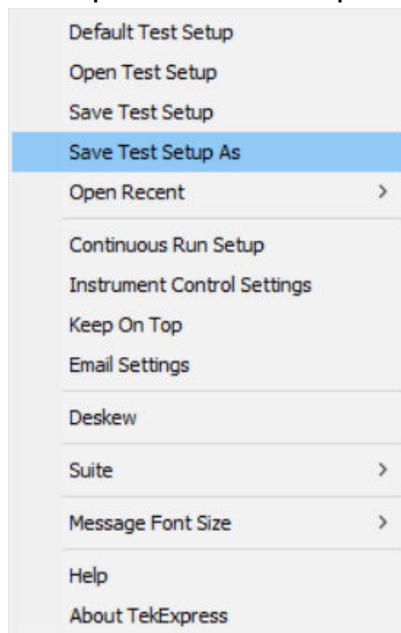
## Save the configured test setup

You can save a test setup before or after running a test. You can create a test setup from already created test setup or using a default test setup. When you save a setup, all the parameters, measurement limits, waveform files (if applicable), test selections, and other configuration settings are saved under the setup name. When you select the default test setup, the parameters are set to the application's default value.

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



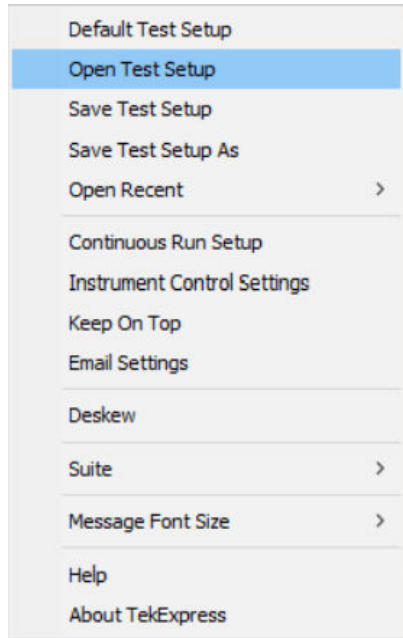
- Select **Options > Save Test Setup As** to save the setup with different name.



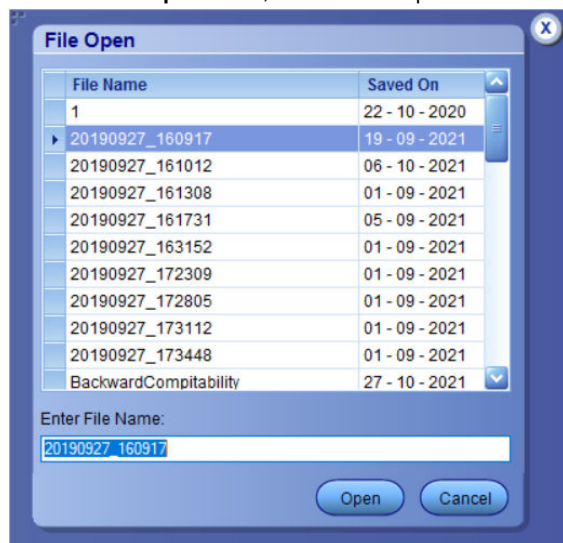
## Load a saved test setup

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

- Select **Options > Open Test Setup**.



- From the **File Open** menu, select the setup file name from the list and click **Open**.

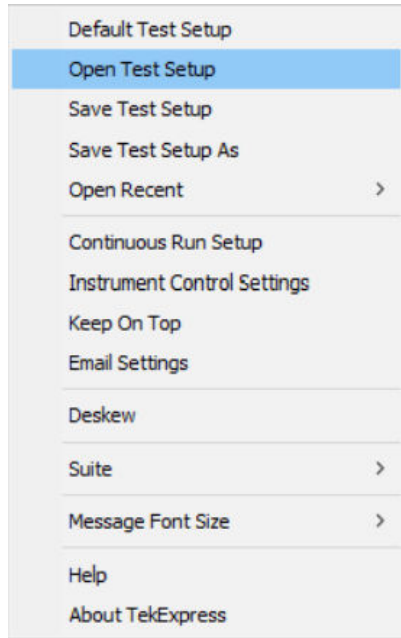


**Note:** Parameters that are set for the respective test setup will enable after opening the file.

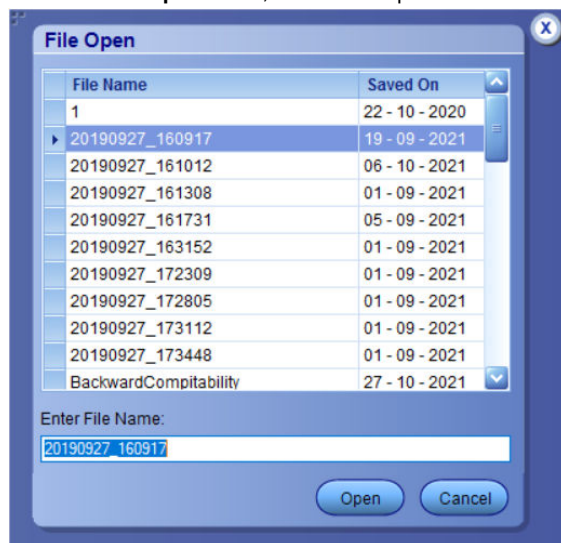
## Perform a test using pre-run session files

Complete the following steps to load a test setup from a pre-run session:

1. Select **Options > Open Test Setup**.

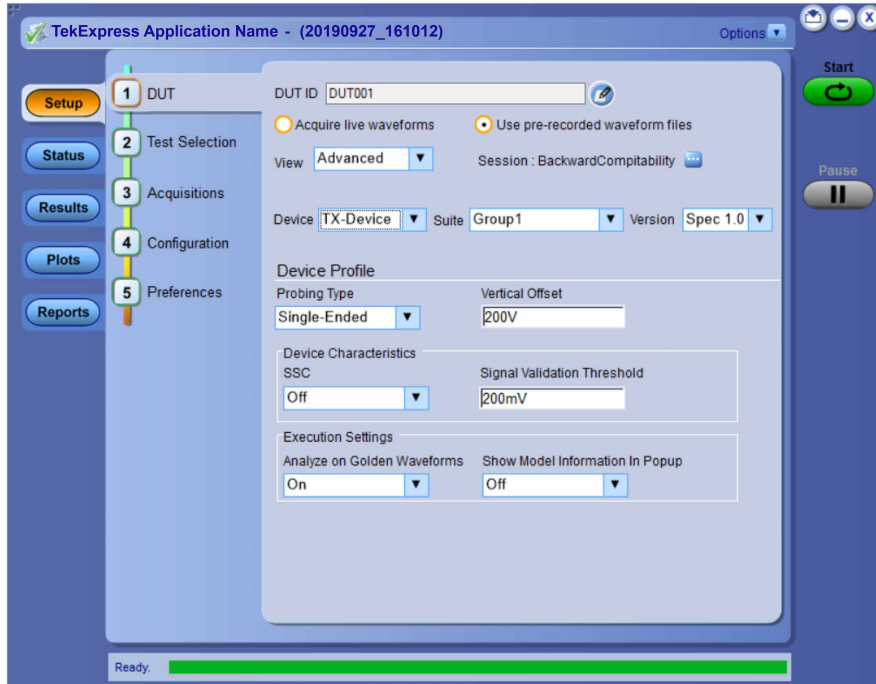


- From the **File Open** menu, select a setup from the list and then click **Open**.

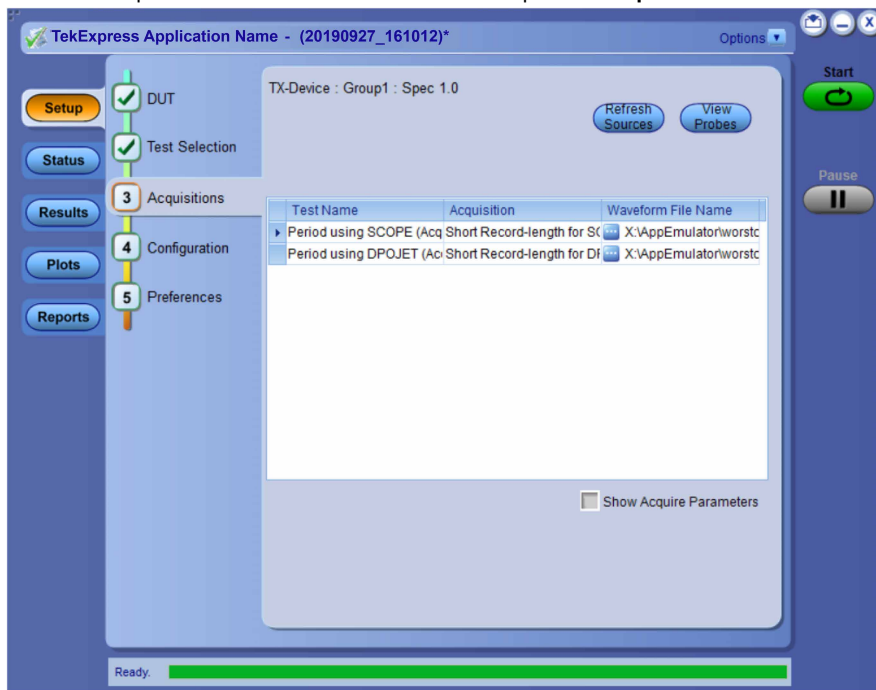



**Note:** Parameters that are set for the respective test setup will enable after opening the file.

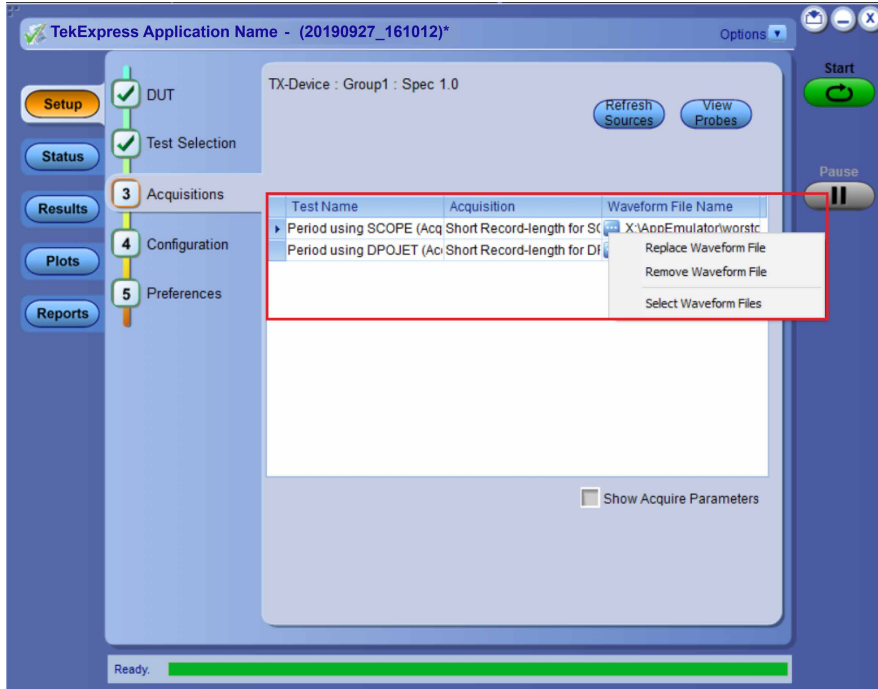
- Switch the mode to **Use Pre-recorded waveform files** in the DUT panel.



4. Select the required waveforms from the selected setup in the **Acquisitions** tab and click **Start**.



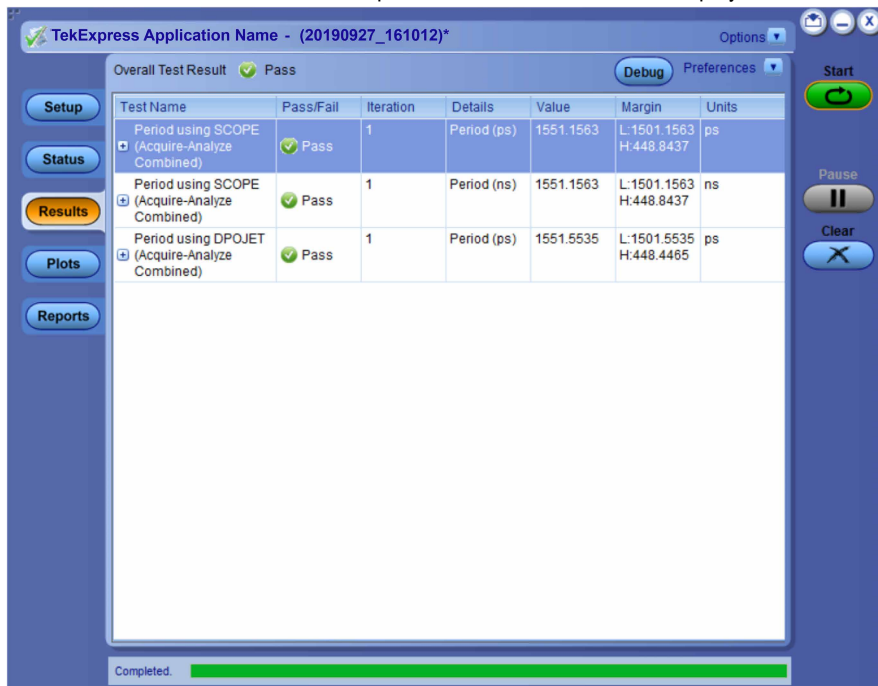
5. The selected waveform file can be removed/replaced by clicking on the (  ) icon.



6. After successful completion of the test, the waveform report files are stored at X:\<Application Name>\Reports.

| Name            | Date modified      | Type           |
|-----------------|--------------------|----------------|
| DUT001_2266.mht | 10/27/2021 4:25 AM | MHTML Document |
| DUT001_2265.mht | 10/27/2021 1:24 AM | MHTML Document |
| DUT001_2264.mht | 10/6/2021 2:58 AM  | MHTML Document |
| DUT001_2263.mht | 10/6/2021 2:40 AM  | MHTML Document |
| DUT001_2262.mht | 10/6/2021 2:35 AM  | MHTML Document |
| DUT001_2261.mht | 10/6/2021 2:23 AM  | MHTML Document |

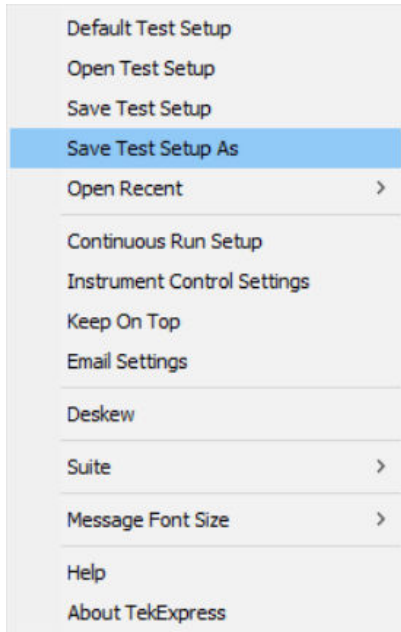
7. The overall test result status after completion of the test execution is displayed in the Results Panel.



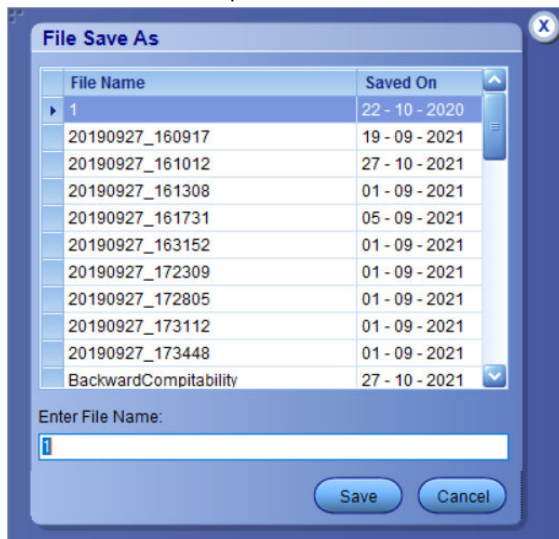
## Save the test setup with a different name

To save a test setup with a different name, follow the steps:

1. Select **Options > Save Test Setup As**.



2. Enter the new test setup name and click **Save**.





# SCPI Commands

## About SCPI command

You can use the Standard Commands for Programmable Instruments (SCPI) to communicate remotely with the TekExpress application. Complete the TCP/IP socket configuration and the TekVISA configuration in the oscilloscope or in the device where you are executing the script.



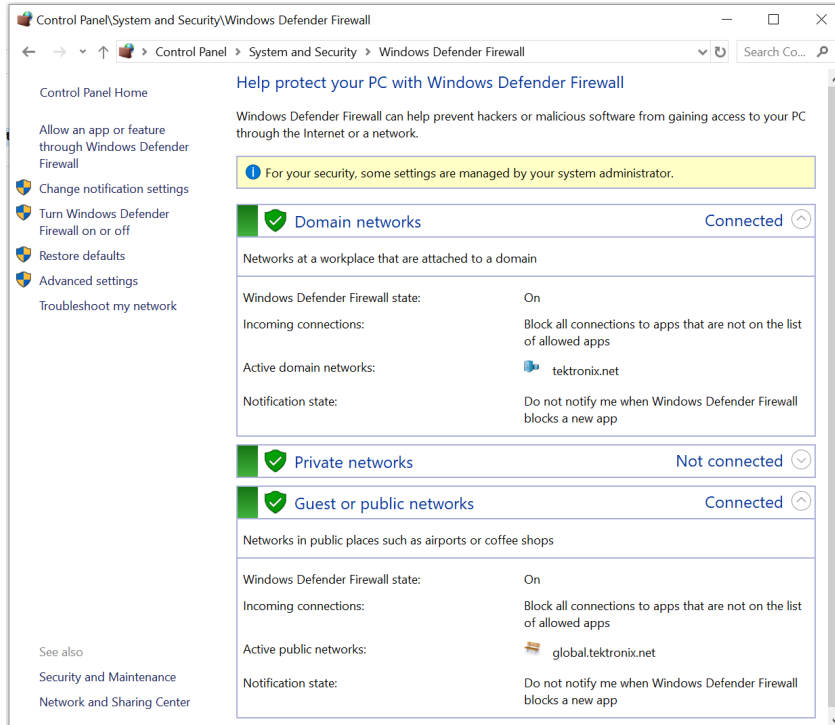
**Note:** If you are using an external PC to execute the remote interface commands, then install TekVISA in the PC to make the configurations.

## Socket configuration for SCPI commands

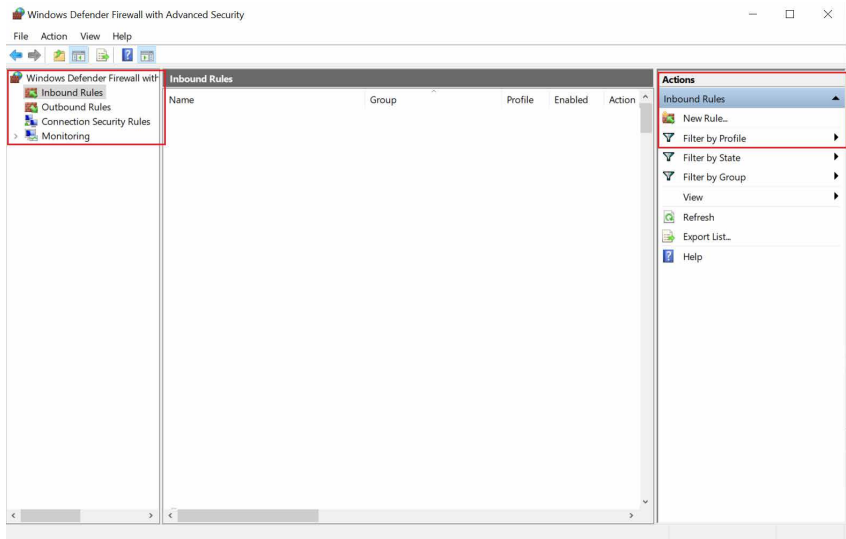
This section describes the steps to configure the TCP/IP socket configuration in your script execution device and the steps to configure the TekVISA configuration in the oscilloscope to execute the SCPI commands.

### TCP/IP socket configuration

1. Click **Start > Control Panel > System and Security > Windows Firewall > Advanced settings**.

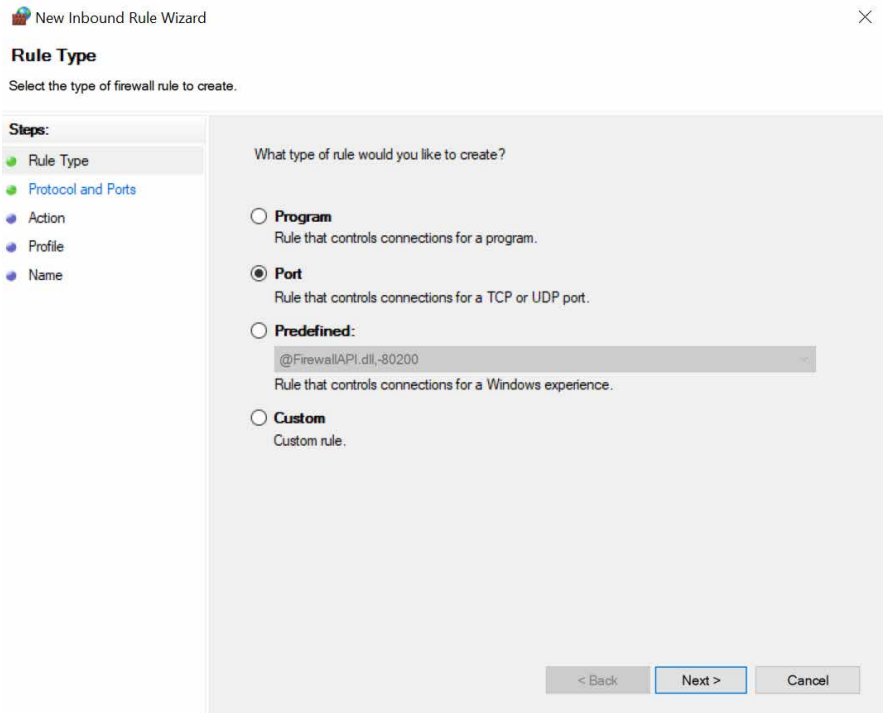


2. In Windows Firewall with Advanced Security menu, select **Windows Firewall with Advanced Security on Local Computer > Inbound Rules** and click **New Rule...**

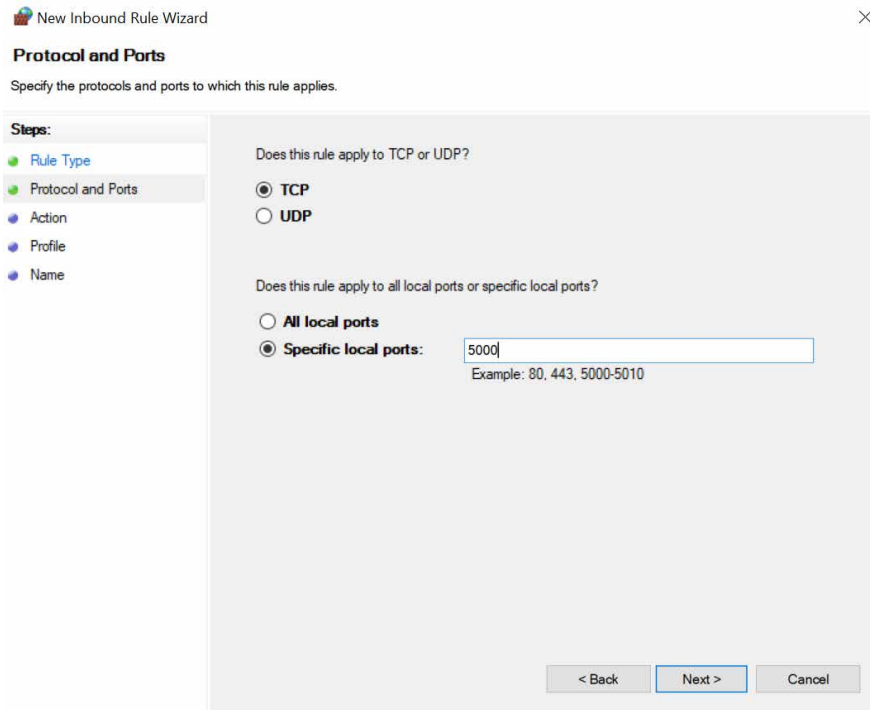


3. In **New Inbound Rule Wizard** menu

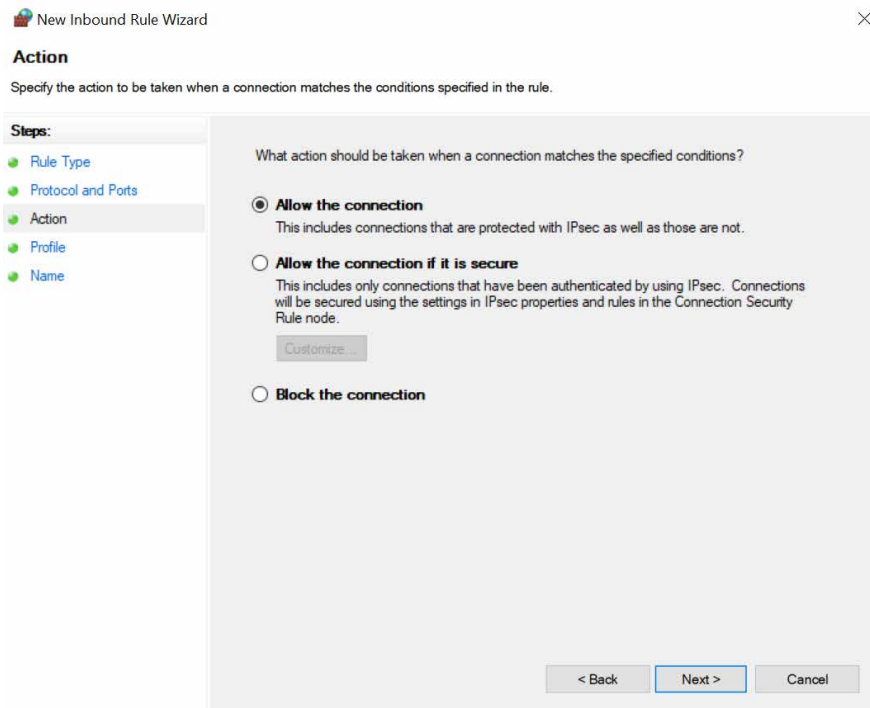
a. Select **Port** and click **Next**.



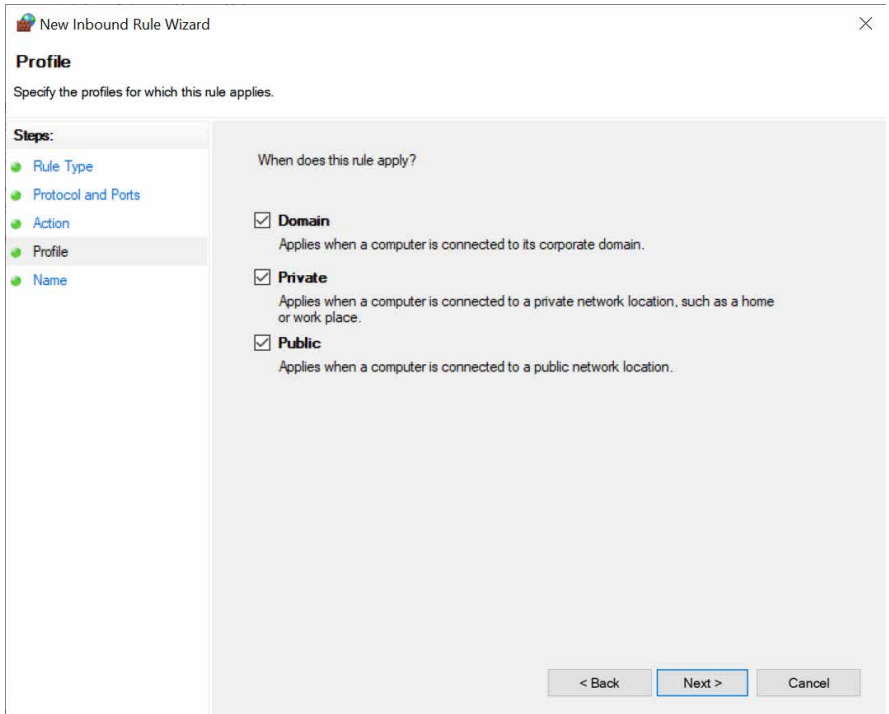
b. Select **TCP** as rule apply, enter 5000 for **Specific local ports** and click **Next**.



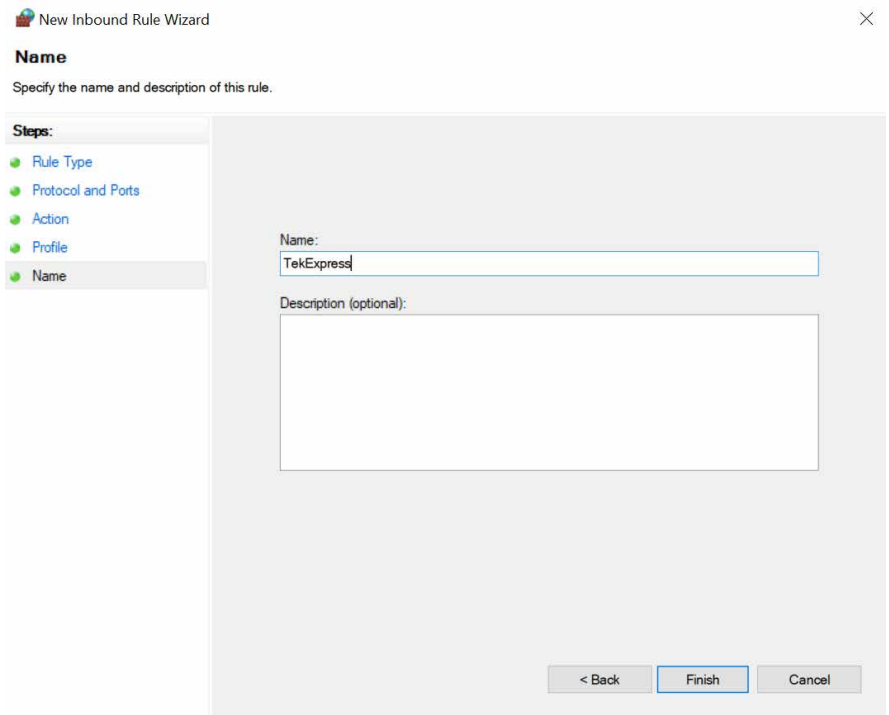
c. Select **Allow the connection** and click **Next**.



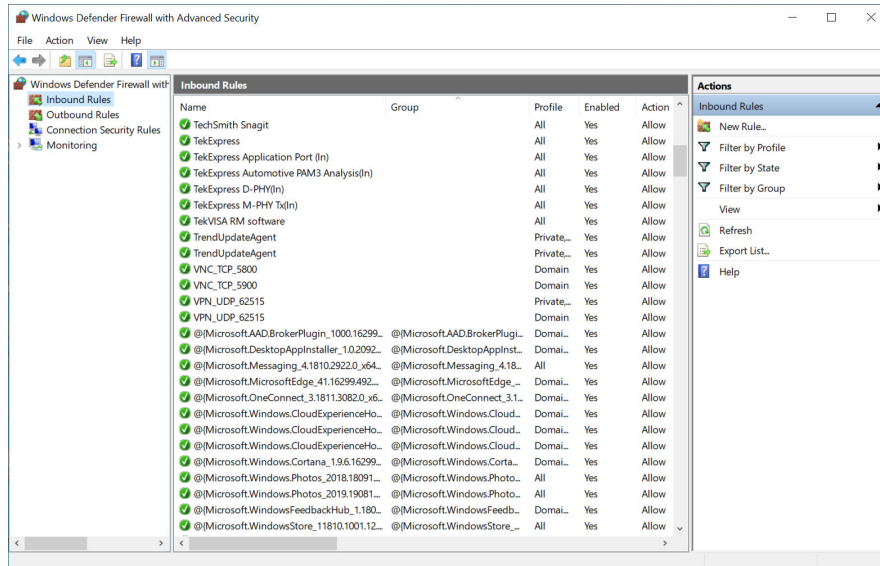
d. Select **Domain, Private, Public** checkbox and click **Next**.



e. Enter **Name**, Description (optional), and click **Finish**.

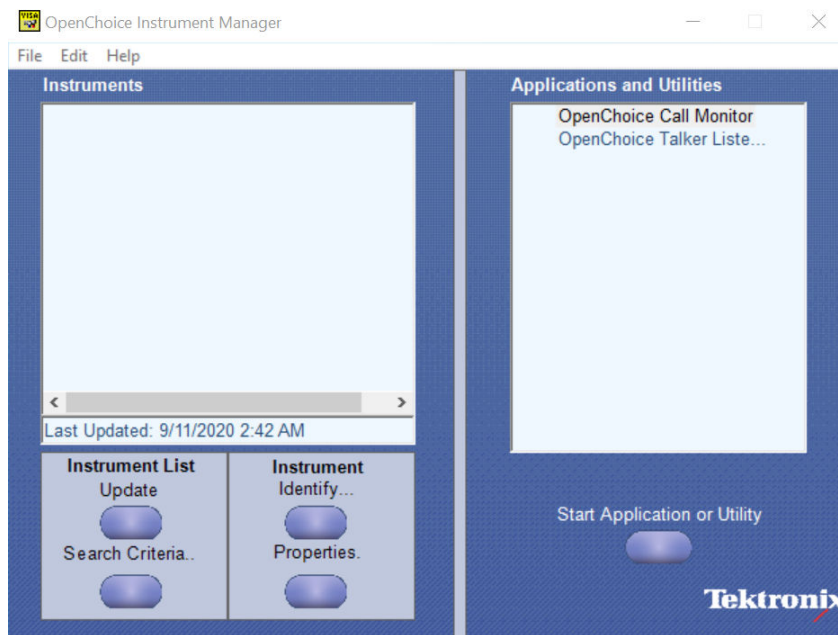


4. Check whether the Rule name is displayed in **Windows Firewall with Advanced Security** menu > **Inbound Rules**.



## TekVISA configuration

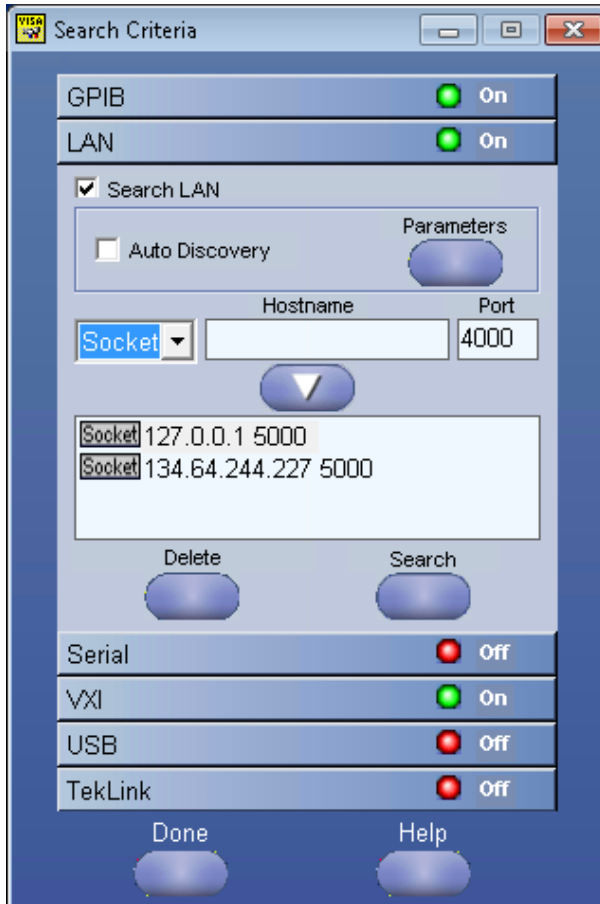
1. Click **Start > All Programs > TekVISA > OpenChoice Instrument Manager**.



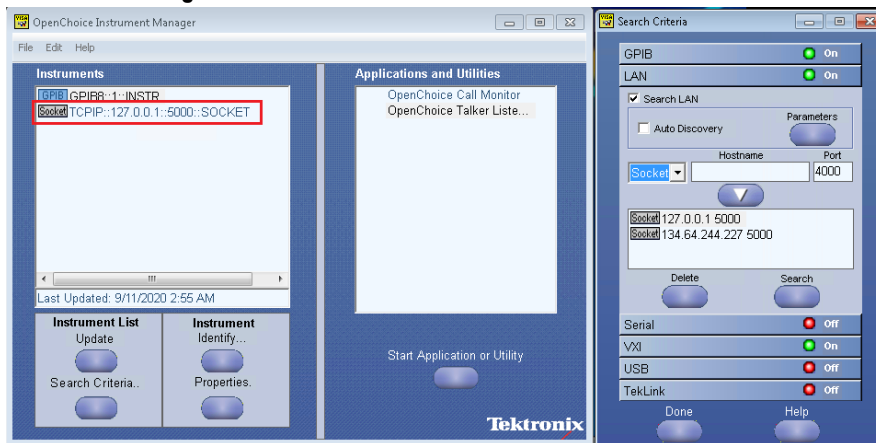
2. Click **Search Criteria**. In **Search Criteria** menu, click **LAN** to Turn-on. Select **Socket** from the drop-down list, enter the IP address of

the TekExpress device in **Hostname** and type **Port** as 5000. Click  to configure the IP address with Port.

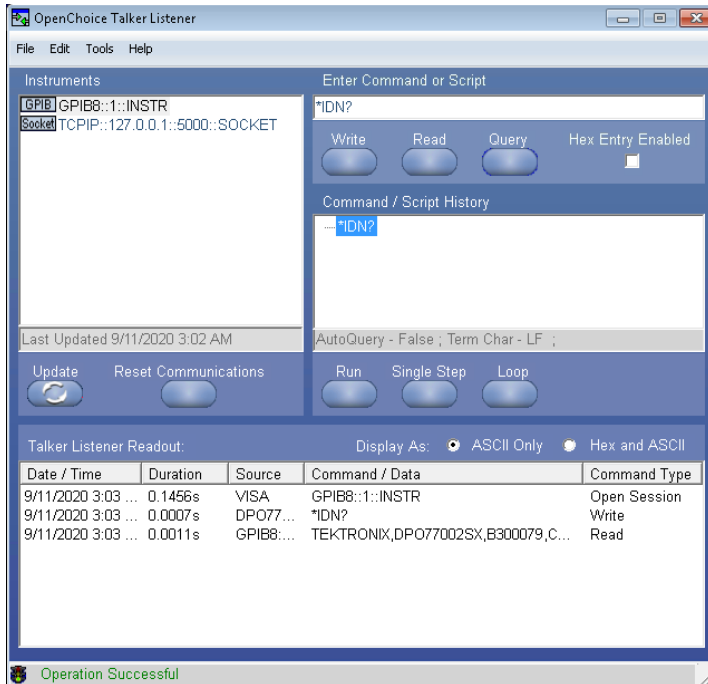
Enter the Hostname as 127.0.0.1 if the TekVISA and TekExpress application are in the same system, else enter the IP address of the oscilloscope where the TekExpress application is running.



3. Click **Search** to setup the TCPIP connection with the host. Check whether the TCPIP host name is displayed in **OpenChoice Instrument Manager > Instruments**.



4. Double-click **OpenChoice Talker Listener** and enter the Command **\*IDN?** in command entry field and click **Query**. Check that the Operation is successful and Talker Listener Readout displays the Command / Data.



## Set or query the device name of application

This command sets or queries the device name of the application.

### Syntax

TEKEXP:SELECT DEVICE, "<DeviceName>" (Set)

TEKEXP:SELECT? DEVICE (Query)

### Command arguments

| Argument Name  | Argument Type |
|--|---------------|
| <DeviceName> <ul style="list-style-type: none"> <li>• Drive</li> <li>• Host</li> </ul> | <String>      |

### Returns

<String>

### Examples

TEKEXP:SELECT DEVICE, "<DeviceName>" command sets the device name of the application.

TEKEXP:SELECT? DEVICE command returns the selected device name of the application.

## Set or query the suite name of the application

This command sets or queries the suite name of the application.

### Syntax

TEKEXP:SELECT SUITE, "<SuiteName>" (Set)

TEKEXP:SELECT? SUITE (Query)

### Command arguments

| SuiteName   |
|-------------|
| Transmitter |

### Returns

<String>

### Examples

TEKEXP:SELECT SUITE, "<SuiteName>" command sets the suite name of the application.

TEKEXP:SELECT? SUITE command returns the selected suite of the application.

## Set or query the test name of the application

This command selects or deselects the specified test name of the application.

### Syntax

TEKEXP:SELECT TEST, "<TestName>", <Value> (Set)

TEKEXP:SELECT TEST, "<ALL>" (Set)

TEKEXP:SELECT? TEST (Query)

### Command arguments

| TestName | Value |
|----------|-------|
|          |       |

### Returns

{True | False} or {1 | 0}

### Examples

TEKEXP:SELECT TEST, "<TestName>", 1 command selects the specified test in the Test Panel.

TEKEXP:SELECT TEST, "<ALL>" command select all the tests in the Test Panel.

TEKEXP:SELECT? TEST command returns the list of selected tests.

## Set or query the version name of the application

This command sets or queries the version name of the application.

### Syntax

TEKEXP:SELECT VERSION, "<VersionName>" (Set)

TEKEXP:SELECT? VERSION (Query)



## Command arguments

| VersionName |  |
|-------------|--|
|             |  |

## Returns

<String>

## Examples

TEKEXP:SELECT VERSION, "<VersionName>" command sets the version name of application.

TEKEXP:SELECT? VERSION command returns the version name of application.

## Set or query the general parameter values

This command sets or queries the general parameter values of the application.

## Syntax

TEKEXP:VALUE GENERAL, "<ParameterName>", "<Value>" (Set)

TEKEXP:VALUE? GENERAL, "<ParameterName>" (Query)

## Command arguments

Table 18: Command arguments for general settings

| ParameterName | Value   |
|---------------|---|
| Data+         | <b>Options:</b> CH1, CH2, CH3, CH4<br><b>Default:</b> CH1<br><b>Actual:</b> CH1 |
| Data-         | <b>Options:</b> CH1, CH2, CH3, CH4<br><b>Default:</b> CH3<br><b>Actual:</b> CH3 |

Table continued...

| ParameterName              |         | Value   |
|----------------------------|---------|---|
| Data Rates                 | 1.5Gbps | <b>Options:</b> Included, Excluded<br><b>Default:</b> Included<br><b>Actual:</b> Included   |
|                            | 3Gbps   | <b>Options:</b> Included, Excluded<br><b>Default:</b> Included<br><b>Actual:</b> Included   |
|                            | 6Gbps   | <b>Options:</b> Included, Excluded<br><b>Default:</b> Included<br><b>Actual:</b> Included   |
|                            | 12Gbps  | <b>Options:</b> Included, Excluded<br><b>Default:</b> Included<br><b>Actual:</b> Included   |
| Deskew Alert Enabled       |         | <b>Options:</b> True, False<br><b>Default:</b> True<br><b>Actual:</b> True  |
| DUT control                |         | <b>Options:</b> Manual, Custom<br><b>Default:</b> Manual<br><b>Actual:</b> Manual   |
| Enable DeEmbed Filter      |         | <b>Options:</b> Yes, No<br><b>Default:</b> No<br><b>Actual:</b> No  |
| DeEmbed Filter             |         | <b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\Drive\FilterFiles\SAS receptacle.flt<br><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\Drive\FilterFiles\SAS receptacle.flt |
| On Failure Stop and Notify |         | <b>Options:</b> True, False<br><b>Default:</b> False<br><b>Actual:</b> False  |

Table continued...

| ParameterName                              | Value   |
|--|---|
| Signal Validation                          | <p><b>Options:</b></p> <ul style="list-style-type: none"> <li>• Prompt me if signal fails</li> <li>• Use signals as is - do not check</li> <li>• Skip test if signal check fails</li> </ul> <p><b>Default:</b> Prompt me if signal fails<br/> <b>Actual:</b> Prompt me if signal fails</p>  |
| Number of retries for instrument IO errors | <p><b>Default:</b> 3<br/> <b>Actual:</b> 3</p>  |
| Time between retries (seconds)             | <p><b>Default:</b> 20<br/> <b>Actual:</b> 20</p>  |
| DUTID Comment                              | <p><b>Default:</b> SAS Transmitter<br/> <b>Actual:</b> SAS Transmitter</p>  |
| DSP Filter (8 Ghz)                         | <p><b>Options:</b></p> <ul style="list-style-type: none"> <li>• True</li> <li>• False</li> </ul> <p><b>Default:</b> True<br/> <b>Actual:</b> True</p>   |
| Show Channel Configuration Pop-Up          | <p><b>Options:</b></p> <ul style="list-style-type: none"> <li>• True</li> <li>• False</li> </ul> <p><b>Default:</b> True<br/> <b>Actual:</b> True</p>   |
| Device Control Scripts                     | <p><b>Options:</b> OOB, 12Gbps:D10.2, 6Gbps:D10.2, 3Gbps:D10.2, 1.5Gbps:D10.2, 12Gbps:D10.2:SSC_ON, 6Gbps:D10.2:SSC_ON, 3Gbps:D10.2:SSC_ON, 1.5Gbps:D10.2:SSC_ON, 12Gbps:CJTPAT, 6Gbps:CJTPAT, 3Gbps:CJTPAT, 1.5Gbps:CJTPAT, 12Gbps:D24.3, 6Gbps:D24.3, 3Gbps:D24.3, 1.5Gbps:D24.3, 12Gbps:D24.3:SSC_ON, 6Gbps:D24.3:SSC_ON, 3Gbps:D24.3:SSC_ON, 1.5Gbps:D24.3:SSC_ON, 12Gbps:D30.3, 6Gbps:D30.3, 3Gbps:D30.3, 1.5Gbps:D30.3, 12Gbps:SCRAMBLED0, 6Gbps:SCRAMBLED0, 3Gbps:SCRAMBLED0, 1.5Gbps:SCRAMBLED0, 12Gbps:K28.5, 6Gbps:K28.5, 3Gbps:K28.5, 1.5Gbps:K28.5</p> <p><b>Default:</b> OOB<br/> <b>Actual:</b> OOB</p> |

Table continued...

| ParameterName                                 | Value  |
|---|--|
| Script for Device control-OOB                 | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\OOB.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\OOB.bat</p>                             |
| Script for Device control-12Gbps:D10.2        | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\12G\D10.2_12G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\12G\D10.2_12G.bat</p>         |
| Script for Device control-6Gbps:D10.2         | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\6G\D10.2_6G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\6G\D10.2_6G.bat</p>             |
| Script for Device control-3Gbps:D10.2         | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\3G\D10.2_3G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\3G\D10.2_3G.bat</p>             |
| Script for Device control-1.5Gbps:D10.2       | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\1.5G\D10.2_1.5G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\1.5G\D10.2_1.5G.bat</p>     |
| Script for Device control-12Gbps:D10.2:SSC_ON | C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\12G\D10.2_SSCOn_12G.bat  |
| Script for Device control-6Gbps:D10.2:SSC_ON  | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\6G\D10.2_SSCOn_6G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\6G\D10.2_SSCOn_6G.bat</p> |

Table continued...

| ParameterName                                  | Value  |
|--|--|
| Script for Device control-3Gbps:D10.2:SSC_ON   | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\3G\D10.2_SSCOn_3G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\3G\D10.2_SSCOn_3G.bat</p>         |
| Script for Device control-1.5Gbps:D10.2:SSC_ON | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\1.5G\D10.2_SSCOn_1.5G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\1.5G\D10.2_SSCOn_1.5G.bat</p> |
| Script for Device control-12Gbps:CJTPAT        | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\12G\CJTPAT_12G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\12G\CJTPAT_12G.bat</p>               |
| Script for Device control-6Gbps:CJTPAT         | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\6G\CJTPAT_6G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\6G\CJTPAT_6G.bat</p>                   |
| Script for Device control-3Gbps:CJTPAT         | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\3G\CJTPAT_3G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\3G\CJTPAT_3G.bat</p>                   |
| Script for Device control-1.5Gbps:CJTPAT       | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\1.5G\CJTPAT_1.5G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\1.5G\CJTPAT_1.5G.bat</p>           |

Table continued...

| ParameterName                                 | Value  |
|---|--|
| Script for Device control-12Gbps:D24.3        | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\12G\D24.3_12G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\12G\D24.3_12G.bat</p>             |
| Script for Device control-6Gbps:D24.3         | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\6G\D24.3_6G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\6G\D24.3_6G.bat</p>                 |
| Script for Device control-3Gbps:D24.3         | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\3G\D24.3_3G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\3G\D24.3_3G.bat</p>                 |
| Script for Device control-1.5Gbps:D24.3       | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\1.5G\D24.3_1.5G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\1.5G\D24.3_1.5G.bat</p>         |
| Script for Device control-12Gbps:D24.3:SSC_ON | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\12G\D24.3_SSCOn_12G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\12G\D24.3_SSCOn_12G.bat</p> |
| Script for Device control-6Gbps:D24.3:SSC_ON  | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\6G\D24.3_SSCOn_6G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\6G\D24.3_SSCOn_6G.bat</p>     |

Table continued...

| ParameterName                                  | Value  |
|--|--|
| Script for Device control-3Gbps:D24.3:SSC_ON   | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\3G\D24.3_SSCOn_3G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\3G\D24.3_SSCOn_3G.bat</p>         |
| Script for Device control-1.5Gbps:D24.3:SSC_ON | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\1.5G\D24.3_SSCOn_1.5G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\1.5G\D24.3_SSCOn_1.5G.bat</p> |
| Script for Device control-12Gbps:D30.3         | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\12G\D30.3_12G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\12G\D30.3_12G.bat</p>                 |
| Script for Device control-12Gbps:D30.3         | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\12G\D30.3_12G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\12G\D30.3_12G.bat</p>                 |
| Script for Device control-6Gbps:D30.3          | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\6G\D30.3_6G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\6G\D30.3_6G.bat</p>                     |
| Script for Device control-3Gbps:D30.3          | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\3G\D30.3_3G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\3G\D30.3_3G.bat</p>                     |

Table continued...

| ParameterName                                | Value  |
|--|--|
| Script for Device control-1.5Gbps:D30.3      | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\1.5G\D30.3_1.5G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\1.5G\D30.3_1.5G.bat</p>           |
| Script for Device control-12Gbps:SCRAMBLED0  | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\12G\SCRAMBLED0_12G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\12G\SCRAMBLED0_12G.bat</p>     |
| Script for Device control-6Gbps:SCRAMBLED0   | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\6G\SCRAMBLED0_6G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\6G\SCRAMBLED0_6G.bat</p>         |
| Script for Device control-3Gbps:SCRAMBLED0   | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\3G\SCRAMBLED0_3G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\3G\SCRAMBLED0_3G.bat</p>         |
| Script for Device control-1.5Gbps:SCRAMBLED0 | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\1.5G\SCRAMBLED0_1.5G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\1.5G\SCRAMBLED0_1.5G.bat</p> |
| Script for Device control-12Gbps:K28.5       | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\12G\K28.5_12G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\12G\K28.5_12G.bat</p>               |

Table continued...



| ParameterName                             | Value  |
|---|--|
| Script for Device control-6Gbps:K28.5     | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\6G\K28.5_6G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\6G\K28.5_6G.bat</p>         |
| Script for Device control-3Gbps:K28.5     | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\3G\K28.5_3G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\3G\K28.5_3G.bat</p>         |
| Script for Device control-1.5Gbps:K28.5   | <p><b>Default:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\1.5G\K28.5_1.5G.bat</p> <p><b>Actual:</b> C:\Program Files\Tektronix\TekExpress\TekExpress SAS\Compliance Suites\AutomatedBatchFile\1.5G\K28.5_1.5G.bat</p> |
| Show SSC Acquisition for TJ               | <p><b>Options:</b></p> <ul style="list-style-type: none"> <li>• True</li> <li>• False</li> </ul> <p><b>Default:</b> False</p> <p><b>Actual:</b> False</p>  |
| Timer Warning Info Message Popup          | <p><b>Options:</b></p> <ul style="list-style-type: none"> <li>• True</li> <li>• False</li> </ul> <p><b>Default:</b> False</p> <p><b>Actual:</b> False</p>  |
| Timer Warning Info Message Popup Duration | <p><b>Default:</b> 10</p> <p><b>Actual:</b> 10</p>   |
| Timer Error Message Popup                 | <p><b>Options:</b></p> <ul style="list-style-type: none"> <li>• True</li> <li>• False</li> </ul> <p><b>Default:</b> False</p> <p><b>Actual:</b> False</p>  |

Table continued...

| ParameterName                      | Value                                   |
|------------------------------------|---|
| Timer Error Message Popup Duration | <b>Default:</b> 10<br><b>Actual:</b> 10 |

**Table 19: Command arguments for report settings**

| ParameterName                              | Value   |
|--|---|
| Auto increment report name if duplicate    | <b>Options:</b> True, False<br><b>Default:</b> True<br><b>Actual:</b> True                |
| Include Pass/Fail Results Summary          | <b>Options:</b> True, False<br><b>Default:</b> True<br><b>Actual:</b> True                |
| Include Detailed Results                   | <b>Options:</b> True, False<br><b>Default:</b> True<br><b>Actual:</b> True                |
| Include Plot Images                        | <b>Options:</b> True, False<br><b>Default:</b> True<br><b>Actual:</b> True                |
| Create report at the end                   | <b>Options:</b> Included, Excluded<br><b>Default:</b> Included<br><b>Actual:</b> Included |
| Include Setup Configuration                | <b>Options:</b> True, False<br><b>Default:</b> True<br><b>Actual:</b> True                |
| Include Complete Application Configuration | <b>Options:</b> True, False<br><b>Default:</b> False<br><b>Actual:</b> False              |
| Include User Comments                      | <b>Options:</b> True, False<br><b>Default:</b> True<br><b>Actual:</b> True                |

Table continued...

| ParameterName                   | Value   |
|---------------------------------|---|
| Save As Type                    | <b>Options:</b> Web Archive (*.mht;*.mhtml)<br><b>Default:</b> Web Archive (*.mht;*.mhtml)<br><b>Actual:</b> Web Archive (*.mht;*.mhtml)  |
| View Report After Generating    | <b>Options:</b> True, False<br><b>Default:</b> True<br><b>Actual:</b> True  |
| Report Update Mode              | <b>Options:</b> New, Append, Replace<br><b>Default:</b> New<br><b>Actual:</b> New   |
| Report Group Mode               | <b>Options:</b> Test Name, Lane Name, Test result, Equilization<br><b>Default:</b> Test Name<br><b>Actual:</b> Test Name  |
| Report Generation Template Path | <b>Default:</b> INSTALL_FOLDER\Report<br>Generator\TekExpress_Template_GroupByTestName.xml<br><b>Actual:</b> INSTALL_FOLDER\Report<br>Generator\TekExpress_Template_GroupByTestName.xml |
| Report Settings:Column Template | <b>Default:</b> INSTALL_FOLDER\Report<br>Generator\TekExpress_ReportColumnTemplate.xml<br><b>Actual:</b> INSTALL_FOLDER\Report<br>Generator\TekExpress_ReportColumnTemplate.xml         |
| Report Path                     | <b>Default:</b> WORKING_FOLDER\Reports\DUT001.mht<br><b>Actual:</b> WORKING_FOLDER\Reports\DUT001.mht   |

## Returns

<NRf> or <String>

## Examples

TEKEXP:VALUE GENERAL, "<ParameterName>", "<Value>" command set the value for the specified general parameter.

TEKEXP:VALUE? GENERAL, "<ParameterName>" command returns the value for the specified general parameter.

## Set or query the acquire parameter values

This command sets or queries the acquire parameter values of the application.

### Syntax

TEKEXP:VALUE

ACQUIRE, "<TestName>", "<AcquireType>", "<ParameterName>", "<ParameterValue>" (Set)

TEKEXP:VALUE? ACQUIRE, "<TestName>", "<AcquireType>", "<ParameterName>" (Query)

### Command arguments

| TestName | AcquireType | ParameterName | ParameterValue |
|----------|-------------|---------------|----------------|
|          |             |               |                |

### Returns

<Nrf>

### Examples

TEKEXP:VALUE

ACQUIRE, "<TestName>", "<AcquireType>", "<ParameterName>", "<ParameterValue>" command sets the value for the specified test and its acquire parameter.

TEKEXP:VALUE? ACQUIRE, "<TestName>", "<AcquireType>", "<ParameterName>" command returns the value for the specified test and its acquire parameter.

## Set or query the analyze parameter values

This command sets or queries the analyze parameter values of the application.

### Syntax

TEKEXP:VALUE ANALYZE, "<TestName>", "<ParameterName>", "<ParameterValue>" (Set)

TEKEXP:VALUE? ANALYZE, "<TestName>", "<ParameterName>" (Query)

### Command arguments

| TestName | ParameterName | ParameterValue |
|----------|---------------|----------------|
|          |               |                |

### Returns

<Nrf>

### Examples

TEKEXP:VALUE ANALYZE, "<TestName>", "<ParameterName>", "<ParameterValue>" command set the value for the specified test and its analyze parameter.

TEKEXP:VALUE? ANALYZE, "<TestName>", "<ParameterName>" command returns the value for the specified test and its analyze parameter.

## Query the available devices in the DUT panel of the application

This command queries the list of available devices on the DUT panel as comma separated values.

## Syntax

```
TEKEXP:LIST? DEVICE (Query)
```

## Command arguments

## Returns

<String>

## Examples

TEKEXP:LIST? DEVICE command returns the list of available devices.

## Query the available suites for the selected device

This command queries the list of available suites for the selected device as comma separated values.

## Syntax

```
TEKEXP:LIST? SUITE (Query)
```

## Returns

<String>

## Examples

TEKEXP:LIST? SUITE command returns the list of available suites for the selected device.

## Query the list of available tests of the application

This command queries the list of available tests of the application for the selected device as comma separated values.

## Syntax

```
TEKEXP:LIST? TEST (Query)
```

## Command arguments

## Returns

<String>

## Examples

TEKEXP:LIST? TEST command returns the list of available tests for the selected device.

## Query the available version names of the application

This command queries the list of available version names of the application for the selected device as comma separated values.

## Syntax

```
TEKEXP:LIST? VERSION (Query)
```

## Returns

<String>

### Examples

TEKEXP:LIST? VERSION command returns the list of version names for the selected device.

## Query the list of available instruments based on the specified instrument type

This command queries the list of available instruments based on the specified instrument type.

### Syntax

TEKEXP:LIST? INSTRUMENT, "<InstrumentType>" (Query)

### Command argument

| Argument Name    | Argument value |
|------------------|----------------|
| <InstrumentType> | <String>       |

### Returns

<String>

### Examples

TEKEXP:LIST? INSTRUMENT, "Real Time Scope" command returns the list of available instruments based on the real time scope type.

## Set or query the IP address of the instrument based on the specified instrument type

This command sets or queries the IP address of the instrument based on the specified instrument type.

### Syntax

TEKEXP:INSTRUMENT? "<InstrumentType>" (Query)

TEKEXP:INSTRUMENT, "<InstrumentType>", "<Value>" (Set)

### Command argument

| Argument Name    | Argument Type                            |
|------------------|--|
| <InstrumentType> | <String>                                 |
| <Value>          | <String><br>TCPIP::XXX.XX.XXX.XXX::INSTR |

### Returns

<String>

### Examples

TEKEXP:INSTRUMENT? "<InstrumentType>" command returns the IP address of the oscilloscope.

TEKEXP:INSTRUMENT, "<InstrumentType>", "<value>" command sets the oscilloscope to the specified IP address.

## Query the information of the generated report file

This command queries the information of the generated report file in the format "<FileSize>",<FileName>".

### Pre-requisite

A session should be run earlier and the report should be generated to get the information of the report.

### Syntax

```
TEKEXP:INFO? REPORT (Query)
```

### Returns

<FileSize>:: <String>

<FileName>:: <String>

### Examples

TEKEXP:INFO? REPORT command returns the information of the generated report in the format ("1215","DUT001.mht").

## Query the information of the generated waveform files

This command queries the information of the generated waveform files in the format.

<File1Size,"File1Name">.

If there are more than one waveform, the waveform file names are displayed with the comma separated values in the format

<File1Size,"File1Name">,<File2Size,"File2Name">.

### Syntax

```
TEKEXP:INFO? WFM (Query)
```

### Returns

<FileSize>:: <String>

<FileName>:: <String>

### Examples

TEKEXP:INFO? WFM command returns the information of the generated waveform in the format (20000858,"X:\SAS Transmitter\Untitled Session\DUT001\20200916\_041609\Iter1\_Short Record-length for SCOPE Period\_NoSSC\_DIFF.wfm").

## Query the information of the generated image files

This command queries the information of the generated image files in the format.

<File1Size,"File1Name">.

If there are more than one image, the image file names are displayed with the comma separated values in the format

<File1Size,"File1Name">,<File2Size,"File2Name">.

### Syntax

```
TEKEXP:INFO? IMAGE (Query)
```

### Returns

<FileSize>:: <String>

<FileName>:: <String>

### Examples

TEKEXP:INFO? IMAGE command returns the information of the generated image in the format (109058,"X:\SAS Transmitter\Untitled Session\DUT001\20200916\_041609\Iter1\_Short Record-length for SCOPE Period\_NoSSC\_DIFF.png";22794,"X:\SAS Transmitter\UntitledSession\DUT001\20 200916\_041609\ScopePeriodPlot\_Iteration1WithCursor.png").

## Query the active TekExpress application name

This command queries the active TekExpress application name running on the oscilloscope.

### Syntax

TEKEXP:\*IDN? (Query)

### Returns

<String>

### Examples

TEKEXP:\*IDN? command returns the active TekExpress application name running on the oscilloscope.

## Set or query the DUTID of application

This command sets or queries the DUTID of the application.

### Syntax

TEKEXP:VALUE DUTID,"<Value>" (Set)

TEKEXP:VALUE? DUTID (Query)

### Command arguments

| Argument Name | Argument Type |
|---------------|---------------|
| <Value>       | <String>      |

### Returns

<String>

### Examples

TEKEXP:VALUE DUTID,"DUT001" command sets the DUTID of the application to DUT001.

TEKEXP:VALUE? DUTID command returns the DUTID of the application.

## Sets or query the acquire mode status

This command sets or queries the acquire mode status.

### Syntax

TEKEXP:ACQUIRE\_MODE <Mode> (Set)

TEKEXP:ACQUIRE\_MODE? (Query)



## Command arguments

| Argument Name | Argument value   |
|---------------|--|
| <Mode>        | <ul style="list-style-type: none"> <li>LIVE</li> <li>PRE-RECORDED</li> </ul> |

## Returns

LIVE | PRE-RECORDED

## Examples

TEKEXP:ACQUIRE\_MODE LIVE command sets the acquire mode to the Live mode.

TEKEXP:ACQUIRE\_MODE? command returns the current acquire mode.

## Set or query the execution mode status

This command sets or queries the execution mode status.

## Syntax

TEKEXP:MODE <Mode> (Set)

TEKEXP:MODE? (Query)

## Command arguments

| Argument Name | Argument value   |
|---------------|--|
| <Mode>        | <ul style="list-style-type: none"> <li>COMPLIANCE</li> <li>USER-DEFINED</li> </ul> |

## Returns

COMPLIANCE | USER-DEFINED

## Examples

TEKEXP:MODE COMPLIANCE command sets the execution mode to the compliance mode.

TEKEXP:MODE? command returns the current execution mode.

## Generate the report for the current session

This command generates the report for the current session.

## Syntax

TEKEXP:REPORT GENERATE(Set)

## Arguments

N/A

## Examples

TEKEXP:REPORT GENERATE command generates the report for the current session.

## Query the value of specified report header field in the report

This command queries the value of specified report header field in the report.

### Syntax

TEKEXP:REPORT? "<Device Field>" (Query)

### Command arguments

| Argument Name   | Argument Type |
|---|---------------|
| <Device Field><br><br>Device field is the header name of each field in the setup information section of the report. | <String>      |

| Setup Information  |                       |                             |                  |
|--|-----------------------|-----------------------------|------------------|
| DUT ID   | DUT001                | Scope Model                 | DPO77002SX       |
| Date/Time  | 11/23/2022 1:12:31 PM | Scope Serial Number         | 8300009          |
| Acquisition Mode   | Live                  | Scope F/W Version           | 10.14.0 Build 15 |
| DUT Control  | Automated             | SCPI Factory SW Calibration | PASS/PASS        |
| DUT Type   | Device                | TekExpress Thunderbolt TX   | 10.1.3.3         |
| DUT Part Number  | 1                     | TekExpress Framework        | 5.8.6.71         |
| Test Method  | SigTest               | SCPIEX version              | 0.75             |
| Total Acquisition Time                                     | 00:28:07.78           | CTS Version                 | v1.03            |
| Total Analysis Time  | 00:13:24.19           |                             |                  |
| Over All Test Result                                       | Pass                  |                             |                  |
| DUT COMMENT: General Comment - Thunderbolt Transmitter DUT |                       |                             |                  |

| Minimum Line Interval |           |        |                |             |                       |           |            |
|-----------------------|-----------|--------|----------------|-------------|-----------------------|-----------|------------|
| Measurement Details   | Data Rate | Line   | Measured Value | Test Result | Margin                | Low Limit | High Limit |
| MinimumInterval       | 10C       | Lane 0 | 100.014 ps     | Pass        | LL: 44.815 fs, HL: NA | 99.97 ps  | NA         |
| LowMax                | 10C       | Lane 0 | 100.017 ps     | Pass        | LL: NA, HL: 12.86 fs  | NA        | 100.03 ps  |
| MinimumInterval       | 10C       | Lane 1 | 100.015 ps     | Pass        | LL: 44.899 fs, HL: NA | 99.97 ps  | NA         |
| LowMax                | 10C       | Lane 1 | 100.017 ps     | Pass        | LL: NA, HL: 12.66 fs  | NA        | 100.03 ps  |

COMMENTS: For Lane0:  
For Lane1:

### Returns

<String>

### Examples

TEKEXP:REPORT? "DUT ID" command returns the value of DUT ID field in the report.

## Query the value of specified result detail available in report summary/details table

This command queries the value of specified result detail available in report summary/details table.

### Syntax

TEKEXP:RESULT? "<TestName>" (Query)  
 TEKEXP:RESULT? "<TestName>", "<ColumnName>" (Query)  
 TEKEXP:RESULT? "<TestName>", "<ColumnName>", <RowNumber> (Query)

### Command arguments

| Argument Name  | Argument Type |
|--|---------------|
| <TestName><br><br>It is the test name of which the details are required in the report. | <String>      |

Table continued...

| Argument Name   | Argument Type |
|---|---------------|
| <ColumnName><br>It is the column header name of which the details are required in the report. | <String>      |
| <RowNumber><br>It is the row number of which the details are required in the report.          | <String>      |

## Returns

<String>

## Examples

TEKEXP:RESULT? "<TestName>" will return the pass fail status of test.

TEKEXP:RESULT? "<TestName>", "<ColumnName>" will return all the row values of specific column for the test with comma separated values.

TEKEXP:RESULT? "<TestName>", "<ColumnName>", <RowNumber> will return the column value of specified row number.

## Restore the setup to default settings

This command restores the setup to default settings.

### Syntax

```
TEKEXP:SETUP Default(Set)
```

### Arguments

N/A

### Examples

TEKEXP:SETUP Default command restores the setup to default settings.

## Save the setup

This command saves the setup.

### Syntax

```
TEKEXP:SETUP Save(Set)
```

### Examples

TEKEXP:SETUP Save command saves the setup.

## Save the settings to a specified session

This command saves the settings to a specified session.

### Syntax

```
TEKEXP:SETUP Save, "<SessionName>"
```

### Command arguments

| Argument Name | Argument value |
|---------------|----------------|
| <SessionName> | <String>       |

### Examples

TEKEXP:SETUP Save, "<SessionName>" command saves the settings to a specified session.

## Open the setup from a specified session

This command opens the setup from a specified session.

### Syntax

TEKEXP:SETUP Open, "<SessionName>"(Set)

### Command arguments

| Argument Name | Argument value |
|---------------|----------------|
| <SessionName> | <String>       |

### Examples

TEKEXP:SETUP Open, "<SessionName>" command opens the setup from a specified session.

## Query the current setup file name

This command queries the current setup file name.

### Syntax

TEKEXP:SETUP? CURRENT (Query)

### Returns

<String>

### Examples

TEKEXP:SETUP? CURRENT command returns the current setup file name.

## Run/stop/pause/resume the selected measurements execution in the application

This command run/stop/pause/resume the selected measurements execution in the application.

### Syntax

TEKEXP:STATE <operation mode> (Set)

## Command arguments

| Argument Name    | Argument value   |
|------------------|--|
| <operation mode> | <ul style="list-style-type: none"> <li>• RUN</li> <li>• STOP</li> <li>• PAUSE</li> <li>• RESUME</li> </ul> |

## Returns

RUN | STOP | PAUSE | RESUME

## Examples

TEKEXP : STATE RUN command runs the execution for the selected measurements.

## Query the current measurement execution status

This command queries the current measurement execution status.

### Syntax

TEKEXP : STATE? (Query)

## Returns

RUNNING | PAUSED | WAIT | ERROR | READY

## Examples

TEKEXP : STATE? command returns the current measurement execution status.

## Query whether the current setup is saved or not saved

This command queries whether the current setup is saved or not saved.

### Syntax

TEKEXP : STATE? SETUP (Query)

## Returns

Saved or Not-Saved

## Examples

TEKEXP : STATE? SETUP command returns whether the current setup is saved or not saved.

## Exit or close the application

The command exits or close the application

### Syntax

TEKEXP : EXIT(Set)

## Examples

TEKEXP : EXIT command close the application.

## Query the status of the previous command execution

This command queries whether the previous command execution is completed successfully.

### Syntax

TEKEXP : \*OPC? (Query)

### Returns

{0 | 1} or {True | False}

1 or True indicates that command execution is successful.

0 or False indicates that command execution is failed.

### Examples

TEKEXP : \*OPC? command returns whether the previous command operation is completed successfully.

## Query the last error occurred

This command queries the last error occurred.

### Syntax

TEKEXP : LASTERROR? (Query)

### Returns

<String>

### Examples

TEKEXP : LASTERROR? command returns the last error occurred.

## Set or query the popup details

This command sets or queries the popup details.

### Syntax

TEKEXP : POPUP? (Query)

TEKEXP : POPUP "<PopupResponse>" (Set)

### Command arguments

| Argument Name   | Argument value  |
|-----------------|---|
| <PopupResponse> | <ul style="list-style-type: none"> <li>• Yes</li> <li>• No</li> </ul> |

### Returns

The pop-up details return in the following format:

"<Title>",<message>",<response1>,<response2>".

Where,

<Title> :: <String>

<message> :: <String>

<response1>,<response2> :: <String>

## Examples

TEKEXP:POPUP? command returns the popup details in following format ": "Do you really want to exit TekExpress?";Responses: "Yes, No".

TEKEXP:POPUP "Yes" command sets the popup response to Yes.

## Sets or query the limit values in the limits editor window

This command sets or queries the limit values in the limits editor window.

### Syntax

TEKEXP:VALUE LIMIT,<TestName>,<LimitHeader>,<Value1>,<CompareString>,<Value2>(Set)

TEKEXP:VALUE? LIMIT,<TestName>,<LimitHeader> (Query)

### Returns

<String> or <NRf>

### Examples

TEKEXP:VALUE LIMIT,<TestName>,<LimitHeader>,<Value1>,<CompareString>,<Value2> command sets the limits value for the specified testname and limit header.

TEKEXP:VALUE? LIMIT,<TestName>,<LimitHeader> command returns the limits value for the specified testname and limit header.

## Set or query the waveform file recalled for the specified test name and acquire type

This command set or queries the waveform file recalled for the specified test name and acquire type.

If there are more than one waveform, the waveform file names are displayed with the symbol "\$" separated values in the format

<WaveformFileName1\$ WaveformFileName2>.

### Syntax

TEKEXP:VALUE WFMFILE,<TestName>,<AcquireType>,<WaveformFileName> (Set)

TEKEXP:VALUE? WFMFILE,<TestName>,<AcquireType> (Query)

### Returns

<String>

### Examples

TEKEXP:VALUE WFMFILE,<TestName>,<AcquireType>,<WaveformFileName> command recalls the specified waveform file for the specified testname and acquire type.

TEKEXP:VALUE? WFMFILE, <TestName>, <AcquireType> command returns the waveform file name recalled for the specified testname and acquire type.

## Set or query the enable/disable status of Verbose function

This command sets or queries the enable/disable status of Verbose function.

### Syntax

TEKEXP:VALUE VERBOSE, "<Value>" (Set)

TEKEXP:VALUE? VERBOSE (Query)

### Arguments

| Argument Name | Argument value   |
|---------------|--|
| <Value>       | {True   False} or {1   0}<br>It represents enabled or disabled.<br>Where, <ul style="list-style-type: none"> <li>• True or 1 - enabled</li> <li>• False or 0 - disabled</li> </ul> |

### Returns

{True | False} or {0 | 1}

### Examples

TEKEXP:VALUE VERBOSE, "<Value>" command enable or disable the Verbose function.

TEKEXP:VALUE? VERBOSE command returns the enable or disable status of Verbose function.

## Query the enable or disable status of Continuous run function.

This command queries the enable or disable status of Continuous run function.

### Syntax

TEKEXP:VALUE? GENERAL, "Enable Continuous Run" (Query)

### Returns

{True | False} or {0 | 1}

Where,

1 or True indicates that the continuous run function is enabled.

0 or False indicates that the continuous run function is disabled.

### Examples

TEKEXP:VALUE? GENERAL, "Enable Continuous Run" command returns the enable or disable status of continuous run function.



## Set or query the enable/disable status of Continuous Run function

This command sets or queries the enable/disable status of Continuous Run function.

### Syntax

TEKEXP:VALUE ContinuousRun, "<Value>" (Set)

TEKEXP:VALUE? ContinuousRun (Query)

### Arguments

| Argument Name | Argument value  |
|---------------|---|
| <Value>       | {True   False} or {1   0}<br>It represents enabled or disabled.<br>Where,<br><ul style="list-style-type: none"> <li>• True or 1 - enabled</li> <li>• False or 0 - disabled</li> </ul> |

### Returns

{True | False} or {0 | 1}

### Examples

TEKEXP:VALUE? ContinuousRun command returns the enable or disable status of Continuous run function.

TEKEXP:VALUE ContinuousRun, "<Value>" command enable or disable the Continuous run function.

## Set or query the continuous run duration time value

This command sets or queries the continuous run duration time value.

### Syntax

TEKEXP:VALUE? ContinuousRun\_Duration (Query)

TEKEXP:VALUE ContinuousRun\_Duration, "<Value>" (Set)

### Arguments

| Argument Name | Argument value   |
|---------------|--|
| <Value>       | Infinite   hh:mm<br>Infinite sets the radio on button to infinite.<br>hh:mm sets the continuous run duration to the specified time in hours and minutes. The minimum time duration you can set is 00:30. |

### Returns

Infinite | hh:mm

### Examples

TEKEXP:VALUE? ContinuousRun\_Duration command returns the continuous run duration time value.

TEKEXP:VALUE ContinuousRun\_Duration, "<Value>" command sets the continuous run duration time value.

## Set or query the session create option in the continuous run function

This command sets or queries the option for session creation in the continuous run function.

### Syntax

TEKEXP:VALUE? ContinuousRun\_RunSessionOptions (Query)

TEKEXP:VALUE ContinuousRun\_RunSessionOptions, "Value" (Set)

### Arguments

| Argument Name | Argument value  |
|---------------|---|
| <Value>       | NewSession   SameSession_ClearResults<br>NewSession - creates new session for each run.<br>SameSession_ClearResults - Clears the test results of the current session and starts the test execution. The session results will be added in the same session, by erasing the previous run results. |

### Returns

NewSession | SameSession\_ClearResults

### Examples

TEKEXP:VALUE? ContinuousRun\_RunSessionOptions command returns the option for session creation in the continuous run function.

TEKEXP:VALUE ContinuousRun\_RunSessionOptions, "Value" command sets the option for session creation in the continuous run function.

## Set or query the View report after generating option status

This command sets or queries the enable/disable status of the View report after generating function.

### Syntax

TEKEXP:VALUE? GENERAL, "View Report After Generating" (Query)

TEKEXP:VALUE GENERAL, "View Report After Generating", <value> (Set)

## Arguments

| Argument Name | Argument value   |
|---------------|--|
| <Value>       | {True   False} or {1   0}<br>It represents enabled or disabled.<br>Where, <ul style="list-style-type: none"> <li>• True or 1 - enabled</li> <li>• False or 0 - disabled</li> </ul> |

## Returns

{True | False} or {0 | 1}

## Examples

TEKEXP:VALUE? GENERAL, "View Report After Generating" command returns the enable or disable status of view report after generating option.

TEKEXP:VALUE GENERAL, "View Report After Generating", <value> command enable or disable the view report after generating option.

## Returns the report as XML string

This command returns the report as XML string.

## Syntax

TEKEXP:REPORTASXML? (Query)

## Returns

<String>

## Examples

TEKEXP:REPORTASXML? command returns the report XML string.

## Copies all the images from current run session to the given destination location

This command copies all the images from current run session to the given destination location.

## Syntax

TEKEXP:COPYIMAGES <DestinationPath> (Set)

## Command argument

<DestinationPath> :: <String>

## Returns

NA

## Examples

TEKEXP:COPYIMAGES C:\Temp command copies all the images from current run session to the mentioned location.

## Selects the specified test(s) and deselect all other tests

This command selects the specified test(s) and deselect all other tests.

### Syntax

TEKEXP:SELECTID <"TestID"> (Set)

### Command argument

| Argument Name | Argument value |
|---------------|----------------|
| TestID        | String         |

### Returns

NA

### Examples

TEKEXP:SELECTID "11101" This command select the test associated with the ID and deselects all other tests in the application.

TEKEXP:SELECTID "11101, 11102" This command selects the tests associated with the IDs and other tests will be deselected.

## Returns the complete information about the selected test

This command returns the complete information about the selected test.

The information includes application name, TestID, Device selected, Suite selected, version, Test name, Test description.

### Syntax

TEKEXP:TESTINFO? (Query)

### Returns

<String>

### Examples

TEKEXP:TESTINFO? This command returns the following details:

<TekExpress> <Test Id="11101" Device="TX-Device" Suite="Group1" Version="Spec 1.0" Name="Algorithm Library Measurement" Description="This is Algorithm Library measurement test. Refer Section-B of TekExpress SampleApp Development Guide for more details.

## Set the default session

Sets the application configurations to default value.

### Syntax

TEKEXP:SESSION DEFAULT (set)

### Examples

TEKEXP:SESSION DEFAULT, sets the application configurations to default value.

## Save the run/config sessions

Enter the name to save/config the session.

## Syntax

```
TEKEXP:SESSION SAVE, "Session Name" (set)
```

## Command arguments

| Argument Name  | Argument value |
|----------------|----------------|
| <Session Name> | <String>       |

## Examples

```
TEKEXP:SESSION SAVE, "Session Name" saves the session.
```

## Load the run/config session

Load the selected config/run session.

## Syntax

```
TEKEXP:SESSION LOAD, "Session Name" (set)
```

## Command arguments

| Argument Name  | Argument value |
|----------------|----------------|
| <Session Name> | <String>       |

## Examples

```
TEKEXP:SESSION LOAD, "Session Name", load the selected config/run session.
```

## Delete the run/config session

Deletes the selected config/run session.

## Syntax

```
TEKEXP:SESSION DELETE, "Session1, Session2" (set)
```

## Command arguments

| Argument Name  | Argument value |
|----------------|----------------|
| <Session Name> | <String>       |

## Examples

```
TEKEXP:SESSION DELETE, "Session1, Session2", deletes the selected config/run session.
```

## Run the run/config saved session

Run the selected config/run session.

## Syntax

```
TEKEXP:SESSION RUN, "Session Name's separated by comma" (set)
```

### Command arguments

| Argument Name  | Argument value |
|--|----------------|
| <Session Name>   | <String>       |
| Session Name's separated by comma (to run the multiple run sessions) | <String>       |

### Examples

TEKEXP:SESSION RUN, "Session Name's separated by comma", runs the selected config/run session.

## Query the available list in the run/config session

Returns the list of available config/run session.

### Syntax

TEKEXP:SESSION? LIST

### Returns

Returns the list of available config/run session.

### Examples

TEKEXP:SESSION? LIST, returns the list of available config/run session.

## Query the current run/config session

Returns the selected config/run session.

### Syntax

TEKEXP:SESSION? CURRENT

### Returns

Returns the selected config/run session.

### Examples

TEKEXP:SESSION? CURRENT, returns the selected config/run session.

## Override the run/config session

Overrides the selected config/run session.

### Syntax

TEKEXP:SESSION SAVE, "SessionName", "True" (set)

## Command arguments

| Argument Name  | Argument Type | Argument Value  |
|----------------|---------------|---|
| <Session Name> | <String>      | {True   False} or {1   0}<br>It represents enabled or disabled.<br>Where, <ul style="list-style-type: none"><li>• True or 1 - enabled</li><li>• False or 0 - disabled</li></ul> |

## Returns

{True | False} or {0 | 1}

## Examples

TEKEXP:SESSION SAVE, "SessionName", "True", overrides the selected config/run session.

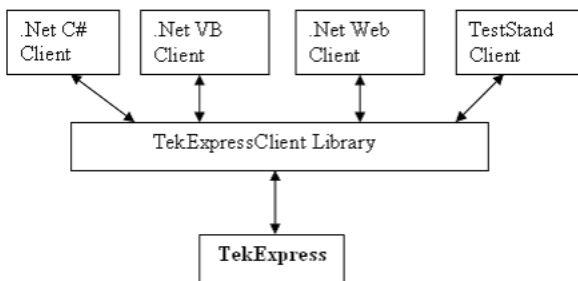
## About the programmatic interface

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

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

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

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



### See also

[Requirements for Developing TekExpress Client](#)

[Remote Proxy Object](#)

[Client Proxy Object](#)

## Requirements for developing TekExpress client

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

### References required

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



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

### Required steps for a client

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

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



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

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

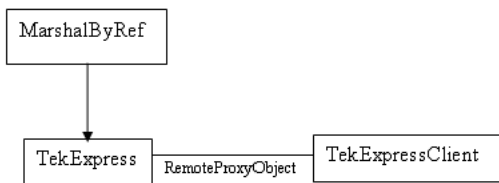
3. When the client has connected to and locked the server, the client can access any of the programmatic controls needed to run the remote automations.
4. After the client operations finish, the client unlocks the server.

#### See also

[SAS application commands flow](#)

## Remote proxy object

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



The following is an example: `RemotingConfiguration.RegisterWellKnownServiceType (typeof (TekExpressRemoteInterface), "TekExpress Remote interface", WellKnownObjectMode.Singleton);`

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

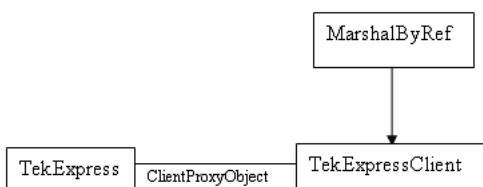
For example,

`//Get a reference to the remote object`

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

## Client proxy object

Client exposes a proxy object to receive certain information.



For example,

`//Register the client proxy object`

```
WellKnownServiceTypeEntry[] e = RemotingConfiguration.GetRegisteredWellKnownServiceTypes(); clientInterface = new ClientInterface();  
RemotingConfiguration.RegisterWellKnownServiceType(typeof(ClientInterface), "Remote Client Interface", WellKnownObjectMode.Singleton);
```

//Expose the client proxy object through marshalling

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

The client proxy object is used for the following:

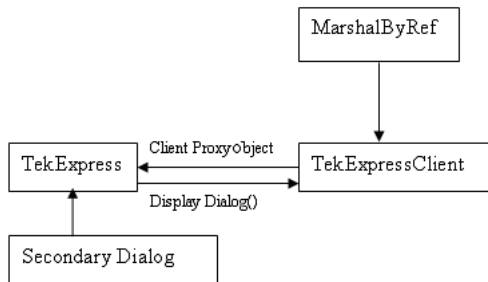
- To get the secondary dialog messages from the server.
- To get the file transfer commands from the server while transferring the report.

Examples

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

For more information, click the following links:

[Secondary Dialog Message Handling](#)



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

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

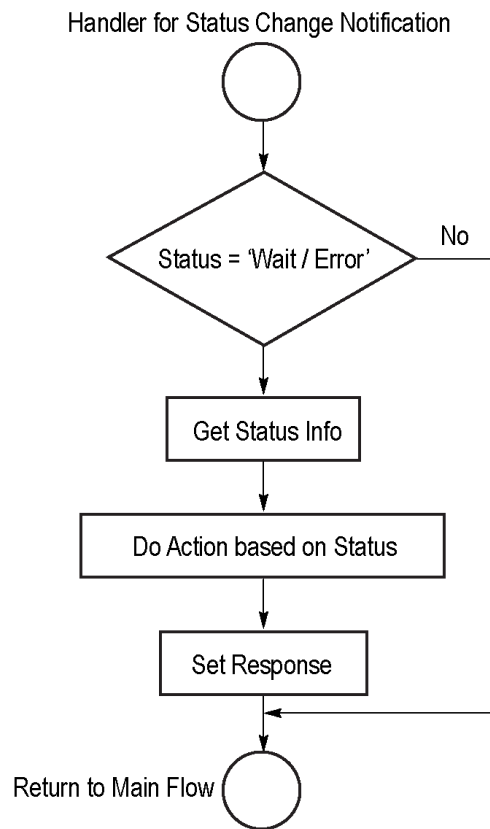
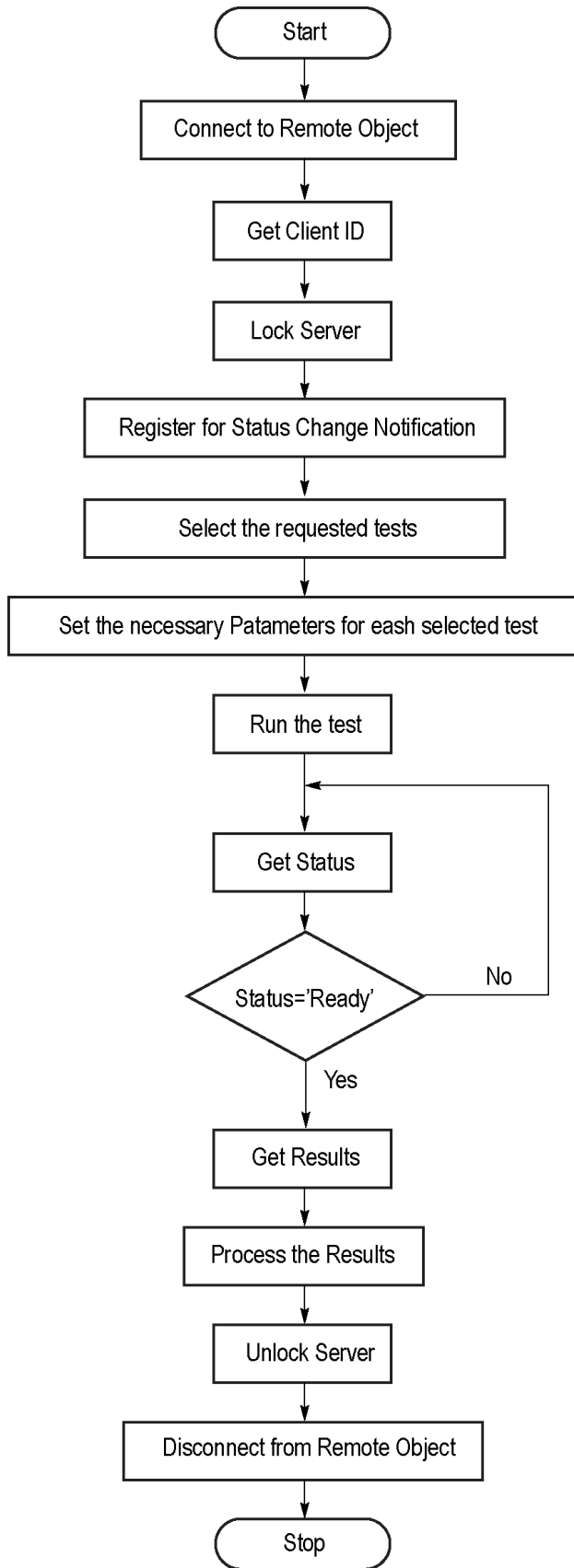
[File Transfer Events](#)

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

## Client programmatic interface overview

The following is an overview of the client programmatic interface:

Process flowchart



0643-001

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



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

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



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

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



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

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



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

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

### Handler of Status Change Notification

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

**See also**

[SAS application commands flow](#)

[Program remote access code example](#)

## Program remote access code example

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

**Table 20: Remote access code example**

| Task                           | Code  |
|--------------------------------|---|
| Start the application          |   |
| Connect through an IP address. | <code>m_Client.Connect("localhost") 'True or False clientID = m_Client.getClientID</code> |
| Lock the server                | <code>m_Client.LockServer(clientID)</code>  |
| Disable the Popups             | <code>m_Client.SetVerboseMode(clientID, false)</code>                                     |
| Table continued...             |   |

| Task                              | Code  |
|-----------------------------------|---|
| Set the DUT ID                    | <code>m_Client.SetDutId(clientID, "DUT_Name")</code>  |
| Select a test                     | For Transmitter:<br><code>mClient.SelectSingleTest(clientID, "Transmitter", "Drive", "5.1.2 -OOB Burst Amplitude", true)</code>                                 |
| Run with set configurations       | <code>m_Client.Run(clientID)</code>   |
| Wait for the test to complete.    | <code>Do Thread.Sleep(500)</code><br><code>m_Client.Application_Status(clientID)</code><br><code>Select Case status Case "Wait"</code>                          |
| Get the current state information | <code>mClient.GetCurrentStateInfo(clientID, WaitingMsBxCaption, WaitingMsBxMessage, WaitingMsBxButtonTexts)</code>  |
| Send the response                 | <code>mClient.SendResponse(clientID, WaitingMsBxCaption, WaitingMsBxMessage, WaitingMsBxResponse) End</code><br><code>Select Loop Until status = "Ready"</code> |
| Save results                      | 'Save all results values from folder for current run<br><code>m_Client.TransferResult(clientID, logDirname)</code>  |
| Unlock the server                 | <code>m_Client.UnlockServer(clientID)</code>  |
| Disconnect from serve             | <code>m_Client.Disconnect()</code>  |
| Exit the application              |   |

## SAS application commands flow

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

[Connect through an IP address](#)

[Lock the server](#)

[Disable the popups](#)

[Set or get the DUT ID](#)

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

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

[Select a test](#)

[Select a suite](#)

[Select a channel](#)

[Configure the selected measurement](#)

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

*Handle error codes*

*Get or set the timeout value*

*Wait for the test to complete*

*After the test is complete*

*Save, recall, or query a saved session*

*Unlock the server*

*Disconnect from the server*

| Name   | Type   | Direction | Description   |
|--|--------|-----------|---|
| <b>string id</b>   |        |           |   |
| id   | string | IN        | Identifier of the client performing the remote function   |
| <p><b>Ready:</b> Test configured and ready to start</p> <p><b>Running:</b> Test running</p> <p><b>Paused:</b> Test paused</p> <p><b>Wait:</b> A popup that needs your inputs</p> <p><b>Error:</b> An error is occurred</p> |        |           |   |
| <b>string dutName</b>  |        |           |   |
| dutName  | string | IN        | The new DUT ID of the setup   |
| <b>out bool saved</b>  |        |           |   |
| saved  | bool   | OUT       | Boolean representing whether the current session is saved   |
| This parameter is used as a check in SaveSession() and SaveSessionAs() functions.  |        |           |   |
| <b>string ipAddress</b>  |        |           |   |
| ipAddress  | string | IN        | The ip address of the server to which the client is trying to connect. This is required to establish the connection between the server and the client.            |
| <b>out string clientID</b>   |        |           |   |
| clientid   | string | OUT       | <p>Identifier of the client that is connected to the server.</p> <p>clientId = unique number + ipaddress of the client</p> <p>For example, 1065-192.157.98.70</p> |

Table continued...






| Name  | Type         | Direction | Description   |
|---|--------------|-----------|---|
|  <b>Note:</b> If the dutName parameter is null, the client is prompted to provide a valid DUT ID.  |              |           |   |
|  <b>Note:</b> 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.   |              |           |   |
|  <b>Note:</b> When the client is disconnected, it is unlocked from the server and then disconnected. The id is reused.   |              |           |   |
| <b>string dutId</b>   |              |           |   |
| dutId   | string       | OUT       | The DUT ID of the setup   |
| The dutId parameter is set after the server processes the request.  |              |           |   |
| <b>string device</b>  |              |           |   |
| device  | string       | IN        | Specifies the name of the device  |
| <b>string suite</b>   |              |           |   |
| suite   | string       | IN        | Specifies the name of the suite   |
| <b>string test</b>  |              |           |   |
| test  | string       | IN        | Specifies the name of the test to obtain the pass or fail status                          |
| <b>string parameterString</b>   |              |           |   |
| parameterString   | string       | IN        | Select or deselect a test   |
| <b>int rowNr</b>  |              |           |   |
| rowNr   | int          | IN        | Specifies the zero based row index of the sub-measurement for obtaining the result value. |
|  <b>Note:</b> When the client tries to lock a server that is locked by another client, the client gets a notification that the server is already locked and it must wait until the server is unlocked. If the client locks the server and is idle for a certain amount of time then the server is unlocked automatically from that client. |              |           |   |
| <b>out string[] status</b>  |              |           |   |
| status  | string array | OUT       | The list of statusmessages generated during the run                                       |
| <b>string name</b>  |              |           |   |
| name  | string       | IN        | The name of the session being recalled  |
| The name parameter cannot be empty. If it is empty, the client is prompted to provide a valid name.   |              |           |   |
|  <b>Note:</b> When the run is performed, the status of the run is updated periodically using a timer.  |              |           |   |

Table continued...




| Name  | Type   | Direction | Description   |
|---|--------|-----------|---|
| <b>string name</b>  |        |           |   |
| name  | string | IN        | The name of the session being saved                           |
| <p>The name parameter cannot be empty. If it is empty, the client is prompted to provide a valid name.</p> <p>Once the session is saved under 'name' you cannot use this method to save the session in a different name. Use SaveSessionAs instead.</p>   |        |           |   |
| <b>string name</b>  |        |           |   |
| name  | string | IN        | The name of the session being recalled                        |
| <p>The same session is saved under different names using this method. The name parameter cannot be empty. If it is empty, the client is prompted to provide a valid name.</p>   |        |           |   |
| <b>bool isSelected</b>  |        |           |   |
| isSelected  | bool   | IN        | Select or deselect a test                                     |
| <b>string time</b>  |        |           |   |
| time  | string | IN        | The time in seconds that refers to the timeout period         |
| <p>The time parameter gives the timeout period, which is the time the client is allowed to be locked and idle. After the timeout period if the client is still idle, it gets unlocked.</p> <p>The time parameter should be a positive integer; otherwise, the client is prompted to provide a valid timeout period.</p> |        |           |   |
| <b>bool_verbose</b>   |        |           |   |
| _verbose  | bool   | IN        | Specifies whether the verbose mode should be turned ON or OFF |
| <p> <b>Note:</b> When the session is stopped, the client is prompted to stop the session and is stopped at the consent.</p>  |        |           |   |
| <b>string filePath</b>  |        |           |   |
| filePath  | string | IN        | The location where the report must be saved in the client     |
| <p> <b>Note:</b> If the client does not provide the location to save the report, the report is saved at C:\ProgramFiles.</p>   |        |           |   |
| <p> <b>Note:</b> When the client is disconnected, the client is unlocked automatically.</p>  |        |           |   |
| <b>out string WaitingMsBxCaption</b>  |        |           |   |
| caption   | string | OUT       | The wait state or error state message sent to you             |

Table continued...



| Name  | Type         | Direction | Description  |
|---|--------------|-----------|--|
| <b>out string WaitingMsbBxMessage</b>       |              |           |  |
| message                                     | string       | OUT       | The wait state/error state message sent to you   |
| <b>out string[] WaitingMsbBxButtontexts</b> |              |           |  |
| buttonTexts                                 | string array | OUT       | An array of strings containing the possible response types that you can send   |
| <b>string WaitingMsbBxResponse</b>          |              |           |  |
| response                                    | string       | IN        | A string containing the response type that you can select (it must be one of the strings in the string array buttonTexts)                              |
| <b>out string clientID</b>                  |              |           |  |
| clientID                                    | string       | OUT       | Identifier of the client that is connected to the server<br><br>clientID = unique number + ipaddress of the client.<br>For example, 1065-192.157.98.70 |

## Connect through an IP address

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

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



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

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

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

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

## Lock the server

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

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

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



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

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

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

## Disable the popups

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

Table 21:

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

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

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



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

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

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

### Set or get the DUT ID

| Command name | Parameters                                      | Description  | Return value  | Example   |
|--------------|---|--|---|---|
| SetDutId()   | <i>string clientID</i><br><i>string dutName</i> | <p>This method changes the DUT ID of the setup.</p> <p>The client must provide a valid DUT ID.</p> | <p>String that gives the status of the operation after it has been performed</p> <p>Return value is "DUT Id Changed" on success</p> | <pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string return=m_Client.SetDutId(clientID, desiredDutId)</pre> <p><i>Note</i></p> |

Table continued...

| Command name | Parameters                                      | Description                                       | Return value  | Example  |
|--------------|---|---|---|--|
| GetDutId()   | <i>string clientID</i><br><i>string dutName</i> | This method gets the DUT ID of the current setup. | String that gives the status of the operation after it has been performed | <pre>m_Client = new Client()  //m_Client is a reference to the Client class in the Client DLL.  returnval as string  return = m_Client.GetDutId(clientID, out DutId)</pre> |

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

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



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

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

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

### Set the configuration parameters for a suite or measurement


| Command name          | Parameters   | Description   | Return value   | Example   |
|-----------------------|--|---|--|---|
| SetGeneralParameter() | <i>string clientID</i><br><i>string device</i><br><i>string suite</i><br><i>string test</i><br><i>string parameterString</i> | <p>This method sets the number of video lanes for the selected measurement.</p> <p> <b>Note:</b> Use this command to select a lane, channel, or source type.</p> | <p>String that gives the status of the operation after it has been performed</p> <p>The return value is "" (an empty String) on success.</p> | <pre>m_Client = new Client()  //m_Client is a reference to the Client class in the Client DLL.  returnval as string  <a href="#">Select channel example</a></pre> |
| SetAnalyzeParameter() | <i>string clientID</i><br><i>string device</i><br><i>string suite</i><br><i>string test</i><br><i>string parameterString</i> | This method sets the configuration parameters in the Analyze panel of the Configuration Panel dialog box for a suite or measurement.  | <p>String that gives the status of the operation after it has been performed</p> <p>The return value is "" (an empty String) on success.</p> | <pre>m_Client = new Client()  //m_Client is a reference to the Client class in the Client DLL.  returnval as string</pre>   |

Table continued...

| Command name          | Parameters   | Description  | Return value   | Example   |
|-----------------------|--|--|--|---|
| SetAcquireParameter() | <i>string clientID</i><br><i>string device</i><br><i>string suite</i><br><i>string test</i><br><i>string parameterString</i> | This method sets the configuration parameters in the Acquire panel of the Configuration Panel dialog box for a suite or measurement. | String that gives the status of the operation after it has been performed.<br><br>The return value is "" (an empty String) on success. | <pre> m_Client = new Client()  //m_Client is a reference to the Client class in the Client DLL.  returnval as string  returnVal = remoteObject.SetAcquireParameter(id, device, suite, test, parameterString) if ((OP_STATUS) returnVal != OP_STATUS.SUCCESS) return CommandFailed(returnVal)                     </pre> |

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

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



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

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

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

### Select clock edge example

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

### Query the configuration parameters for a suite or measurement

| Command name          | Parameters   | Description   | Return value  | Example  |
|-----------------------|--|---|---|--|
| GetGeneralParameter() | <i>string clientID</i><br><i>string device</i><br><i>string suite</i><br><i>string test</i><br><i>string parameterString</i> | This method gets the general configuration parameters for a suite or measurement. | The return value is the general configuration parameter for a specified suite or measurement that is set. | <pre> m_Client = new Client()  //m_Client is a reference to the Client class in the Client DLL.  returnval as string  <a href="#">Query channel example</a>                     </pre> |

Table continued...

| Command name          | Parameters   | Description  | Return value   | Example  |
|-----------------------|--|--|--|--|
| GetAnalyzeParameter() | <i>string clientID</i><br><i>string device</i><br><i>string suite</i><br><i>string test</i><br><i>string parameterString</i> | This method gets the configuration parameters set in the Analyze panel of the Configuration Panel dialog box for a specified suite or measurement. | The return value is the configuration parameter set in the Analyze panel of the Configuration Panel dialog box for a specified suite or measurement. | <pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string</pre>  |
| GetAcquireParameter() | <i>string clientID</i><br><i>string device</i><br><i>string suite</i><br><i>string test</i><br><i>string parameterString</i> | This method gets the configuration parameters set in the Acquire panel for a specified suite or measurement.                                       | The return value is the configuration parameter set in the Acquire panel for a specified suite or measurement.                                       | <pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnVal = remoteObject.SetAcquireParameter(id, device, suite, test, parameterString) if ((OP_STATUS) returnVal != OP_STATUS.SUCCESS) return CommandFailed(returnVal)</pre> |

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



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

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

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

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

### Query parameter example

```
returnval=mClient.GetAnalyzeParameter(clientID, "Transmitter", "Drive", "5.3.9 - Random Jitter (RJ)", "Clock Edge")
```

## Select a test

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

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

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



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

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

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

## Select test example

To select measurement test 5.1.2 - OOB Burst Amplitude:

```
mClient.SelectsingleTest(clientId,"Transmitter","Drive","5.1.2 - OOB Burst Amplitude",true)
```

## Select a suite

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

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




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

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

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

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

## Select a channel

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



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

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



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

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

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

## Configure the selected measurement

Table 22:

| Command name          | Parameters   | Description  | Return value   | Example   |
|-----------------------|--|--|--|---|
| SetAnalyzeParameter() | <i>string clientID</i><br><i>string device</i><br><i>string suite</i><br><i>string test</i><br><i>string parameterString</i> | This method sets the configuration parameters in the Analyze panel of the Configuration Panel dialog box for a suite or measurement. | The return value is "" (an empty String) on success. | <pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string</pre> |

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

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



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

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

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

## Run with set configurations or stop the run operation

Table 23:

| Command name | Parameters             | Description   | Result value   | Example  |
|--------------|------------------------|---|--|--|
| Run()        | <i>string clientID</i> | Runs the selected tests<br><i>Note</i><br>After the server is set up and configured, run it remotely using this function. | String that gives the status of the operation after it has been performed.<br><br>The return value is "Run started..." on success. | <pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.Run(clientID)</pre> |

Table continued...

| Command name | Parameters             | Description                                    | Result value   | Example  |
|--------------|------------------------|--|--|--|
| Stop()       | <i>string clientID</i> | Stops the currently running tests. <i>Note</i> | String that gives the status of the operation after it has been performed.<br><br>The return value is "Stopped..." on success. | <pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.  returnval as string returnval=m_Client.Stop(clientID) xxx</pre> |

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

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



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

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

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

## Handle error codes

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

**Table 24: Error codes**

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

**Note:**

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



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

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

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

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

## Get or set the timeout value

Table 25:

| Command name | Parameters                                   | Description  | Return values  | Example  |
|--------------|--|--|--|--|
| GetTimeOut() | <i>string clientID</i>                       | Returns the current timeout period set by the client   | String that gives the status of the operation after it has been performed.<br><br>The default return value is 1800000.                     | <pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.GetTime-Out()</pre>                         |
| SetTimeOut() | <i>string clientID</i><br><i>string time</i> | Sets a timeout period specified by the client.<br><br>After this timeout period expires, the server is unlocked automatically. | String that gives the status of the operation after it has been performed.<br><br>On success the return value is "TimeOut Period Changed". | <pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.SetTime-Out(clientID, desiredTimeOut)</pre> |

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

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



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

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

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

## Wait for the test to complete

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



| Command name  | Parameters  | Description   | Return value   | Example   |
|---|---|---|--|---|
| ApplicationStatus()   | <i>string clientID</i>  | This method gets the status of the server application.<br><br>The states are <i>Ready</i> , <i>Running</i> , <i>Paused</i> , <i>Wait</i> , or <i>Error</i> .          | String value that gives the status of the server application   | <pre>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)</pre>                                      |
| QueryStatus()   | <i>string clientID</i><br><i>out string[] status</i>  | An interface for the user to transfer Analyze panel status messages from the server to the client   | String that gives the status of the operation after it has been performed.<br><br>On success the return value is "Transferred...".     | <pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string <a href="#">Query status example</a></pre>  |
| GetCurrentState- Info()<br><br> <b>Note:</b> This command is used when the application is running and is in the wait or error state. | <i>string clientID</i><br><i>out string WaitingMsbBxCaption</i><br><i>out string WaitingMsbBxMessage</i><br><i>out string[] WaitingMsbBxButtontexts</i> | This method gets the additional information of the states when the application is in Wait or Error state.<br><br>Except client ID, all the others are Out parameters. | This command does not return any value.<br><br>This function populates the Out parameters that are passed when invoking this function. | <pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL mClient.GetCurrentStateInfo(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxButtontexts)</pre> |

Table continued...

| Command name  | Parameters   | Description  | Return value                            | Example   |
|---|--|--|---|---|
| SendResponse()<br><br> <b>Note:</b><br>This command is used when the application is running and is in the wait or error state. | <i>string clientID</i><br><br><i>out string</i><br><i>WaitingMsbBxCaption</i><br><br><i>out string</i><br><i>WaitingMsbBxMessage</i><br><br><i>string</i><br><i>WaitingMsbBxResponse</i> | After receiving the additional information using the method GetCurrentStateInfo(), the client can decide which response to send and then send the response to the application using this function. The response should be one of the strings that was received earlier as a string array in the GetCurrentStateInfo function. The _caption and _message should match the information received earlier in the GetCurrentStateInfo function. | This command does not return any value. | <pre>m_Client = new Client()  //m_Client is a reference to the Client class in the Client DLL  mClient.SendResponse(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxResponse)</pre> |

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

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



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

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

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

### Query status example

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

## After the test is complete



| Command name                        | Parameters  | Description   | Return value  | Example   |
|-------------------------------------|---|---|---|---|
| GetPassFailStatus()                 | <p><i>string clientID</i></p> <p><i>string device</i></p> <p><i>string suite</i></p> <p><i>string test</i></p>  | <p>This method gets the pass or fail status of the measurement after test completion.</p> <p><b>Note:</b></p>  <p>Execute this command after completing the measurement.</p> | <p>String that gives the status of the operation after it has been performed</p> <p>Returns the pass or fail status in the form of a string</p> | <pre>m_Client = new Client()  //m_Client is a reference to the Client class in the Client DLL.  returnval as string  returnval=m_Client.GetPassFailStatus(clientID, device, suite, "5.3.9 - Random Jitter (RJ)")// Pass or Fail</pre> |
| GetResultsValue()                   | <p><i>string clientID</i></p> <p><i>string device</i></p> <p><i>string suite</i></p> <p><i>string test</i></p> <p><i>string ParameterString</i></p>                         | <p>This method gets the result values of the measurement after the run.</p>   | <p>String that gives the status of the operation after it has been performed</p> <p>Returns the result value in the form of a string</p>        | <pre>m_Client = new Client()  //m_Client is a reference to the Client class in the Client DLL.  returnval as string  returnval=m_Client.GetPassFailStatus(clientID, device, suite, "5.3.9 - Random Jitter (RJ)")</pre>                |
| GetResultsValueForSubMeasurements() | <p><i>string clientID</i></p> <p><i>string device</i></p> <p><i>string suite</i></p> <p><i>string test</i></p> <p><i>string ParameterString</i></p> <p><i>int rowNr</i></p> | <p>This method gets the result values for individual submeasurements after the run.</p>   | <p>String that gives the status of the operation after it has been performed</p> <p>Returns the result value in the form of a string</p>        | <pre>m_Client = new Client()  //m_Client is a reference to the Client class in the Client DLL.  returnval as string  <a href="#">Get results for a submeasurement example</a></pre>   |

Table continued...

| Command name         | Parameters  | Description  | Return value  | Example   |
|----------------------|---|--|---|---|
| GetReportParameter() | <p><i>string clientID</i></p> <p><i>string device</i></p> <p><i>string suite</i></p> <p><i>string test</i></p> <p><i>string ParameterString</i></p> | <p>This method gets the general report details such as oscilloscope model, TekExpress version, and SAS version.</p>  | <p>The return value is the oscilloscope model, TekExpress version, and SAS version.</p>   | <pre>m_Client = new Client()  //m_Client is a reference to the Client class in the Client DLL.  returnval as string  <b>Oscilloscope Model</b>  returnval=m_Client.GetReportParameter(clientID, "ScopeModel")  <b>TekExpress Version</b>  returnval=m_Client.GetReportParameter(clientID, "TekExpressVersion")  <b>SAS Version</b>  returnval=m_Client.GetReportParameter(clientID, "ApplicationVersion")</pre> |
| TransferReport()     | <p><i>string clientID</i></p> <p><i>string filePath</i></p>   | <p>This method transfers the report generated after the run.</p> <p>The report contains the summary of the run.</p> <p>The client must provide the location where the report is to be saved at the client-end.</p> | <p>String that gives the status of the operation after it has been performed</p> <p>Transfers all the result values in the form of a string</p> | <pre>m_Client = new Client()  //m_Client is a reference to the Client class in the Client DLL.  returnval as string  returnval=m_Client.TransferReport(clientID, "C:\Report")</pre>   |

Table continued...

| Command name     | Parameters                                       | Description  | Return value   | Example   |
|------------------|--|--|--|---|
| TransferImages() | <i>string clientID</i><br><i>string filePath</i> | <p>This method transfers all the images (screenshots) from the specified client and folder for the current run (for a suite or measurement).</p> <p><b>Note:</b></p> <p>Every time you click Start, a folder is created in the X: drive. Transfer the waveforms before clicking Start.</p>  | <p>String that gives the status of the operation after it has been performed</p> <p>Transfers all the images in the form of a string</p> | <pre>m_Client = new Client()  //m_Client is a reference to the Client class in the Client DLL.  returnval as string  returnval=m_Client.TransferImages(clientID, "C:\Waveforms"</pre> |

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

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



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

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

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

**Table 26: string parameterString**

| Name            | Type   | Direction | Description   |
|-----------------|--------|-----------|---|
| parameterString | string | IN        | Specifies the oscilloscope model, TekExpress version, and SAS version |

### Get results for a submeasurement example

This example returns the specified submeasurement results for test 5.3.9 - Random Jitter (RJ).

```
returnval=m_Client.GetResultsValue( clientID,"Transmitter", "Drive", "5.3.9 - Random Jitter(RJ)", "Value", 0)
```

```
returnval=m_Client.GetResultsValue( clientID,"Transmitter", "Drive", "5.3.9 - Random Jitter(RJ)", "Value", 1)
```



## Save, recall, or query a saved session

Table 27:

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

Table continued...

| Command name    | Parameters                                   | Description  | Return value  | Example   |
|-----------------|--|--|---|---|
| SaveSessionAs() | <i>string clientID</i><br><i>string name</i> | Saves the current session under a different name every time this method is called.<br><br>The client provides the session name | String that gives the status of the operation after it has been performed<br><br>The return value is "Session Saved..." | <pre>m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.SaveSessionAs(clientID,desiredSessionName)</pre> |

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

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



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

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

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

## Unlock the server

Table 28:

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

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

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



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

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

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

## Disconnect from the server

Table 29:

| Command name | Parameters             | Description  | Return value  | Example   |
|--------------|------------------------|--|---|---|
| Disconnect() | <i>string clientID</i> | This method disconnects the client from the server.<br><br><i>Note</i> | Integer value that gives the status of the operation after it has been performed<br><br>1 for Success<br><br>-1 for Failure | <pre>m_Client = new Client()  //m_Client is a reference to the Client class in the Client DLL.  returnval as string  returnval = m_Client.Disconn ect(m_clientID)</pre> |

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

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



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

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

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

## SAS test definitions

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

### Maximum noise OOB IDLE

Verifies that the maximum noise value during OOB idle is within the conformation limits.

The maximum noise during OOB idle is measured at the transmitter device output while the DUT is connected to the Zero-Length Test Load and transmitting OOB signaling. The maximum differential noise is measured during the period when there is no burst (idle). The maximum pk-pk noise value must be less than 120 mVppd in order to be considered conformant.

### OOB burst amplitude

Verifies that the OOB amplitude is within the conformation limits.

The DUT's transmitted OOB burst signaling is captured. The minimum amplitude is measured using the lowest amplitude bit in the burst. The maximum amplitude is measured as the maximum peak-to-peak differential amplitude across the entire burst. The minimum OOB amplitude must be  $\geq 240$  mVppd in order to be considered conformant. The maximum OOB amplitude must be  $\leq 1600$  mVppd in order to be considered conformant.

### OOB offset delta

Measures the absolute value of the OOB offset delta.

The measurement defines the OOB offset delta as the maximum difference in the average differential voltage (DC offset) component between the burst times and the idle times of an OOB signal. In this test, the OOB offset delta will be measured at the transmitter device output using a real-time DSO while the DUT is connected to the Zero-Length test load. The edges of the burst will be computed at the time points where the differential signal crosses the  $\pm 50$  mV thresholds. The differential waveform samples between these two points will be extracted and the mean value will be computed. (The average value of the differential signal during the idle times will not be directly measured, as it is understood to be zero volts due to the fact that the DUT will be connected to the DSO using DC blocking capacitors.) Thus, the mean value of the differential burst waveform samples will be computed as the OOB offset delta. The absolute value of the OOB offset delta must be less than or equal to 25 mV in order to be considered conformant.

### OOB common mode delta

Measures the OOB common-mode delta.

The measurement defines the OOB common mode delta as the maximum difference in the average common mode voltage between the burst times and the idle times of an OOB signal. In this test, the OOB common-mode delta will be measured at the transmitter device output using a real-time DSO while the DUT is connected to the Zero-Length test load. The common-mode signal will be computed as:

$$(TX_p - TX_n) / 2$$

The differential signal will be computed as  $(TX_p - TX_n)$ ,

where TXp and TXn are the positive and negative halves of the TX differential signal, respectively. The edges of the burst will be computed at the time points where the differential signal crosses the  $\pm 50$  mV thresholds. The common-mode waveform samples between these two points will be extracted and the mean value will be computed. (The average value of the common-mode signal during the idle times will not be directly measured, as it is understood to be zero volts due to the fact that the DUT will be connected to the DSO using DC blocking capacitors.) Thus, the mean value of the common-mode burst samples will be computed as the OOB common-mode delta.

## SSC modulation type

This measurement deduces the SSC modulation type used. The SSC can be up-spread, down spread, or center spread. This measurement is required only for devices which support SSC. This measurement is informational only.

## SSC modulation frequency

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

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

## SSC modulation deviation

This measurement verifies that the max SSC Spread is less than 0.23% or less than 2300 ppm.

The deviation of the modulation will be determined by measuring maximum and minimum profile peak values per period, over at least 10 complete SSC cycles. From these values, the average maximum and average minimum peak values will be computed, and the results compared against the requirements listed above for the appropriate SSC modulation types.

## SSC modulation balance

This measurement verifies that the Modulation balance is less than 288 ppm.

In Modulation balance the difference between the amount of up spreading (i.e., averaged maximum peak level) and down spreading (averaged minimum peak level) must be no greater than 288 ppm. The average peak level are calculated over at least 10 SSC cycles. For the purposes of this test, this is formally calculated as:

Deviation Asymmetry =  $\text{abs}(\text{averaged max peak level} - \text{averaged minimum peak value})$ .

## SSC DFDT

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

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

## Physical link rate long term stability

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

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

The inverse of the UI values will be taken to produce an array of instantaneous frequency values. The low pass filter will then be applied to the inverse of the UI values. The resulting waveform that is produced at the output of the test filter will then be converted to ppm. This result will represent the instantaneous bitrate of the DUT. All of the values must be between +100 ppm and -100 ppm of 12.0 Gbps in order to be considered conformant.

## Common mode RMS voltage

This test is to calculate the common mode RMS voltage of the DUT signal. The differential signal is defined as: the difference of the TXp(positive waveform) and TXn (negative) waveforms, a transmitter's common-mode signal is mathematically computed as the average

of the TXp and TXn waveforms ( $V_{cm} = (TX_p - TX_n) / 2$ )

The characteristics of a device's common-mode signal are valuable to look at primarily because the common-mode signal provides a measure of how symmetric the positive and negative halves are of the differential signal. The common mode signal will also have high rms value if there is Skew between the positive and negative lanes.

## Common mode spectrum

This measurement calculates the AC Common mode spectrum at Fundamental frequency and at second harmonics. If the channels contain skew, then most of the energy will be in the first and second harmonics.

In general, the specification says that the Common Mode spectrum should be below a certain value for every point between a certain range of frequency. Therefore, a frequency mask is usually a line and all points of spectrum must be below this mask. The Pass/Fail limits are also given by Frequency Mask Hits.

## Peak-to-peak voltage

This test is to calculate the maximum differential Voltage of the DUT signal. This measurement is used to calculate the maximum differential voltage. The Vp-p measurement is simply the absolute peak-to-peak voltage of the DUT's transmitted signaling.

## VMA

This measurement is used to calculate the post-cursor ratio. This is a custom measurement for SAS 3.

This measurement will be available in DPOJET standard tab for standard SAS 3. The following definitions are used in this measurement, where VMA is v2 - v5:

- T is the symbol period
- t1 is the symbol period
- t2 is the zero-crossing point of the falling edge of the positive 5 UI CID
- t3 is the zero-crossing point of the falling edge of the positive 5 UI CID
- t4 is the zero-crossing point of the rising edge of the positive 5 UI CID
- v1 is the maximum voltage measured in the interval t1 to t1 + T
- v2 is the average voltage measured in the interval t1 + 2T to t1 + 3T
- v3 is the maximum voltage measured in the interval t2 - T to t2
- v4 is the minimum voltage measured in the interval t3 to t3 + T
- v5 is the average voltage measured in the interval t3 + 2T to t3 + 3T
- v6 is the maximum voltage measured in the interval t4 - T to t4

## Equalization

The transmitter equalization measurement shall be based on the following values:

- VMA: a mode (i.e., the most frequent value of a set of data) measurement
- VP-P: a peak-to-peak measurement with a repeating 7Eh (i.e., D30.3) pattern (see the phy test patterns in the Protocol-Specific diagnostic page in SPL-2)

The VMA and Vp-p measurements must be made with the transmitter device terminated through the interoperability point into a zero length test load.

The VMA and Vp-p measurements must be made using an equivalent time sampling oscilloscope with a histogram function with the following or an equivalent procedure:

- Determine VMA and Vp-p as discussed in this section.
- The following formula must be used to calculate the following transmitter equalization value:

$$\text{Transmitter equalization} = 20 \times \log_{10} (\text{VP-P} / \text{VMA}) \text{ dB}$$

## Rise time

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

This is a DPOJET based measurement. DPOJET is set in such a way that rise time/fall time is calculated as the time to rise from 20% to 80% of the waveform and vice versa. A D10.2 signal of 1010 pattern is used to make sure that the reference levels can be calculated without errors. This will lead to a correct estimation of rise time.

## Fall time

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

This is a DPOJET based measurement. DPOJET is set in such a way that rise time/fall time is calculated as the time to rise from 20% to 80% of the waveform and vice versa. A D10.2 signal of 1010 pattern is used to make sure that the reference levels can be calculated without errors. This will lead to a correct estimation of rise time.

## Random Jitter (RJ)

Random Jitter is a measure of how the random event affects the signal. Lower jitter means less errors and higher jitter means more errors.

This test will be performed while the DUT is transmitting D24.3 pattern. The specification requires that the measurement include the effects of the JTF (Jitter Transfer Function), which is a Standard-defined weighting function that is intended to separate the low-frequency timing variations due to SSC from the actual jitter. The RJ measurement requires that the Jitter Measurement Device (JMD) be configured to use the proper JTF characteristics required by the Standard.

## Total Jitter (TJ)

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

Measurements are made on the differential signal.

## Waveform Distortion Penalty (WDP)

The Standard defines WDP as “A simulated measure of the deterministic penalty of the signal waveform from a particular transmitter device transmitting a particular pattern and a particular test load with a reference receiver device.” It also describes it as “A characterization of the signal output within the reference receiver device after equalization.” The latter description may be conceptually easier to understand, showing that the WDP measurement is an example of what the DUT’s transmitted signaling would ‘look like’ to a receiver device after passing through an interconnect (such as a channel, backplane, cable, etc.), and being received and processed by an equalizer circuit inside the receiver device. Because it is not typically possible to observe the signal at this point (as it is conceptually located inside the actual receiver IC, post-equalization) it is not possible to practically measure this signal; however, it can be mathematically computed based on a reference model of a SAS interconnect and a reference receiver equalizer. This mathematical modeling is performed by a set of MATLAB code that is included as part of the Standard.

This measurement runs the Matlab code and determine the WDP value and report it.

## SAS3\_EYEOPENING

The SAS-3 specifications mention the eye opening. As the signal makes its transition into a well-isolated transmission path (such as a backplane) it has to manage the significant frequency dependant losses that can be over 20 dB at the first harmonic. The amount of data dependant jitter (DDJ/ISI) generated in this channel is large, but if it is pure ISI, it can be compensated for with a Decision Feedback Equalizer (DFE) circuit at the receiver. The eye will be completely closed, but with the SAS specified reference DFE, near full recovery of the signal is possible.

SAS is currently considering using a 5 tap DFE. The signal that contains a large amount of crosstalk/ISI and an almost closed eye is passed through a DFE filter to recover all of its amplitude. The algorithm should contain a method for calculating the optimized DFE parameter based on the data by passing the signal through the DFE filter and calculating the eye opening. Currently, specifications do not define the methodology for calculating the DFE coefficients. Currently, equalization settings are used that will result in the least square error of the equalized signal. This can be understood by looking at the points sampled in the middle of the eye. These points should be either a digital 1 or a digital 0. This means that their analog amplitude should optimally only take two values: either +T or -T, for a digital 1 or a digital 0, respectively. You will get +T+error or -T+error, "error" being what your equalizer cannot correct. The algorithm is trying to reduce the sum of the error<sup>2</sup> terms over a very long sequence of bits.

## Pre Cursor Equalization Ratio

This measurement is used to calculate the precursor ratio. The following definitions are used in this measurement, where the precursor equalization is defined as  $R_{pre}=V3/V2$ :

- T is the symbol period
- t1 is the zero-crossing point of the rising edge of the positive 5 UI CID
- t2 is the zero-crossing point of the falling edge of the positive 5 UI CID
- t3 is the zero-crossing point of the falling edge of the positive 5 UI CID
- t4 is the zero-crossing point of the rising edge of the positive 5 UI CID
- v2 is the average voltage measured in the interval  $t1 + 2T$  to  $t1 + 3T$
- v3 is the maximum voltage measured in the interval  $t2 - T$  to  $t2$

## Post Cursor Equalization Ratio

This measurement is used to calculate the postcursor ratio. The following definitions are used in this measurement, where the postcursor equalization is defined as  $R_{post}=V1/V2$ :

- T is the symbol period
- t1 is the zero-crossing point of the rising edge of the positive 5 UI CID
- t2 is the zero-crossing point of the falling edge of the positive 5 UI CID
- t3 is the zero-crossing point of the falling edge of the positive 5 UI CID
- t4 is the zero-crossing point of the rising edge of the positive 5 UI CID
- v2 is the average voltage measured in the interval  $t1 + 2T$  to  $t1 + 3T$
- v1 is the maximum voltage measured in the interval  $t1$  to  $t1 + T$

## VHL (transition bit voltage Pk-Pk)

This custom measurement for SAS 3 is for measuring the peak to peak transition bit voltage. This measurement is available in DPOJET standard tab for standard SAS 3. The following definitions are used in this measurement, where VHL is the peak-to-peak voltage measured in the interval  $t1$  to  $t1 + 80T$ :

- T is the symbol period



- $t_1$  is the zero-crossing point of the rising edge of the positive 5 UI CID

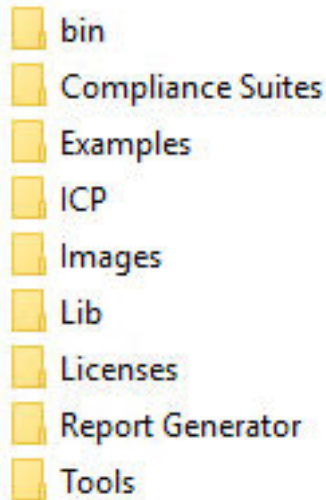
## Unit Interval

This measurement is for measuring the unit interval or bitrate of a signal. It verifies that the unit interval is within the conformation limits. This measurement stores the waveform files and calls the DPOJET period measurement to provide results.

# References

## Application directories

You can find the application files at *C:\Program Files\Tektronix\TekExpress SAS Transmitter Automated Solution*. The application directory and associated files are organized as follows:



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

**Table 30: Application directories and usage**

| Directory names   | Usage  |
|-------------------|--|
| Bin               | Contains application libraries                                   |
| Compliance Suites | Contains test suite specific files                               |
| Examples          | Contains various support files                                   |
| ICP               | Contains instrument and application specific interface libraries |
| Images            | Contains images of the application                               |
| Lib               | Contains utility files specific to the application               |
| Licenses          | Contains all the license files                                   |
| Report Generator  | Contains style sheets for report generation                      |
| Tools             | Contains instrument and application specific files               |

## File name extensions

The TekExpress SAS Transmitter Automated Solution software uses the following file name extensions:

**Table 31: File name extension**

| File name extension | Description   |
|---------------------|---|
| *.TekX              | Application session files (the extensions may not be displayed) |
| *.py                | Python sequence file.   |

Table continued...

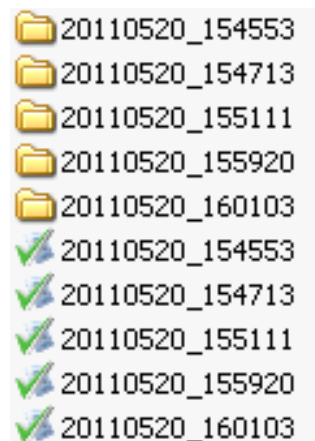
| File name extension | Description   |
|---------------------|---|
| *.xml               | Test-specific configuration information (encrypted) files.<br>Application log files |
| *.csv               | Test result reports<br>Plot data  |
| *.mht               | Test result reports (default)<br>Test reports can also be saved in HTML format      |
| *.pdf               | Test result reports<br>Application help document                                    |
| *.xslt              | Style sheet used to generate reports  |
| *.png               | Captured images   |
| .wfm                | Test waveform file  |

## View test-related files

Files related to tests are stored in My Documents\TekExpress SAS Transmitter Automated Solution\Untitled session folder. Each test setup in this folder has both a test setup file and a test setup folder, both with the test setup name. The test setup file is preceded by the TekExpress icon.

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

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



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

The first time you run a new, unsaved session, the session files are stored in the Untitled Session folder located at X:\TekExpress SAS Transmitter Automated Solution. 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 application.

# Index

## A

- About SCPI command [57](#)
- Acquisition tab
  - Acquisition [34](#)
- Activate the license
  - license [13](#)
- Application controls
  - Application controls menu [23](#)
- Application directories [130](#)

## C

- Configuration tab
  - Configuration [36](#)
- Configure report view settings [47](#)
- Configure the test setup [29](#)
- Connected instruments
  - searching for [28](#)
- Contacting Tektronix [11](#)
- Conventions [10](#)

## D

- Delete the run/config session [93](#)
- Downloading and installing the software
  - Download the software
  - Install the software [13](#)
- DUT
  - DUT tab [30](#)
- DUT settings [30](#)

## E

- Email notification and setup [26](#)

## F

- File name extensions [130](#)
- Filter the test results [43](#)

## G

- Getting help and support [10](#)
- Getting started [12](#)
- GPIB [28](#)

## H

- Hardware requirements
  - Hardware [12](#)

## I

- Icons used
  - symbols [10](#)
- Instrument Control Settings [16](#)
- Instruments
  - discovering connected [28](#)

## L

- Launch the application [22](#)
- license key [14](#)
- Load the run/config session [93](#)
- load the saved test setup
  - load the test setup [51](#)
- Log view [42](#)
- Log View [41](#)

## M

- measurement limits for tests [36](#)
- Multiple-session run [32](#)

## N

- Non-VISA [28](#)

## O

- Options menu
  - Instrument control settings [28](#)
  - Options menu functions [25](#)
- Override the run/config session [94](#)

## P

- Preferences menu [43](#)
- Preferences tab
  - Preferences [40](#)
- Product documents
  - TekExpress SAS Transmitter Automated Solution documents [10](#)

## Q

- Query the available list in the run/config session [94](#)
- Query the current run/config session [94](#)

## R

- recalling test setup [50](#)
- References [130](#)
- Remote commands [57](#)
- report generation [44](#)
- report generation options [44](#)

- report generation settings
  - Configure report generation settings [44](#)
- Reports
  - receiving in email notifications [26](#)
- Reports panel [44](#)
- Results panel [43](#)
- Run the run/config saved session [93](#)

## S

- Save the configured test setup
  - Save the test setup [50](#)
- Save the run/config sessions [92](#)
- Save the test setup [56](#)
- Save the test setup with a different name [56](#)
- Saving test setup [50](#)
- SCPI command [57](#)
- SCPI Commands
  - SCPI [57](#)
- Search instruments connected
  - instruments connected to the application [16](#)
- Select a loaded test setup [52](#)
- Select a pre-run session loaded test setup [52](#)
- Select the tests [33](#)
- Set the default session [92](#)
- Setting up the test environment
  - test environment [15](#)
- Setup panel [29](#)
- Socket configuration for SCPI commands
  - Socket configuration [57](#)
- Software requirements
  - Softwares [13](#)
- software version [14](#)
- Start the application [22](#)
- Status panel [41](#)
- Support [11](#)

## T

- Technical support [11](#)
- Tek Link [28](#)
- TEKEXP:\*IDN? [80](#)
- TEKEXP:\*OPC? [86](#)
- TEKEXP:ACQUIRE\_MODE LIVE [80](#)
- TEKEXP:ACQUIRE\_MODE? [80](#)
- TEKEXP:COPYIMAGES <DestinationPath> [91](#)
- TEKEXP:EXIT [85](#)
- TEKEXP:INFO? IMAGE [79](#)
- TEKEXP:INFO? REPORT [79](#)
- TEKEXP:INFO? WFM [79](#)
- TEKEXP:INSTRUMENT, "<InstrumentType>","<value>" [78](#)
- TEKEXP:INSTRUMENT? "<InstrumentType>" [78](#)
- TEKEXP:LASTERROR? [86](#)
- TEKEXP:LIST? DEVICE [76](#)
- TEKEXP:LIST? INSTRUMENT,"<Real Time Scope>" [78](#)
- TEKEXP:LIST? SUITE [77](#)
- TEKEXP:LIST? TEST [77](#)
- TEKEXP:LIST? VERSION [77](#)
- TEKEXP:MODE COMPLIANCE [81](#)

- TEKEXP:MODE? [81](#)
- TEKEXP:POPOP "Yes" [86](#)
- TEKEXP:POPOP? [86](#)
- TEKEXP:REPORT GENERATE [81](#)
- TEKEXP:REPORT? "<Device Field>" [82](#)
- TEKEXP:REPORTASXML? [91](#)
- TEKEXP:RESULT? "<TestName>" [82](#)
- TEKEXP:RESULT? "<TestName>","<ColumnName>" [82](#)
- TEKEXP:RESULT? "<TestName>","<ColumnName>","<RowNumber>" [82](#)
- TEKEXP:SELECT DEVICE,"DeviceName" [63](#)
- TEKEXP:SELECT SUITE,"<SuiteName>" [63](#)
- TEKEXP:SELECT TEST,"<ALL>" [64](#)
- TEKEXP:SELECT TEST,"<TestName>","1" [64](#)
- TEKEXP:SELECT VERSION,"<VersionName>" [64](#)
- TEKEXP:SELECT? DEVICE [63](#)
- TEKEXP:SELECT? SUITE [63](#)
- TEKEXP:SELECT? TEST [64](#)
- TEKEXP:SELECT? VERSION [64](#)
- TEKEXP:SELECTID "<TestID>" [92](#)
- TEKEXP:SESSION DEFAULT [92](#)
- TEKEXP:SESSION DELETE, "Session1, Session2" [93](#)
- TEKEXP:SESSION LOAD, "Session Name" [93](#)
- TEKEXP:SESSION RUN, "Session Name's separated by comma" [93](#)
- TEKEXP:SESSION SAVE, "SessionName", "True" [94](#)
- TEKEXP:SESSION SAVE, "Session Name" [92](#)
- TEKEXP:SESSION? CURRENT [94](#)
- TEKEXP:SESSION? LIST [94](#)
- TEKEXP:SETUP Default [83](#)
- TEKEXP:SETUP Open,"<SessionName>" [84](#)
- TEKEXP:SETUP Save [83](#)
- TEKEXP:SETUP Save,"<SessionName>" [83](#)
- TEKEXP:SETUP? CURRENT [84](#)
- TEKEXP:STATE RUN [84](#)
- TEKEXP:STATE? [85](#)
- TEKEXP:STATE? SETUP [85](#)
- TEKEXP:TESTINFO? [92](#)
- TEKEXP:VALUE
- ACQUIRE,"<TestName>","<AcquireType>","<ParameterName>","<ParameterName>","<ParameterName>" [76](#)
- TEKEXP:VALUE
- ANALYZE,"<TestName>","<ParameterName>","<ParameterName>" [76](#)
- TEKEXP:VALUE ContinuousRun\_Duration,"<Value>" [89](#)
- TEKEXP:VALUE ContinuousRun\_RunSessionOptions,"Value" [90](#)
- TEKEXP:VALUE ContinuousRun,"<Value>" [89](#)
- TEKEXP:VALUE DUTID,"Value" [80](#)
- TEKEXP:VALUE GENERAL,"<ParameterName>","<Value>" [65](#)
- TEKEXP:VALUE GENERAL,"View Report After Generating",<value> [90](#)
- TEKEXP:VALUE
- LIMIT,<TestName>,<LimitHeader>,<Value1>,<CompareString>,<Value2> [87](#)
- TEKEXP:VALUE VERBOSE,"<Value>" [88](#)
- TEKEXP:VALUE
- WFMLE,"<TestName>,<AcquireType>,<WaveformFileName>" [87](#)
- TEKEXP:VALUE?
- ACQUIRE,"<TestName>","<AcquireType>","<ParameterName>" [76](#)
- TEKEXP:VALUE? ANALYZE,"<TestName>","<ParameterName>" [76](#)
- TEKEXP:VALUE? ContinuousRun [89](#)
- TEKEXP:VALUE? ContinuousRun\_Duration [89](#)

TEKEXP:VALUE? ContinuousRun\_RunSessionOptions [90](#)  
TEKEXP:VALUE? DUTID [80](#)  
TEKEXP:VALUE? GENERAL,"<ParameterName>" [65](#)  
TEKEXP:VALUE? GENERAL,"Enable Continuous Run" [88](#)  
TEKEXP:VALUE? GENERAL,"View Report After Generating" [90](#)  
TEKEXP:VALUE? LIMIT,<TestName>,<LimitHeader> [87](#)  
TEKEXP:VALUE? VERBOSE [88](#)  
TEKEXP:VALUE? WFMFILE,<TestName>,<AcquireType> [87](#)  
test execution status [41](#)  
Test results  
    send by email [26](#)  
test run preferences [40](#)  
Test Selection  
    Test Selection tab [33](#)  
Test setup files overview  
    Test setup files [50](#)  
Test Status [41](#)

## V

View a report [49](#)  
View summary of test results  
    summary of test results [43](#)  
View test execution logs [42](#)  
View the test execution status [41](#)

## W

waveform acquisition settings [34](#)