TekExpress BroadR-Reach (100BASE-T1) Analysis Solution

Printable Application Help

077-0881-03

TekExpress BroadR-Reach (100BASE-T1) Analysis Solution

Printable Application Help

www.tek.com
077-0881-03

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

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

Contacting Tektronix

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

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

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

Table of Contents

Welcome v

Operating Basics

AWG\AFG Automation 1
Related Documentation 3
Application Directories and Usage 3
View Test-Related Files 5
File Name Extensions 6

Getting Started

Compatibility 7
Minimum System Requirements 8
Install the Software 10
Activate the License 11
View Version and License Information 12
Start the Application 12
Options Menu 14
View Connected Instruments: the Instrument Control Settings 15
Application Controls 16
Exit the Application 17

Application Basics

Application Basics 19

Setting Up Tests

Setting Up Tests: the Setup Panel 21
Selecting Device Parameters: the DUT tab 23
Choosing Tests: the Test Selection tab 24
Selecting Acquisitions 26
- Acquiring Waveforms: the Acquisitions tab 26
- Acquire Live Waveforms for Analysis 29
- Use Prerecorded Waveforms for Analysis 30
- Acquire Step By Step 30

Configuring Tests: the Configuration Screen 33
Compliance Mode or User Defined Mode 36

Measurement Parameter Descriptions 37
Calibration 40

Running Tests

Before You Click Start 43
Pre-Run Check List 46
Configure Email Notification 47
Running the Tests and Viewing their Progress: the Status Panel 48

Viewing Plots

Viewing Plots 51

Viewing Results

Viewing Test Results: the Results Panel 59

Configuring and Viewing Reports

Configuring and Viewing Reports: the Reports Panel 61

Saving and Recalling Test Setups

About Saving and Recalling Test Setups 65
Saving a Test Setup 66
Recalling a Saved Test Setup 66
Creating a New Test Setup Based on an Existing One 67
Deleting a Test Setup 67

BroadR-Reach measurements

Transmitter Clock Frequency 69
Transmitter Timing Jitter Master and Slave 71
Transmitter Output Droop 77
Transmitter Power Spectral Density 79
Transmitter Distortion 82
Return Loss 95
Known Limitations 105

Measurement Error Messages

Measurement Error Messages 107

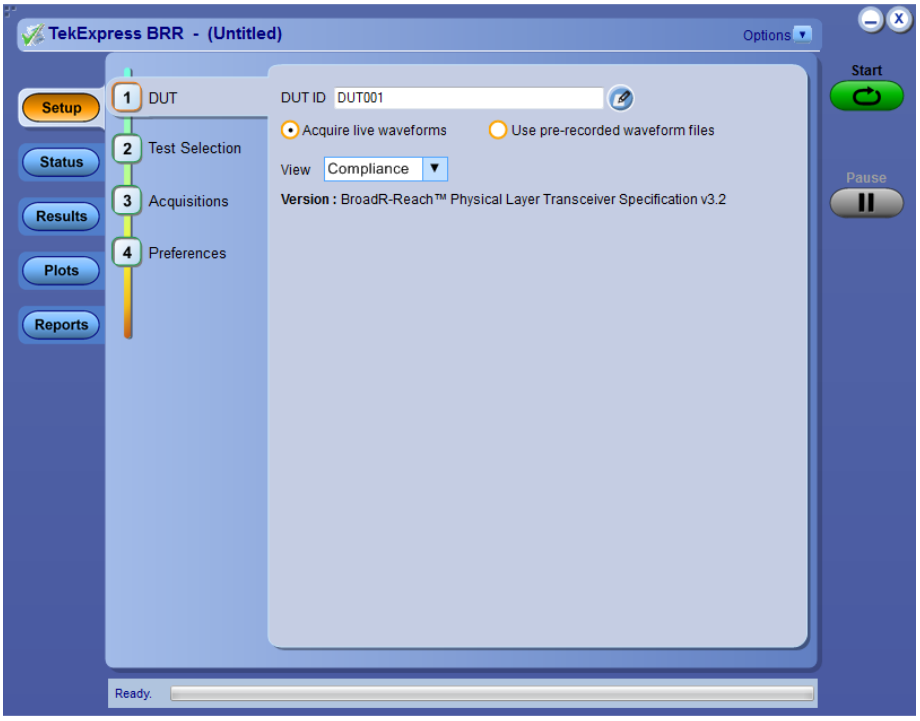
TekExpress Programmatic Interface

About the Programmatic Interface 109
Requirements for Developing TekExpress Client 110
Remote Proxy Object 111
Client Proxy Object 112
Program Examples 113
Client Programmatic Interface Example 116
BRR Application Commands 119
- About TekExpress BRR Application Commands 119
- Connect Through an IP Address 119
- Lock the Server 121
- Disable the Popups 122
- Set or Get the DUT ID 123
- Set the Configuration Parameters for a Suite or Measurement 125
- Query the Configuration Parameters for a Suite or Measurement 127
- Select a Measurement 130
- Select a Suite 132
- Run with Set Configurations or Stop the Run Operation 134
- Get or Set the Timeout Value 135
- Wait for the Test to Complete 137
- After the Test is Complete 140
- Save Recall or Check if a Session is Saved 146
- Unlock the Server 148
- Disconnect from the Server 149
- Handle Error Codes 150

Welcome

The TekExpress BroadR-Reach (100BASE-T1) (BRR) is an automated compliance test solution for performing electrical characterization of the BroadR-Reach transmitter as per OPEN Alliance BroadR-Reach (OABR) specification version 3.2. The IEEE P802.3bw D3.3 (100BASE-T1) specification has a stated objective to provide electrical interoperability for 100 Mb/s client interface in reference to the OABR PHY v3.2 specification. BRR and 100BASE-T1 maintain the same electrical test requirements as it relates to the Physical Medium Attachment (PMA) transmitter electrical specifications.

BroadR-Reach (100BASE-T1) is an Ethernet-based point-to-point technology for automotive applications. It provides five test mode signals for electrical characterization of the Physical Medium Attachment (PMA) sublayer.



Supported Tests

- *Transmit Clock Frequency*
- *Transmitter Timing Jitter Master and Slave*
- *Transmitter Output Droop*
- *Transmitter Power Spectral Density*
- *Transmitter Distortion*
- *Return Loss*

Test modes

Required test mode	BroadR-Reach measurement
1	Transmitter Output Droop
2	Transmitter Timing Jitter ■ Transmitter Timing Jitter-Master Jitter ■ Transmitter Timing Jitter-Slave Jitter Transmit Clock Frequency
3	Transmitter Distortion Return Loss
4	Transmitter Power Spectral Density

See also:

Application Basics

Equipment Connection Setup

Install the Software

Application Directories and Usage

File Name Extensions

Operating Basics

AWG\AFG Automation

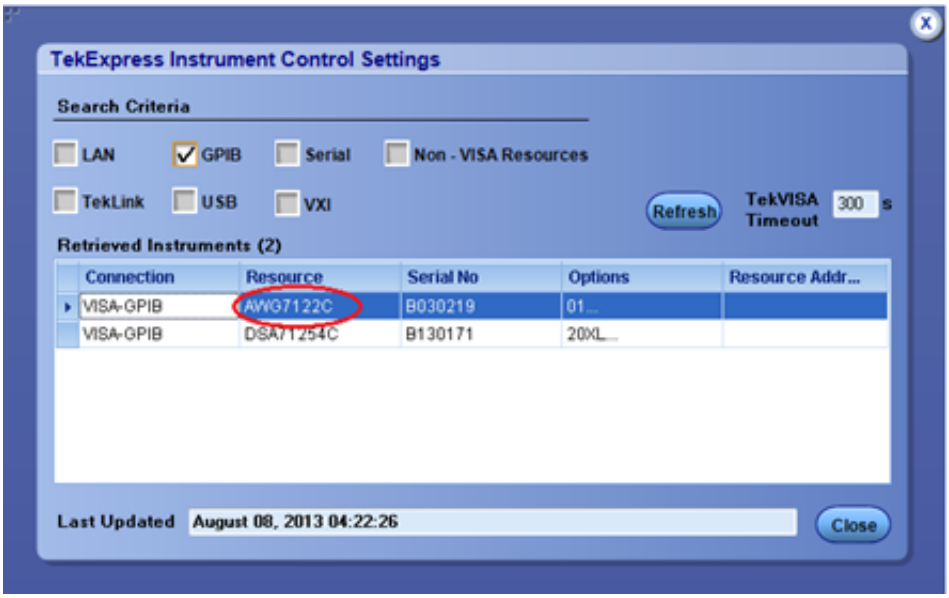
AWG\AFG automation is supported in the TekExpress BroadR-Reach (100BASE-T1) application for the following measurements. Before selecting the AWG\AFG automation feature, make sure that a GPIB connection is established between the Oscilloscope and the AWG\AFG. The AFG automation depends on the measurement used for testing

- *Transmitter Distortion with disturbing signal*: AWG\AFG is used to transmit a disturbing signal. CH1 and CH1_inverted are used.
- *Return Loss measurement and calibration*: AWG\AFG is used to transmit a wide band signal. CH1 and CH1_inverted are used for transmission of the signal. A marker signal has to be connected to the auxiliary channel of the Oscilloscope (used as trigger source).

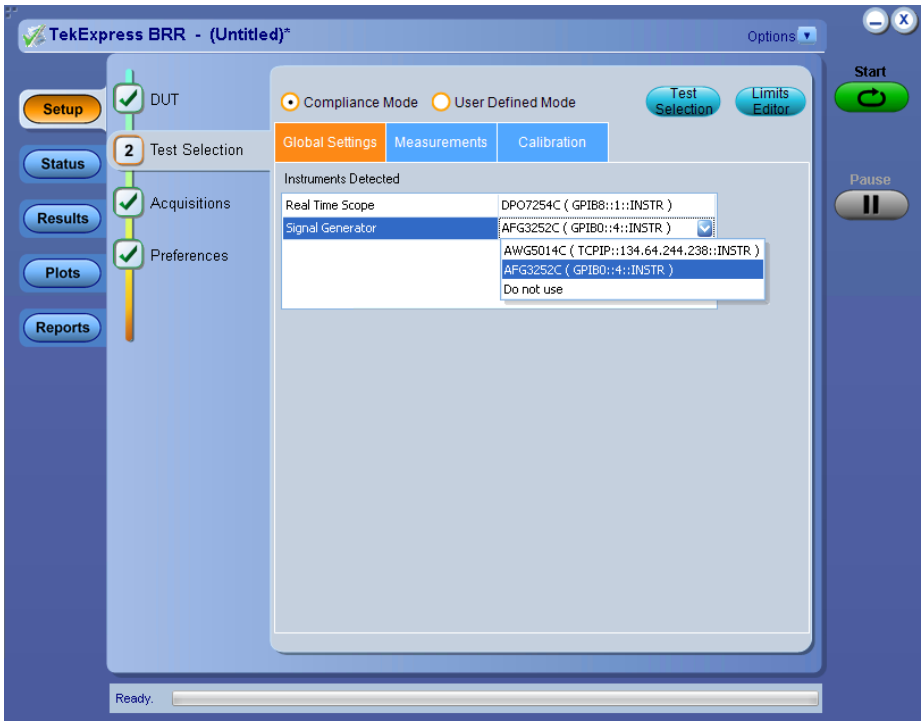
How BRR Automates AWG \AFG

Automation of the AWG\AFG is taken care in the BRR application in the following way:

1. Refresh the Instruments Control Settings to observe the connected AWG \AFG through GPIB.



2. Once the AWG\AFG is listed as shown above, go to the **Global Settings** tab from the Configuration menu of the Tests. If the AWG\AFG is supported, it will be listed as a drop-down menu option next to the **Automate with AWG \AFG** label. By default, the application will consider the connected AWG to be used for automation.



3. If you do not want to automate the use of an AWG\AFG, select the **Do not use** option in the drop-down menu for that parameter, and then start the testing.

NOTE. *If **Automate with AWG\AFG** is set as **Do not use**, you will need to manually copy the AWG\AFG waveforms from the Oscilloscope to the AWG \AFG. Waveforms are available at C:\Program Files\Tektronix\TekExpress \TekExpress BroadR-Reach\AWG Waveforms.*

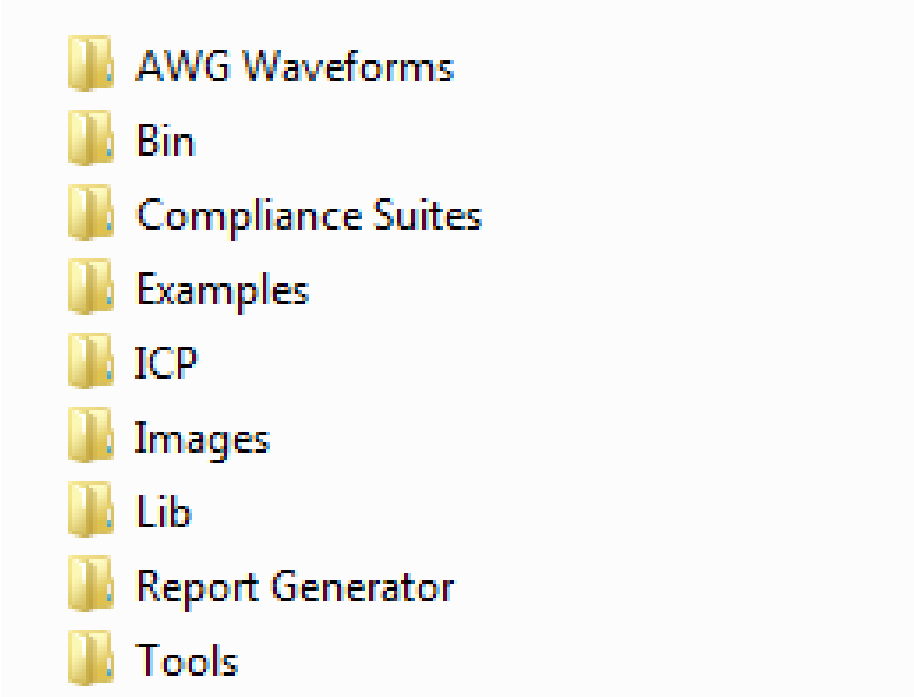
Related Documentation

The following information is available as part of the CPHYXpress documentation set.

Item	Purpose	Location
Help	Application operation and User Interface help	Application Help menu
PDF of the help	Printable version of the compiled help	PDF file that ships with CPHYXpress software. Downloadable from www.tek.com

Application Directories and Usage

The application directory and associated files are organized in the C:\Program Files\Tektronix\TekExpress\TekExpress BroadR-Reach folder



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

Table 1: Application directories and usage

Directory names	Usage
InstallDir\TekExpress\TekExpress BroadR-Reach	Contains the application and associated files.
TekExpress BroadR-Reach\AWG Waveforms	Contains AWG waveforms (AWG 5K, 7K and AFG waveforms) that get used in return loss and distortion with disturbing signal .
TekExpress BroadR-Reach\Bin	Contains miscellaneous BroadR-Reach application libraries.
TekExpress BroadR-Reach\Compliance Suites	Contains compliance-specific files and filter files.
TekExpress BroadR-Reach	Contains the Manuals.
TekExpress BroadR-Reach\Examples	Contains various support files.
TekExpress BroadR-Reach\ICP	Contains instrument and BroadR-Reach application-specific interface libraries.
TekExpress BroadR-Reach\Images	Contains images required for the application.
TekExpress BroadR-Reach\Lib	Contains utility files specific to the BroadR-Reach application.
TekExpress BroadR-Reach\Report Generator	Contains style sheets for report generation.
TekExpress BroadR-Reach\Tools	Contains instrument and BroadR-Reach application-specific files.

See Also

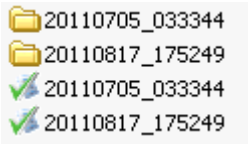
View Test-Related Files

View Test-Related Files

Files related to BRR tests are stored in the X:\BRR\Untitled Session shared folder. In the BRR folder, each test setup has a test setup file and a test setup folder, both with the test setup name. The test setup file is preceded by the BRR icon and usually has no file extension displayed.

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

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



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

The first time you run a new, unsaved session, the session files are stored in the X:\BRR\Untitled Session folder. Once you name and save the session, the Untitled Session folder name is changed to the one you specified.

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

See Also *File Name Extensions*

File Name Extensions

The TekExpress BRR software uses the following file name extensions:

File name extension	Description
.TekX	Session files are saved in this format but the extensions may not be displayed
.py	The test sequence file
.xml	The encrypted XML file that contains the test-specific configuration information The log file extension is also xml
.wfm	The test waveform file
.mht .pdf	Test result reports are saved in this format by default. Test reports can also be *saved in MHTML format* and .pdf.
.cal	Calibration file used with transmitter tests

See Also *Application Directories and Usage*

Getting Started

Compatibility

Supported Oscilloscopes

The TekExpress BRR application runs on the following Tektronix Oscilloscopes with bandwidths of 1 GHz or greater:

- DPO5104, DPO5204, MSO5104, MSO5204
- DPO5104B, DPO5204B, MSO5104B, MSO5204B
- DPO7104C, DPO7254C, DPO7354C
- DPO70404C, DPO70604C, DPO70804C, DPO71254C, DPO71604C, DPO72004C
- DPO/DSA 70KD series
- MSO70404C, MSO70604C, MSO70804C, MSO71254C, MSO71604C, MSO72004C
- DPO72504DX, DPO73304DX, DPO72304DX, DPO72504DX, DPO73304DX, MSO72304DX, MSO72504DX, MSO73304DX

Supported AWG/AFGs

The TekExpress BRR application supports the following Tektronix arbitrary waveform and arbitrary function generators:

- AWG7102 (Option 2): 10 GS/s
- AWG5014: 1.2 GS/s, 14-bit AWG
- AWG7122C: 12 GS/s
- AFG3252: 2 GS/s, 14-bit AFG
- AFG3102: 1 GS/s, 14-bit AFG

NOTE.

- *You may use AWG automation for Return Loss and Transmitter Distortion with Disturbing Signal measurements.*
- *AFG3100 & AFG3200 (having limit up to 50 MHz) cannot be used for Return Loss measurements.*
- *Without an amplifier, the AWG7000 series generates a maximum voltage of 2 Volt peak-to-peak signal, which cannot meet the 5.4 Volt requirement for the disturber signal for Transmitter Distortion test.*

Probes

The TekExpress BRR application supports the following Tektronix differential probes:

- TDP1500, TDP3500, P6248, and P6247 differential probes

Test Fixtures

The TekExpress BRR application supports the TF-GBE-BTP and TF-BRR-CFD test fixtures.

See Also

Minimum System Requirements

Minimum System Requirements

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

Table 2: System requirements

Oscilloscope	For a list of compatible Oscilloscopes, see *Compatibility*.
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 [1]
Firmware	TekScope 7.6.0 and later (Windows 7, 64-bit only)
Software	■ TekExpress Framework version 4.0 ■ Iron Python 2.7.3 ■ PyVisa 1.0.0.25 ■ Microsoft .NET 4.0 framework ■ Microsoft Internet Explorer 6.0 SP1 or later ■ Adobe Reader 8.0 or equivalent software for viewing portable document format (PDF) files

[1] If TekExpress is running on an instrument having a video resolution lower than 800 x 600. it is recommended that you connect a secondary monitor, which must be enabled before launching the application.

Probes	Following differential probes are supported: ■ All measurements: P6247, P6248, P6330,TDP1500, TDP3500. For more details, refer to the BroadR-Reach measurements section.
BRR Fixtures	TF-GBE-BTP, TF-BRR-CFD
Other Devices	■ Microsoft compatible mouse or compatible pointing device ■ Four USB ports (two USB ports minimum) ■ PCI-GPIB or equivalent interface for instrument connectivity [2]
Frequency Divider	The TF-BRR-CFD test fixture allows you to phase lock frequency between DUT Transmitter Clock, Oscilloscope , and AWG/AFG sources. Converts DUT Transmitter Clock from 66.666MHz to 10MHz.
Cables	Six BNC cables/SMA cables with BNC to SMA connectors: one TCA-BNC or TCA-SMA.

See Also

Compatibility

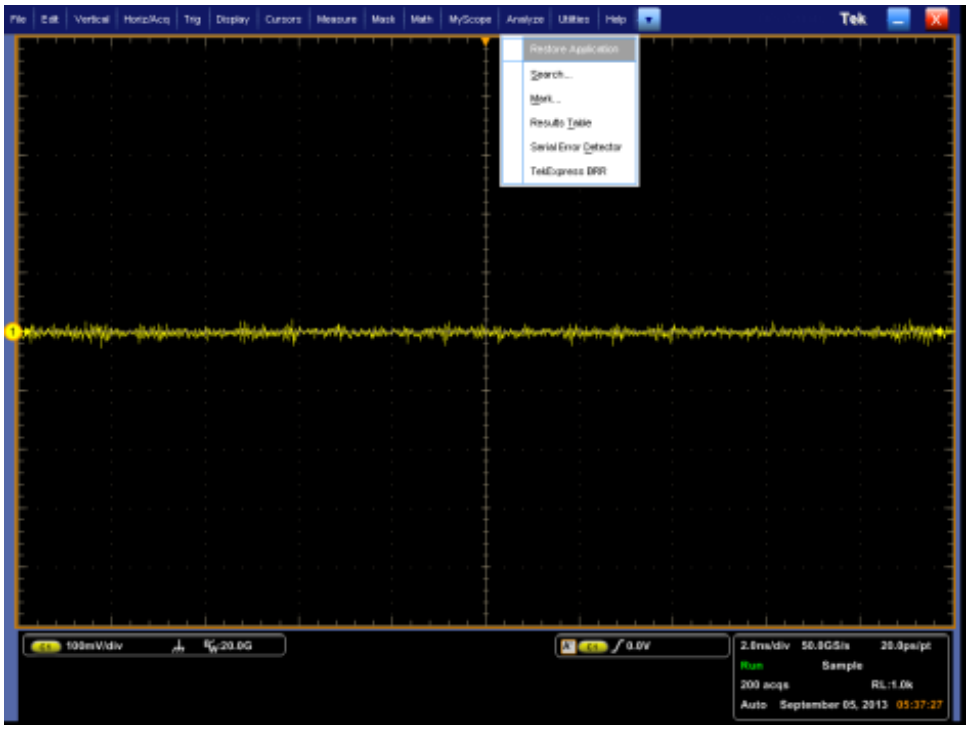
[2] If TekExpress is installed on a Tektronix Oscilloscope, the virtual GPIB port will be used by TekExpress for communicating with Oscilloscope applications. If external GPIB communication devices such as USB-GPIB-HS or equivalent are used for instrument connectivity, ensure that the Talker Listener utility is enabled in the DPO/DSA Oscilloscope's GPIB menu. For ease of use, connect to an external (secondary) monitor.

Install the Software

The software can be installed on any compatible instrument running Windows 7.

1. Close all applications (including the TekScope application).
2. Go to the www.tektronix.com Web site and search for BRR to locate the installation and download the latest installation file.
3. Double-click the executable file to extract the installation files. After extraction, the installer launches and displays the Install shield Wizard.

   Follow the prompts to install the TekExpress BRR application.
4. The software automatically installs in C:\Program Files\Tektronix \TekExpress\TekExpress BroadR-Reach
5. The installer updates the TekScope Analyze menu to include TekExpress BRR:



See Also

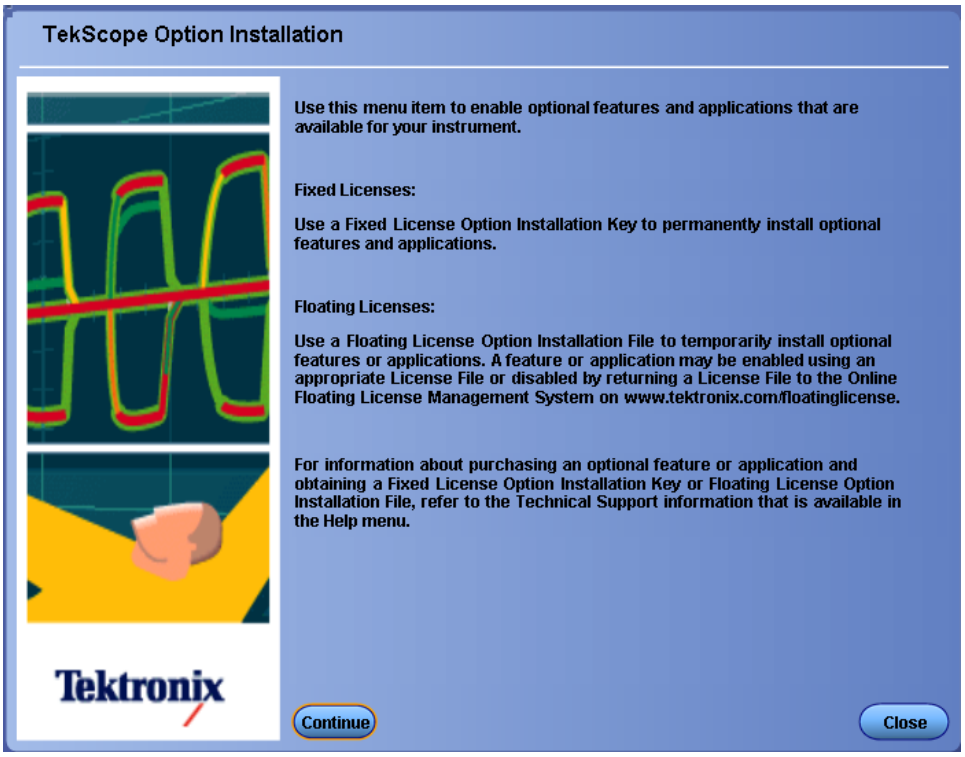
Minimum System Requirements

Activate the License

Activate the license using the option installation wizard on the Oscilloscope. Follow these steps to activate the TekExpress BRR license:

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

   The TekScope Option Installation wizard opens.



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

See Also

View Version and License Information

View Version and License Information

Use the following instructions to view application version information and license information for the application modules.

To view version information:

1. From the Options menu, select **About TekExpress**.
2. Click the **View Version Details** link to check the version numbers of the installed test suites. Close the dialog box when finished.

To view license information:

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

   The Options section in the dialog box displays a list of installed options, including TekExpress BRR.
2. To view the Option key, look in the Option Installation Key section. When finished, click **OK** to close the dialog box.

See Also *Activate the License*

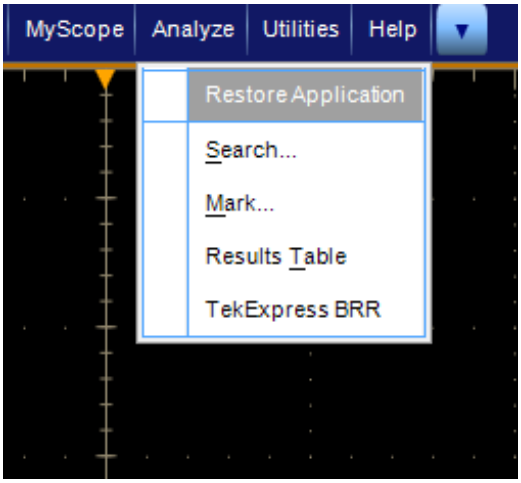
Options Menu

Start the Application

The first time you open the application after installation, you are required to enter a license key. The application also checks for a file called Resources.xml located in the X: drive. If this file is not found, instrument discovery is performed before launching TekExpress BRR. The Resources.xml file contains information regarding instruments available on your network.

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

To run the BRR application, from the TekScope menu, select **Analyze** > **TekExpress BRR**.



A initializing screen is displayed, before the launch of the application.

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

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

Options Menu

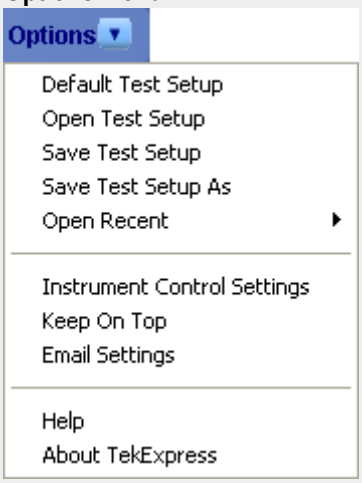
Options menu

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

It has the following selections:

Menu	Function
Default Test Setup	Opens an untitled test setup with defaults selected
Open Test Setup	Opens a saved test setup
Save Test Setup	Saves the current test setup selections
Save Test Setup As	Creates a new test setup based on an existing one
Open Recent	Displays a menu of recently opened test setups to select from
Instrument Control Settings	Shows the list of instruments connected to the test setup and allows you to locate and refresh connections to connected instruments
Keep On Top	Keeps the TekExpress BRR application on top of other open windows on the desktop
Email settings	Use to configure email options for test run and results notifications
Help	Displays TekExpress Help
About TekExpress	■ Displays application details such as software name, version number, and copyright ■ Provides access to *license information* for your BRR installation ■ Provides a link to the Tektronix Web site

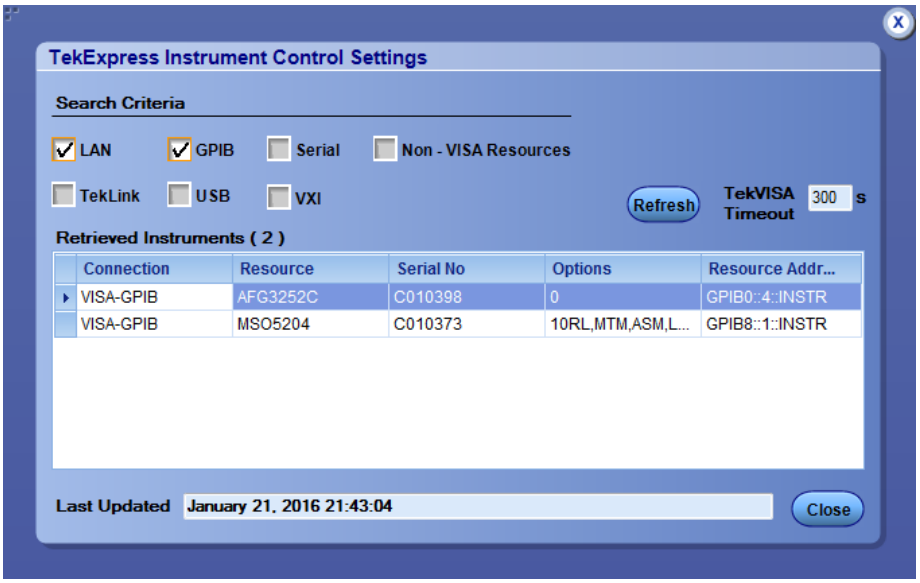
Options Menu



View Connected Instruments: the Instrument Control Settings

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

Instrument Control Settings



To refresh the list of connected instruments:

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

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

   NOTE. *Make sure GPIB option is enabled to detect the real time Oscilloscope in the Global Settings tab.*

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

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

Application Controls

Table 3: Application controls descriptions

Item	Description
Options menu	Opens the Options menu for access to global controls
Panels	Visual frames with sets of related options. Some panels are further divided into tabs and other sections.
Start button	Use the **Start** button to continuously acquire and accumulate measurements. If prior acquired measurements have not been cleared, the new measurements are added to the existing set.
Stop button	Use the **Stop** button to abort a test in progress.
Pause \ Continue button	Use the **Pause** button to interrupt the current acquisition. When a test is paused, the button name changes to Continue.

Item	Description
Clear button	 Available only on the *Results panel*. Use the **Clear** button to clear all existing measurement results. ***NOTE.*** *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.*
Application window move icon	 Place the cursor over the three-dot pattern in the upper left corner of the application window. When the cursor changes to a hand, drag the window to the desired location.

See Also

Options menu

Exit the Application

Use the following method to exit the application:

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

1. Click on the application title bar.
2. Do one of the following:
   - A message box appears asking if you really want to exit TekExpress. To exit, click **Yes.** Otherwise, click **No**.
   - If you have an unsaved session or test setup open, you will be asked if you want to save it before exiting. To save it, click **Yes.** If you do not wish to save, click **No**. To remain in the session, click **Cancel.**.

Application Basics

Application Basics

The TekExpress BRR software user interface is intuitive and easy to use. In addition to the UI, a *programmatic interface* is available.

The user interface has four main panels, which allows you do the following:

- *Set up the tests.*
- *Prepare to run the tests*.
- *Run the tests and view the progress of analysis*.
- *View the results of the tests*.
- *Configure and view reports*.
- *Save and recall test setups*.

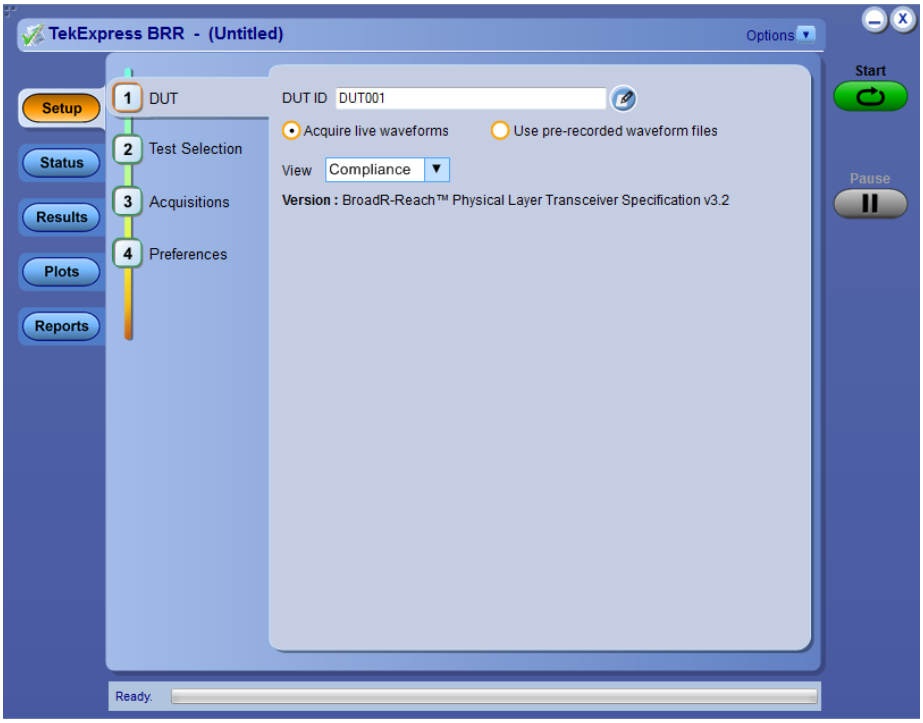
See Also *Overview and Key Specifications*

Setting Up Tests

Setting Up Tests: the Setup Panel

The Setup panel guides you through the BRR test setup process using tabs. The options selected in a tab affect the options available in the next tab down.

Setup panel



NOTE. *A check mark next to a tab title means you have made changes. However, if you make changes on a tab that significantly affect a prior tab, the check mark of the affected tab is replaced with its number, indicating that you should check the settings on that tab.*

- *DUT tab*

  Here, you can specify the DUT ID and the view settings. These settings apply to all tests for the current session, and affect the test list in the Test Selection tab.

- *Test Selection tab*

  Here, select tests individually or by group, view a short description of a selected test, and view a schematic showing appropriate device connections.

- *Acquisitions tab*

  View configurations for attached probes, view a *list of signals and sources*, and view the Acquisitions table.

- *Configuration tab* (Displays only when **View: Advanced** is selected in the DUT tab.)

  Here, select either **Compliance Mode** or **User Defined Mode**, view Global Settings, select measurement settings, and view or edit test parameter limits.

- Preferences tab

  Here, you may specify that an email be sent to you upon test failure.

See also

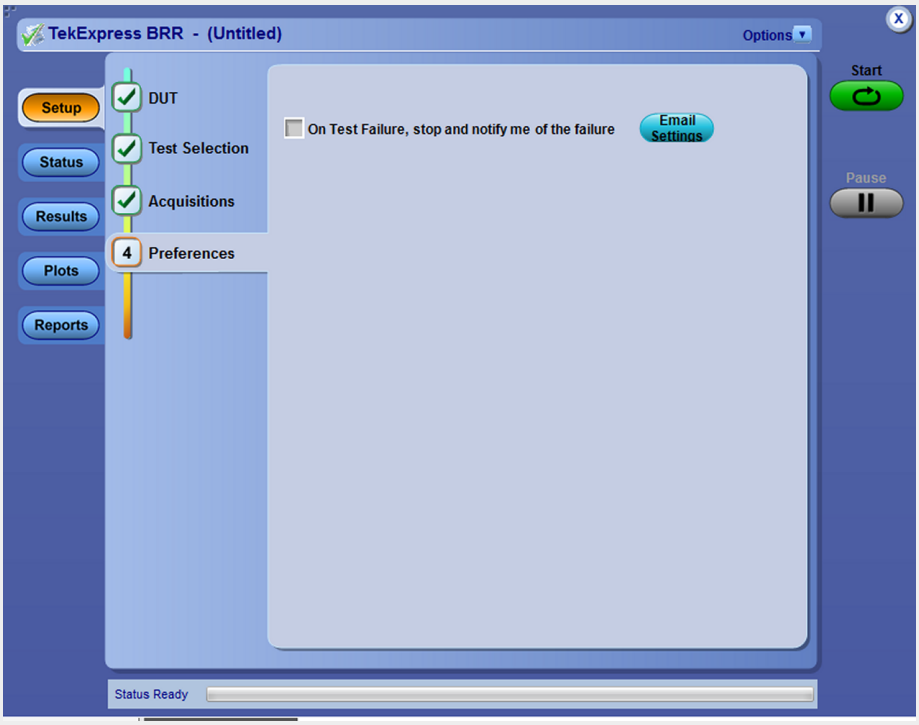
Saving a Test Setup

Running the Tests and Viewing their Progress in the Status Panel

Viewing Test Results in the Results Panel

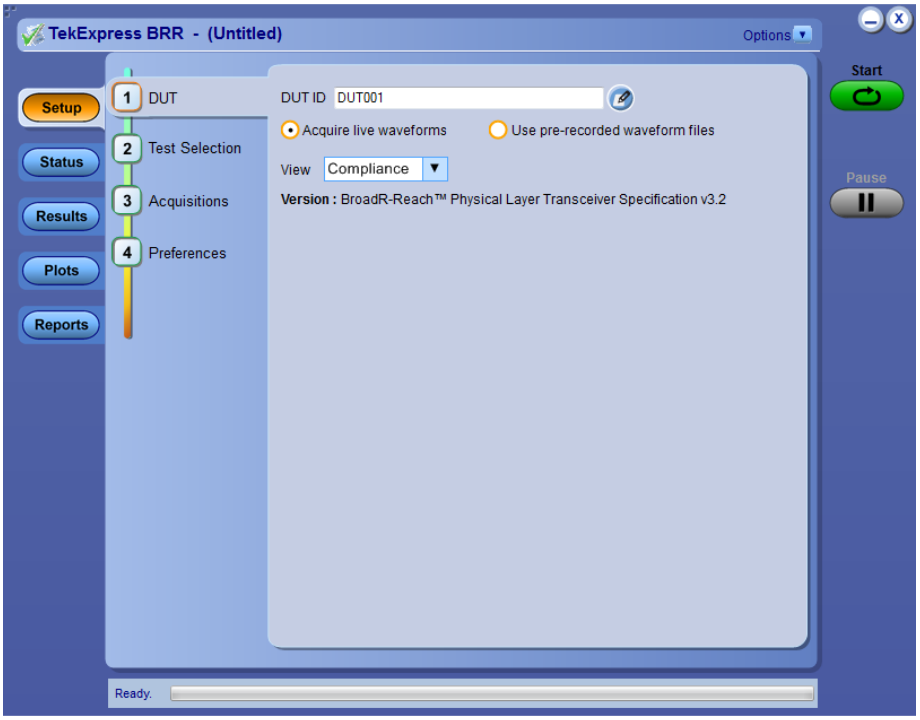
Configuring and Viewing Reports in the Reports Panel

Preferences tab



Selecting Device Parameters: the DUT tab

DUT tab



1. Click the DUT tab on the Setup panel and specify the DUT ID and the view settings.

   These settings apply to all current tests, and affect the test list in the Test Selection tab.

2. Enter the device **ID** (default value is DUT001). This ID appears on test reports. To add comments to the top of a report, click and enter up to 256 characters.

3. For the View option, select either **Compliance** or **Advanced**.

   - If you choose the **Advanced** view, an additional Configuration tab appears.
   - To access configuration options for **Compliance** view, click the **Configure** button on the Test Selection tab.

See Also *Setting Up Tests: the Setup Panel*

Choosing Tests: the Test Selection tab

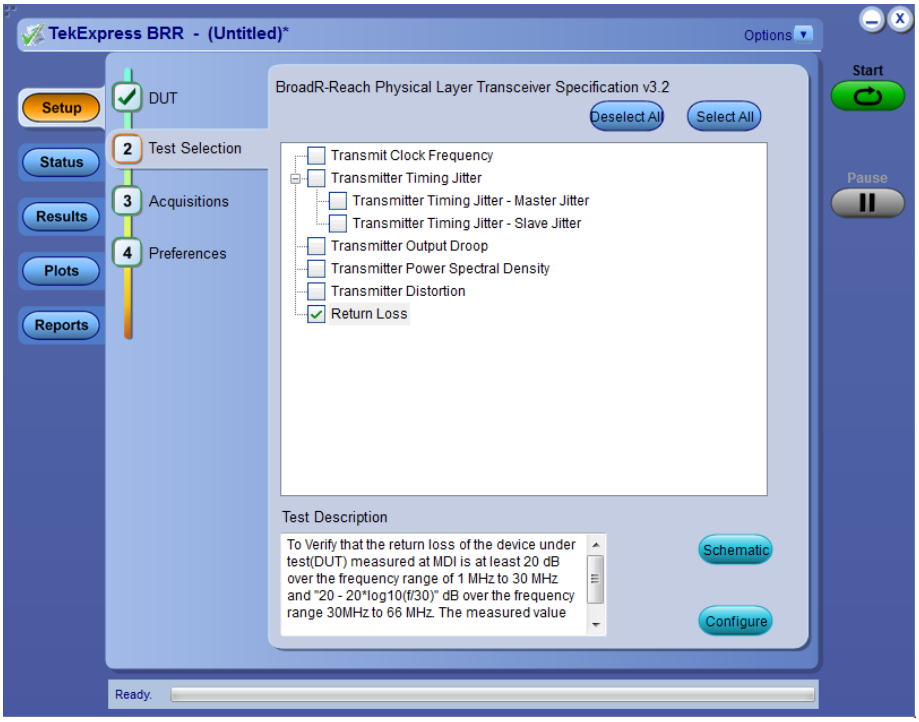
Acquiring Waveforms: the Acquisitions tab

Configuring Tests: the Configure button or Configuration tab

Choosing Tests: the Test Selection tab

Use these instructions to select the tests to run on the connected DUT. The tests that you select here impact the parameters available in the Acquisitions tab.

Test Selection tab



1. In the Setup panel, click the Test Selection tab.
2. Select the desired tests. For a list of supported tests, *click here*.
   - Click the check boxes of individual tests or of entire groups of tests. To select all tests in the list, click the **Select All** button and click **Deselect All** unless you want to run all tests.
   - Click on the **Schematic** button to display the schematic document for the selected test. Use to verify the test setup before running the test.

Once you have selected the tests, then you can *configure the tests*.

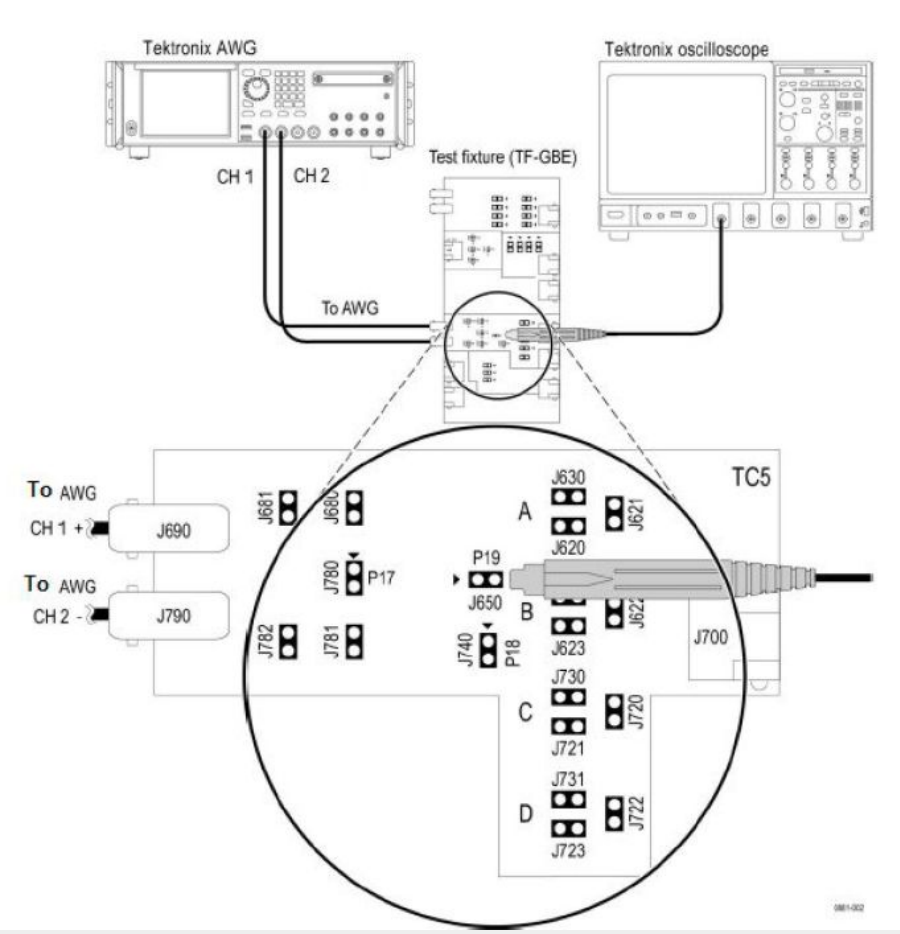
See Also *Setting Up Tests: the Setup Panel*

Acquiring Waveforms

Configuring Tests: the Configure button or Configuration tab

Running the Tests and Viewing their Progress: the Status Panel

Example of Schematic Document



Selecting Acquisitions

Acquiring Waveforms: the Acquisitions tab

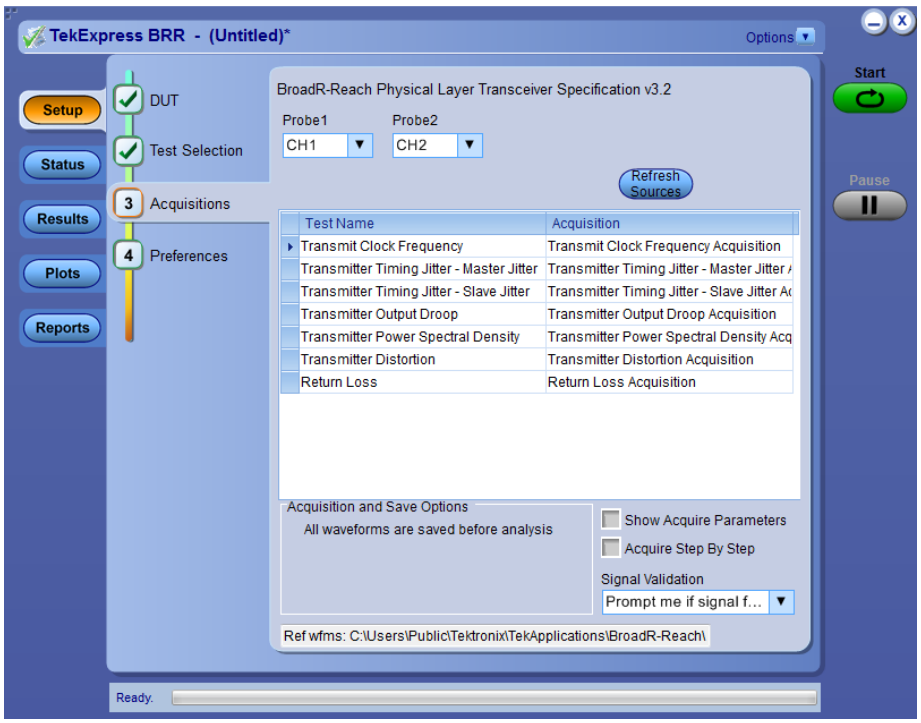
The Acquisitions tab in the Setup panel is used to view and set acquisitions parameters for the selected tests. Before you can do this, you must first:

- Choose either **Acquire Live Waveforms** or **Use Pre-recorded waveform files** on the DUT tab of the Setup panel.
- The acquisition parameters displayed on the Acquisitions tab will differ depending on your choice on the DUT tab.

If you have chosen **Acquire Live Waveforms** on the DUT tab, do the following:

1. In the Setup panel, click the Acquisitions tab.

   Acquisitions Tab with Acquire Live Waveforms selected



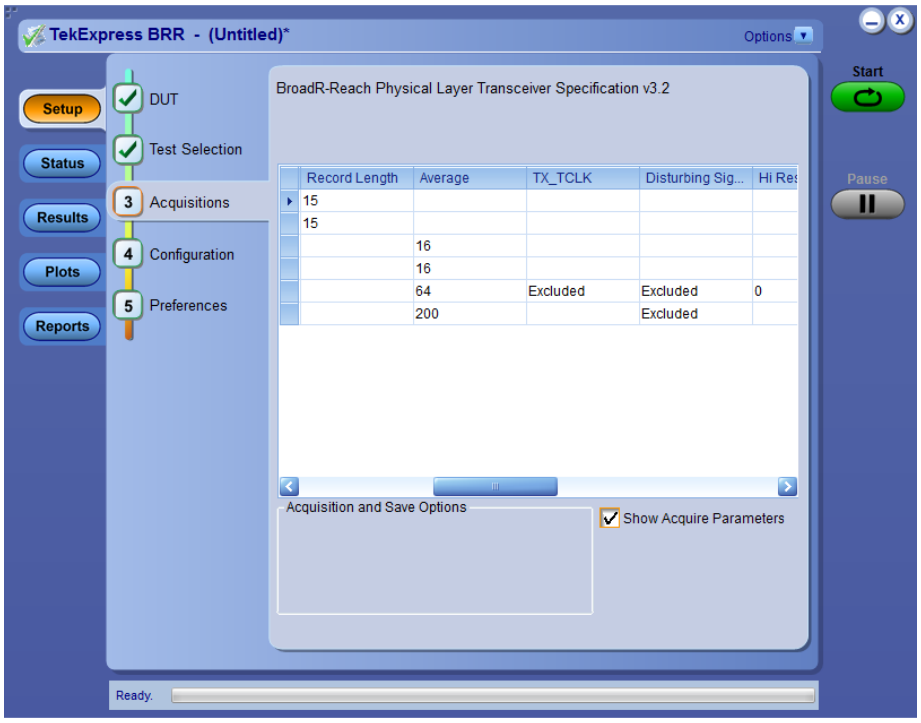
2. Select the probe source channel for each listed signal in the Probe selection drop-down menu.
3. You may select to view the probe configuration for each channel used.
4. You'll see the Acquisitions table. For information about the possible parameter columns displayed in the Acquisitions table, click here.

The table below lists the possible parameter columns displayed in the Acquisitions table in the Acquisitions tab of the Setup panel. Columns displayed depend on the tests selected in the Test Selection tab, and whether or not the **Show Acquire Parameters** check box has been selected.

Column name	Function
Test Name	Displays the name of the selected test for performing acquisitions. One or more tests can perform the same acquisitions.
Acquisition	Displays the named acquisition.
Record Length	Displays the size of the record in samples.
Disturbing signal	Displays whether or not a disturbing signal is excluded.
Average	Displays the number of averages that will be used with average mode acquisition.
AWG Series	Displays the AWG series model, if present.
TX_TCLK	Displays whether or not a TX_TCLK is excluded.
Hi resolution	Displays the HiRes(Vertical resolution) factor.
WaveForm FileName	Displays the name and location of the waveform file to be used for the measurement. Applies only to testing using prerecorded waveforms.

5. Underneath the table, you may check the **Show Acquire Parameters** check box. When selected, the acquisition parameters for each test displays an additional columns in the Acquisitions table.
6. You may check the **Acquire Step by Step** check box. When selected, the application prompts you to continue after each phase of test completes.
7. Select a Signal validation parameter (Signal validation is valid only for Live acquisitions):
   - **Prompt me if signal fails**: Select to prompt if signal fails.
   - **Use signal as is-Don't check**: Select to perform the test even if signal validation fails
   - **Skip test if signal fails**:Select to skip the test for which signal validation fails.
1. In the Setup panel, click the Acquisitions tab.

Acquisitions Tab with Use Pre-recorded Waveform Files selected



2. You'll see the Acquisitions table. Locate the row for the desired test and then click the ellipsis button ( ) in the Waveform FileName column. Select a file. You can select more than one file for each test. For information about the possible parameter columns displayed in the Acquisitions table, click here.
3. Underneath the Acquisitions table, you may check the **Show Acquire Parameters** check box. When selected, the acquisition parameters for each test are displayed in additional columns in the Acquisitions table.

Acquisitions Table

The table below lists the possible parameter columns displayed in the Acquisitions table in the Acquisitions tab of the Setup panel. Columns displayed depend on the tests selected in the Test Selection tab, and whether or not the **Show Acquire Parameters** check box has been selected.

Column name	Function
Test Name	Displays the name of the selected test for performing acquisitions. One or more tests can perform the same acquisitions.
Acquisition	Updates the location of the named acquisition.
Record Length	Displays the size of the record in samples.
Disturbing signal	Displays whether or not a disturbing signal is excluded.
Average	Displays the record average.
AWG Series	Displays the AWG series model, if present.
TX_TCLK	Displays whether or not a TX_TCLK is excluded.
Hi resolution	Displays the resolution.
WaveForm FileName	Displays the name and location of the waveform file to be used for the measurement. Applies only to testing using prerecorded waveforms.

See also. *Use Prerecorded Waveforms for Analysis*

Acquire Live Waveforms for Analysis

Setting Up Tests the Setup Panel

Acquire Live Waveforms for Analysis

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

1. *Open a saved test setup* or *create a new one*.
2. In the Setup panel, select the DUT tab, enter the desired DUT ID and then select **Acquire live waveforms** check box.
3. *Select other DUT options* as desired.
4. In the Test Selection tab, *select the desired test(s)*.
5. In the Acquisitions tab, select or view the desired parameters in the Acquisitions Table.
6. *Configure the tests* if you have not done so already. Then click **Start** to run the test.

See also. *Acquiring Waveforms the Acquisitions tab*

Use Prerecorded Waveforms for Analysis

Use Prerecorded Waveforms for Analysis

NOTE. *If you are using the prerecorded waveform files option, it is recommended that you use a waveform file (.wfm) that was captured from a Tektronix Oscilloscope. This eliminates the need to use an Oscilloscope. You can manually select waveforms and perform the tests by clicking the* ***Start*** *button. The reference waveforms are provided with the installer at C:\Users\Public\Tektronix\TekApplications\BroadR-Reach.*

1. *Open a saved test setup* or *create a new one*.
2. In the Setup panel, select the DUT tab, select the desired DUT and then select **Use pre-recorded waveform files** check box.
3. In the Test Selection tab, *select the desired tests*.
4. In the Acquisitions tab, locate the row for the desired test in the Acquisitions Table and then In the **Waveform FileName** column for that test, click the ellipses button ( ) and select the desired waveform file.
5. Perform any additional desired test setup, such as *test configuration*, and then click **Start** to run the test.

You can select a different waveform file for each test, deselect a test from the current run, or add a test to the current run. Once you click **Start**, data corresponding to that test run populates the other panels (such as the test status and test results).

See also. *Acquiring Waveforms the Acquisitions tab*

Acquire Live Waveforms for Analysis

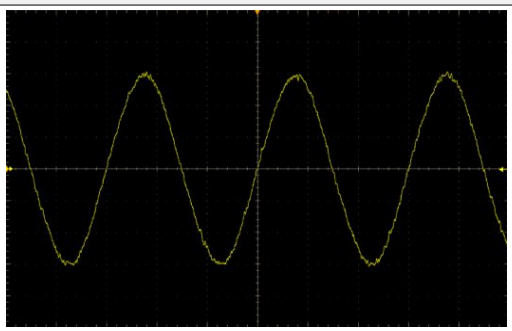
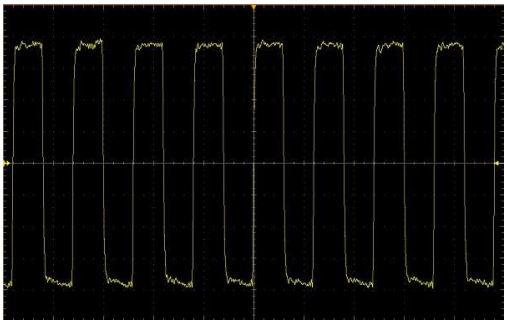
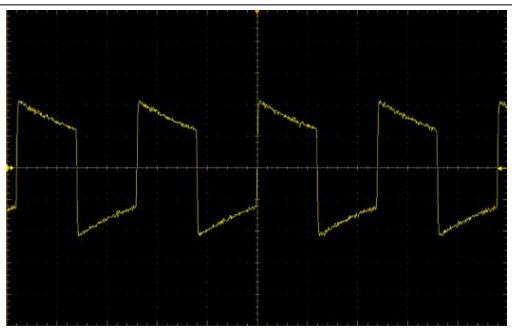
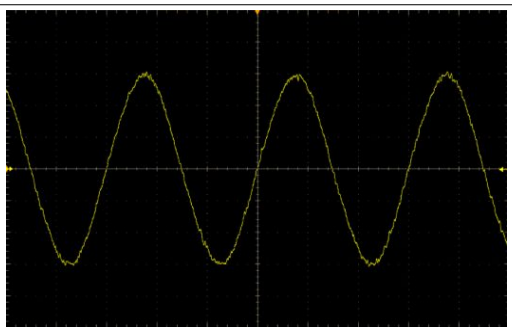
Acquire Step By Step

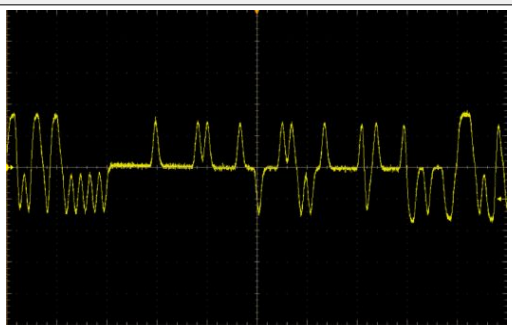
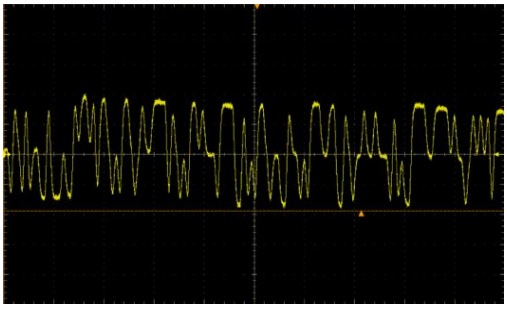
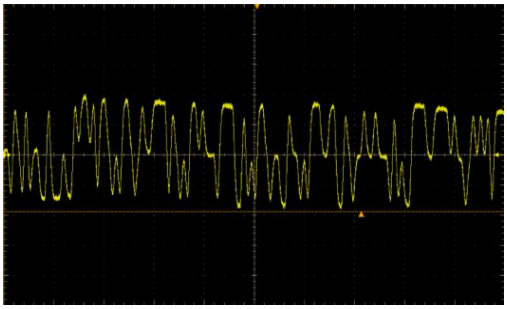
The **Acquire Step By Step** option is available in the Acquisitions panel. This is a global parameter that is applied to all tests when selected. By default, this option is deselected.

When selected, this parameter allows for display of the reference input waveform of the selected measurement. This helps to compare the input waveform coming from DUT with the typical reference waveform (snap shot), allowing you to change the setup before acquiring the waveforms. The following table gives different reference waveform snap shots that appear for different tests.

NOTE. *When using prerecorded waveform files, the* ***Acquire Step By Step*** *option is not available.*

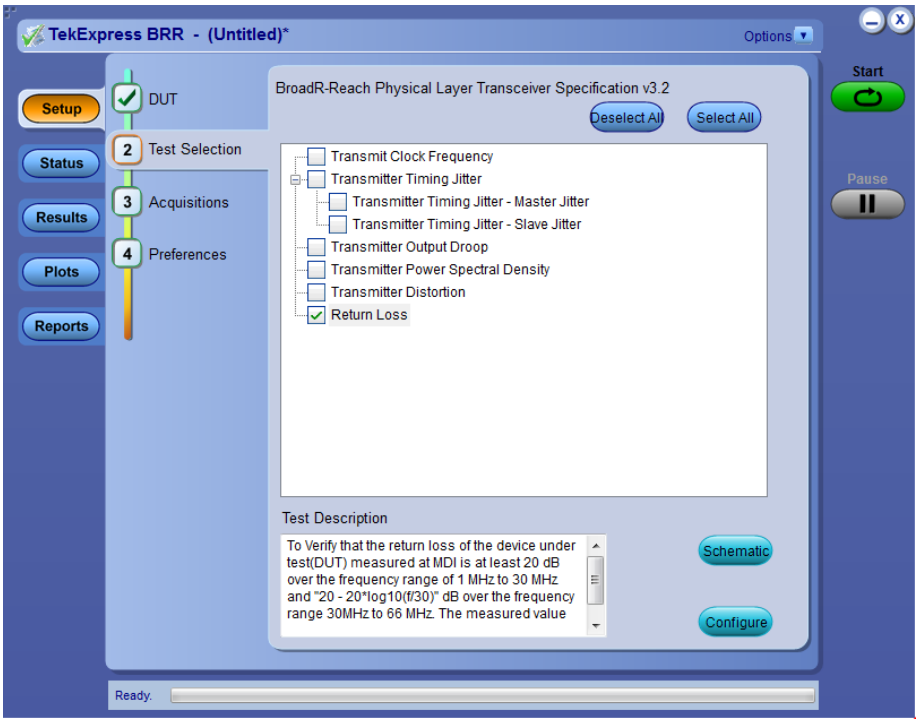
Table 4: Reference waveforms

Test name	Reference waveform
Transmit Timing Jitter – Master Jitter	
Transmit Timing Jitter – Slave Jitter	
Transmitter Output Droop	
Transmit Clock Frequency	

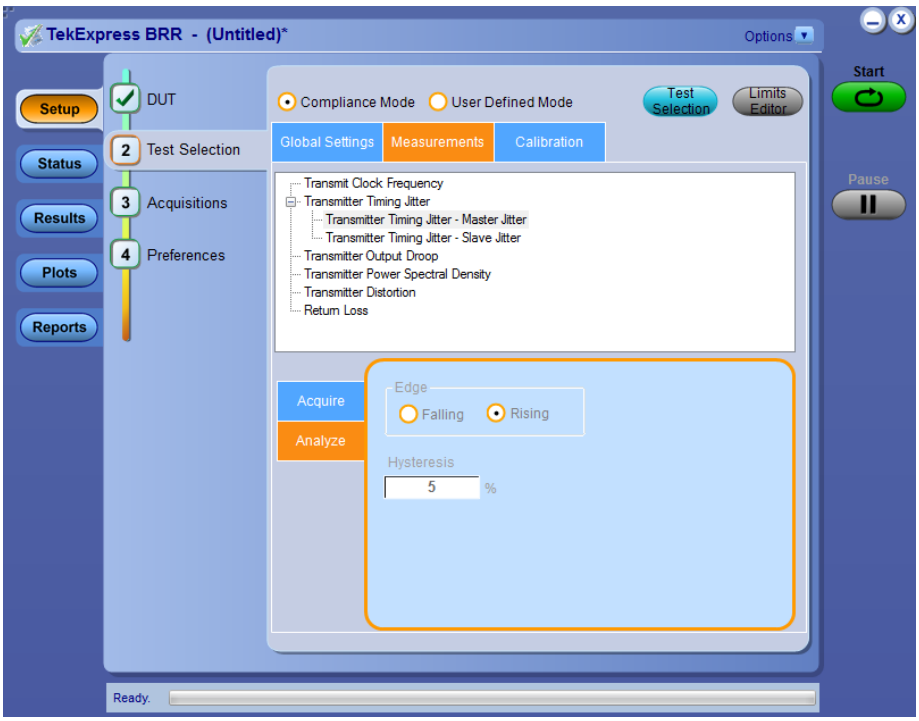
Test name	Reference waveform
Transmitter Distortion	
Transmitter Power Spectral Density	
Return Loss	

Configuring Tests: the Configuration Screen

Configure button on Test Selection tab



Configuration tab



Once you've selected tests, you may use the Configuration screen. There are two ways to reach the screen: the Configuration tab or the Configure button.

The BRR solution you are using determines how you may reach the Configuration screen.

- If you selected **Compliance** View in the Setup panel DUT tab, click on the Test Selection tab and select the desired test. Then click on the **Configure** button in the lower right corner of the Test Selection tab.
- If you selected **Advanced** View in the Setup panel DUT tab, the Configuration tab will appear on the Setup panel – click on it.

The Configuration screen shows **Global** parameters, which are common for all tests, and **Measurement** parameters, which are specific to selected tests, including acquisition, analysis, and limit parameters.

NOTE. *Test parameters that are grayed cannot be changed.*

Settings

- *Compliance Mode or User Defined Mode*
- Global Settings tab

In the **Instruments Detected** section, click in the shaded area opposite **Real Time Scope** and select the desired instrument from the drop-down list. If you don't see the instrument, refresh the list by using the *Instrument Control Settings*.

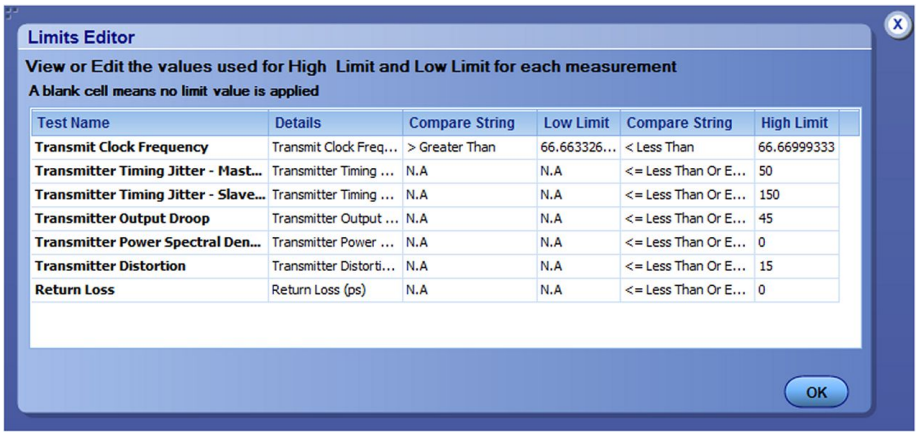
- Measurements tab

Lists all tests for the selected measurement type. Click on a measurement to view the available parameters in the tabbed filed below the list. The parameters and parameter type tabs shown depend on the selected test. To edit the test parameters, select the User Defined Mode check box.

- You may change test parameters. Select the test whose parameters you'd like to change. The options below the test list change to reflect the selected test.

  To modify the parameters, select the desired tab and parameters. For parameter details, see *Measurement Parameter Descriptions*.

Limits Editor



Test Name	Details	Compare String	Low Limit	Compare String	High Limit
Transmit Clock Frequency	Transmit Clock Freq...	> Greater Than	66.663326...	< Less Than	66.66999333
Transmitter Timing Jitter - Mast...	Transmitter Timing ...	N.A	N.A	<= Less Than Or E...	50
Transmitter Timing Jitter - Slave...	Transmitter Timing ...	N.A	N.A	<= Less Than Or E...	150
Transmitter Output Droop	Transmitter Output ...	N.A	N.A	<= Less Than Or E...	45
Transmitter Power Spectral Den...	Transmitter Power ...	N.A	N.A	<= Less Than Or E...	0
Transmitter Distortion	Transmitter Distorti...	N.A	N.A	<= Less Than Or E...	15
Return Loss	Return Loss (ps)	N.A	N.A	<= Less Than Or E...	0

- Use the **Limits Editor** button in the upper right corner to view or change the High Limit and Low Limit values used for each measurement.

Next, *prepare to run the tests.*

See Also

Setting Up Tests: the Setup Panel

Selecting Device Parameters: the DUT tab

Choosing Tests: the Test Selection tab

Acquiring Waveforms: the Acquisitions tab

About Saving and Recalling Test Setups

Compliance Mode or User Defined Mode

From the Configuration screen, you will have the option to select either Compliance Mode or User Defined Mode.

- **Compliance Mode**: Select to use Compliance Mode values. You cannot change most test parameters in Compliance Mode, but you can view the compliance parameters.
- **User Defined Mode**: Select to run tests using custom parameters. You may change parameters that are not grayed out.

NOTE. *These modes are not to be confused with the two views available from the DUT tab:**Compliance View** and **Advanced View**.*

Pairing Modes and Views for Test Operation

Your selections of Modes and Views will change the way the tests operate. These pairings of views and modes are possible:

- **Compliance View** selected with **Compliance Mode**: Tests will run automatically with little or no user intervention. You will not be able to change test parameters to anything that deviates from the compliance standards. To view configuration options, click on the Test Selection tab of the Setup panel and click the Configure button.
- **Compliance View** selected with **User Defined Mode**: Tests will run automatically but you will be able to change test parameters before starting the tests. To view configuration options, click on the Test Selection tab of the Setup panel and click the Configure button.
- **Advanced View** selected with **Compliance Mode**: Tests will run automatically with little or no user intervention. You will not be able to change test parameters to anything that deviates from the compliance standards. To view configuration options, click on the Configuration tab of the Setup panel.
- **Advanced View** selected with **User Defined Mode**: Tests will run automatically but you will be able to change test parameters before starting the tests. To view configuration options, click on the Configuration tab of the Setup panel.

Measurement Parameter Descriptions

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

Measurements tab

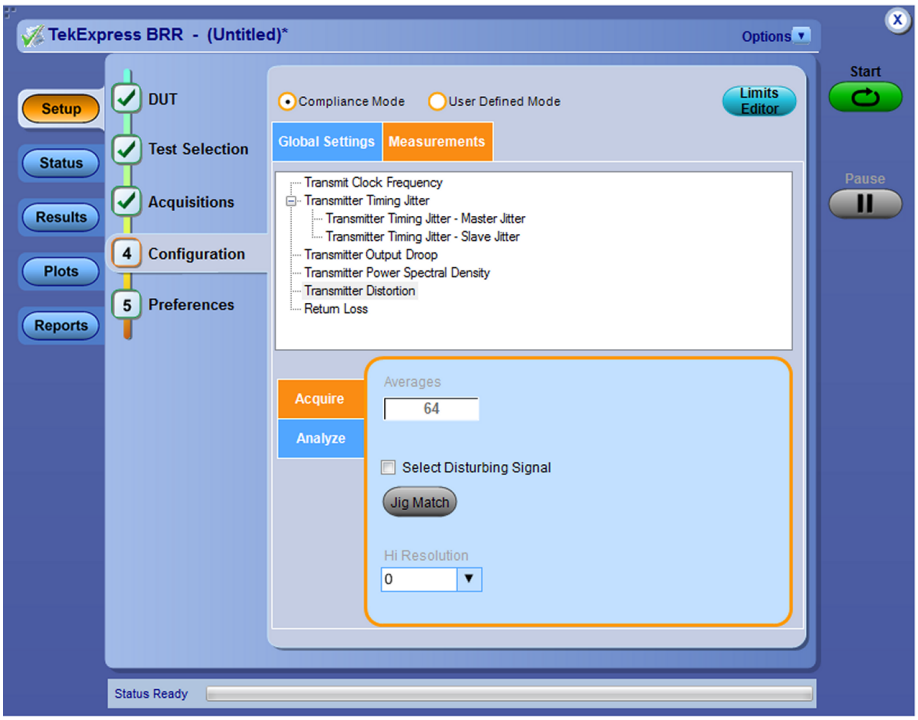


Table 5: Measurement parameters

Type	Name	Unit	Range/ Allowable values	Description	Applies to
Acquire parameters	Record Length	M	1 to 20	Sets the record length to use.	Transmit Clock Frequency Transmitter Timing Jitter (Master Jitter) Transmitter Timing Jitter (Slave Jitter)
	Averages	NA	2 to 100 2 to 256 2 t0 512	Sets the number of averages (number of acquisitions) for average mode acquisition.	Transmitter Output Droop Transmitter Power Spectral Density Return Loss
	Use Clock divider unit	NA	NA	Sets the Oscilloscope in the used source (AFG/ AWG) to the external 10MHz reference clock.	Transmitter Distortion

Type	Name	Unit	Range/ Allowable values	Description	Applies to
Analyze Parameters	Edge ■ Falling ■ Rising			Used to select the type of edges on which RMS jitter will be calculated .	Transmitter Timing Jitter (Master Jitter) Transmitter Timing Jitter (Slave Jitter)
	Hysteresis	%	1 to 10	Sets the hysteresis in percentage that gets used during edge finding	Transmitter Timing Jitter (Master Jitter) Transmitter Timing Jitter (Slave Jitter)
	RBW	KHz	1 to 100	Sets the resolution bandwidth. This controls the bandwidth of the spectral analyzer filters.	Transmitter Power Spectral Density
	Center Frequency	MHz	1 to 500	Sets the center of the frequency span over which spectral analysis is done.	Transmitter Power Spectral Density
	Frequency Span	MHz	1 to 500	Sets the range of frequencies over which spectral analysis is done	Transmitter Power Spectral Density
	Smooth	NA	0 to 10	Sets the number of samples that will be used while smoothing the return loss waveform; sets the averaging filter length.	Return Loss

See Also *Configuring Tests: the Configure button or Configuration tab*

Calibration

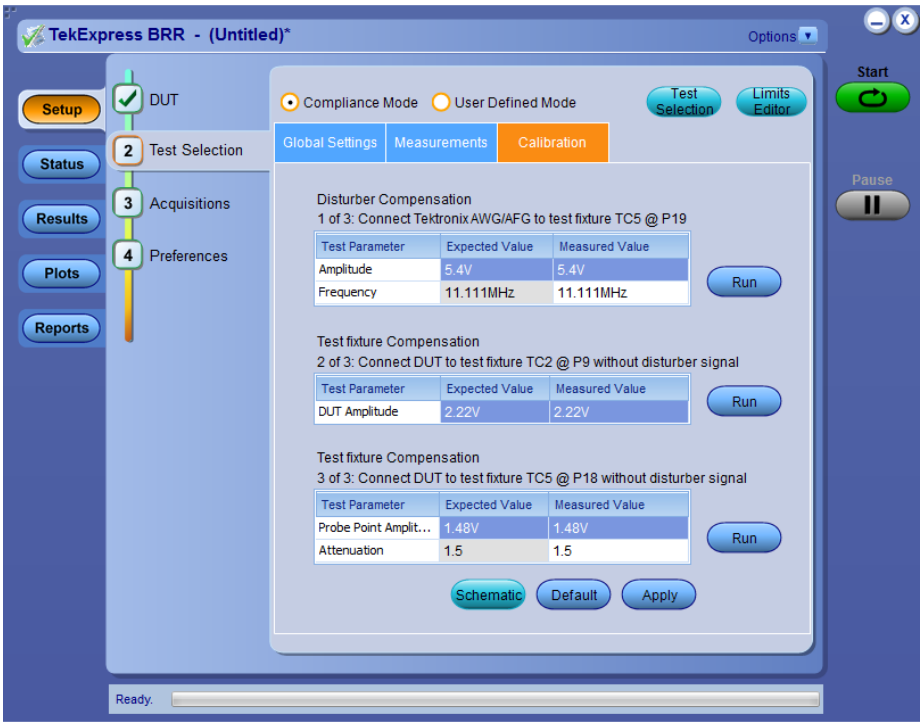
Calibration can be performed on live waveforms or by using the already-calibrated waveforms.

Measurements

Following are the measurements which can be calibrated:

Transmitter Distortion

In Transmitter distortion, calibration is done to effectively remove the disturbing signal and compensate for non-linearity in the disturber and test fixture.



Return Loss

- **Live calibration**

NOTE. *The live Calibration files are saved in* `X:\BRR\Calibration` *folder.*

- **Use prerecorded files**

Table 6: Calibration table parameters

Parameter	Description
Live Calibration	Sets the live calibration process.
Use Prerecorded Calibration	Sets the calibration process with prerecorded calibrated waveforms and allows you to browse and select the calibrated waveforms. The path for selecting the calibrated waveforms is C:\Users\Public\Tektronix\TekApplications\BroadR-Reach.
Channel drop down	Allows you to select relevant channel and probes. ***NOTE.*** *when you change the input sources (Channel) other than calibrated sources, you need to re-calibrate with latest sources.*
Cal Type	Displays the type of calibration: Load, Open, and Short.
Cal Status	Displays the status of the calibration: Pending, Done.
Cal Time	Displays the previous calibration time: Date, Month, and Year .
Schematic button	Click to view the schematic.
Plot button	Click to view the plot.
Apply button	Click to apply the configured parameters to calibration.
Run button	Click to run the process of calibration.
Default button	Click to perform calibration by using default values. ***NOTE.*** *This button is applicable only for Transmitter distortion. The expected and measured values for Transmitter distortion on Calibration tab will be initially empty, when you launch the application for the first time and they get populated once you click on the* ***Default*** *button .*
To manually set up signal source: Click here and copy files to USB drive and recall on connected signal source.	In case of manual signal source (AFG/AWG) setup, click the link and copy the relevant folder and recall the setup on connected signal source.

Running Tests

Before You Click Start

Before Running a Test for the First time

Before you run a test for the first time, review these steps:

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

After you install and launch TekExpress BroadR-Reach, it creates the following folders on the Oscilloscope:

- \Program Files\Tektronix\TekExpress\TekExpress BroadR-Reach
- \My Documents\My TekExpress\BRR
- \My Documents\My TekExpress\BRR\Untitled Session

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

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

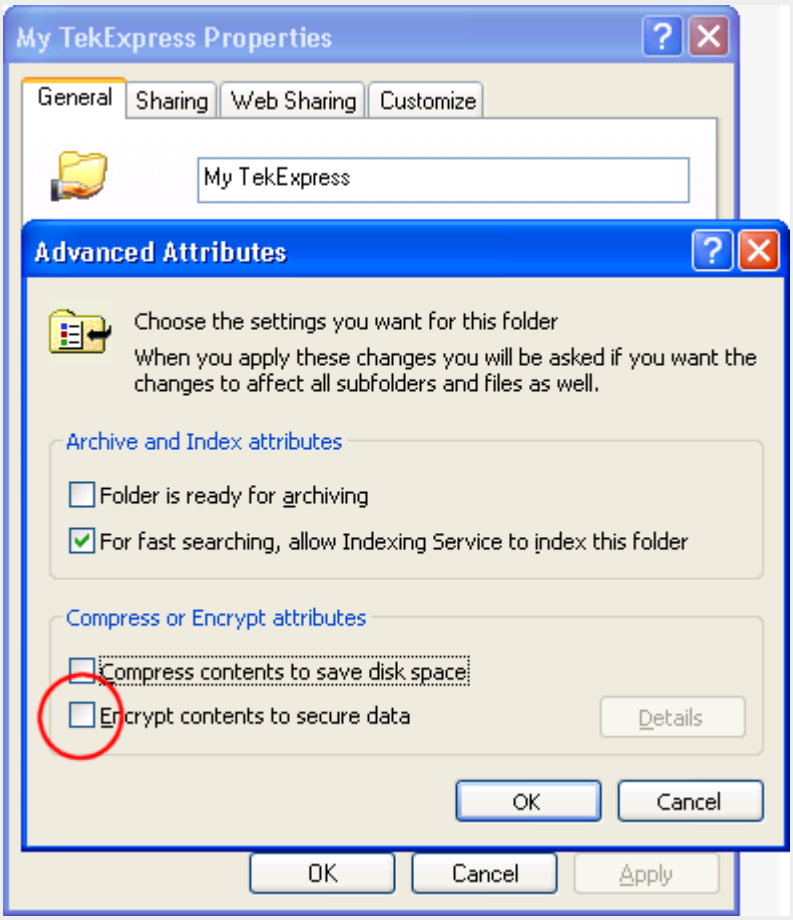
2. Ensure that the **My TekExpress** folder has read and write access, and that the contents are not set to be encrypted:

a. Right-click the folder and select Properties.

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

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

Example:



Before running any test

1. Review the *Pre Run Check List*.
2. Configure the *Email notification options* if you want the application to notify you by email when a test completes or produces an error. Access the email options either from the Options menu in the upper right corner, or from the Preferences tab on the Setup panel.
3. Select the *Report Options*.

See Also

PreRun Check List

Pre-Run Check List

Do the following before you click **Start** to run a test. If this is the first time you are running a test on a setup, then refer also to the guidelines above.

1. Ensure that all the required instruments are properly warmed up (about 20 minutes).
2. Perform the Signal Path Compensation (SPC).

   a. On the Oscilloscope main menu, select the Utilities menu.

   b. Select Instrument Calibration.

3. Deskew any cables.
4. Ensure that the application is able to find the DUT. If it cannot, *perform a search for connected instruments*.

   To find the DUT:

   a. Launch the TekExpress BRR application.

   b. Select the Setup panel and then click the DUT tab.

   c. Click the Test Selection tab. Select any test, and then click **Configure** button.

   d. In the Configuration section, click Global Settings.

   e. In the Instruments Detected section, click in the shaded area opposite Real Time Scope and make sure that the Oscilloscope with the (GPIB8::1::INSTR) designation is in the drop-down list.

See also

Before You Click Start

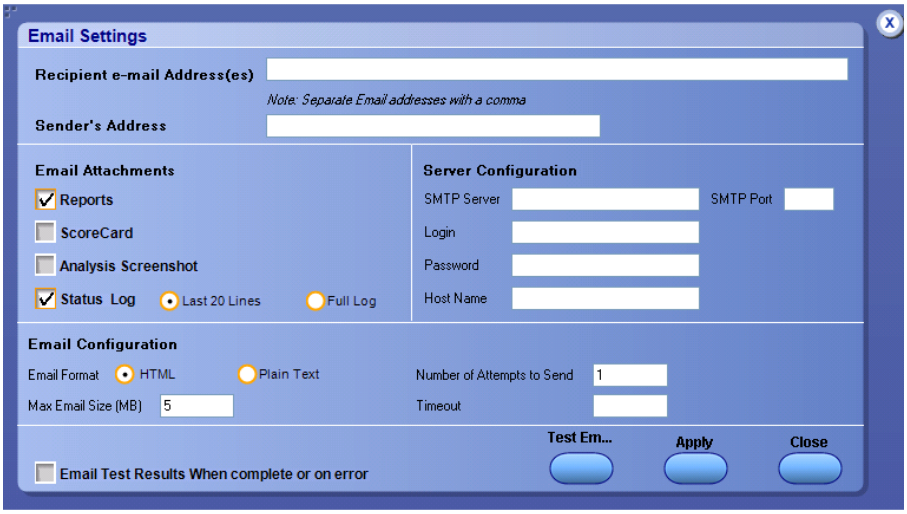
Configure Email Notification

Running the Tests and Viewing their Progress

Configure Email Notification

Set up these email settings if you want the application to notify you by email when a test completes or produces an error. Configure email from the Options menu.

1. From the Options menu in the upper right corner, select Email Settings to open the Email Settings dialog box, or click the Preferences tab on the Setup panel.



2. (Required) For **Recipient email Address(es)**, enter your email address. You can include multiple addresses as long as you separate the addresses with commas.

3. (Required) For **Sender's Address**, enter the email address used by the instrument. This address consists of the instrument name, followed by an underscore, followed by the instrument serial number, the @ symbol, and the email server used. For example: DPO72004C_B130099@yourcompany.com.

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

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

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

---

5. In the **Email Attachments** section, select from the following options:

   - **Reports**: Select to receive the test report with the notification email.

- **Status Log**: Select to receive the test status log with the notification email. If you select this option, then also select whether you want to receive the full log or just the last 20 lines.

NOTE. *The ScoreCard and Analysis Screenshot options are not available in BRR.*

---

6. In the **Email Configuration** section, select as desired:
   - Select the message file format to send: HTML (the default) or plain text.
   - Enter a maximum file size for the email message. Messages with attachments larger than this limit will not be sent. The default is 5 MB.
   - To limit the number of attempts the system makes to send a notification, enter the number in the **Number of Attempts to Send** field. The default is 1. You can also specify a timeout.
7. Select the **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** button.
10. Click **Close** button to exit the Email Settings dialog box.

See also *Before You Click Start*

PreRun Check List

Running the Tests and Viewing their Progress

Running the Tests and Viewing their Progress: the Status Panel

Once you've configured the tests and gone through the *Pre-run Check List*, from any screen, click the green **Start** button. The application acquires and analyzes the data, then displays a report when the tests are completed.

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

Viewing the Progress of Analysis

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

Test Status view

The Test Status tab presents a collapsible table with information about each test as it is running. To collapse and expand the table rows, click the expand (+)/(−) collapse button.

Table 7: Status tab table

Column	Description
Test Name	Name of the test
Acquisition	Describes the type of data being acquired
Acquire Status	Progress of the acquisition: ■ To be started ■ Started Acquisition ■ Completed Acquisition
Analysis Status	Progress state of the analysis: ■ To be started ■ In Progress ■ Completed ■ Aborted ■ Skipped

Log View

The Log View tab provides a list of actions that happen as the test executes. You can use this tab to review or troubleshoot tests.

Table 8: Log View options

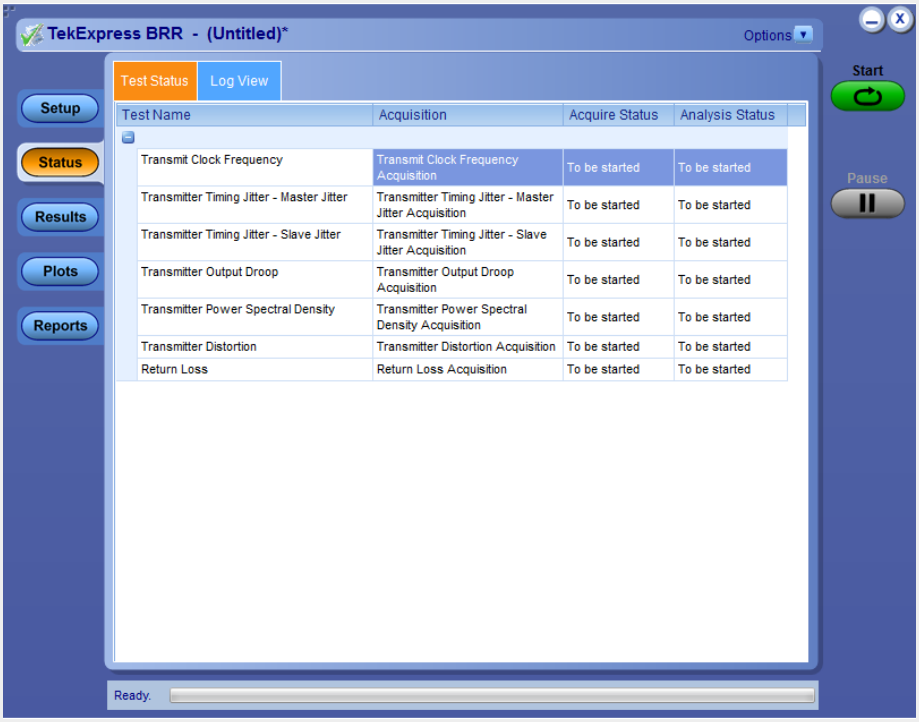
Item	Description
Message History	This window time-stamps and displays all run messages.
Auto Scroll	Select this check box to have the program automatically scroll down as information is added to the log during test execution.
Clear Log	Click this button to clear all messages in the Message History area.
Save	Use to save the log file as a text file for examination. Displays a standard Save File window and saves the status messages file that you specify.

See also *Before You Click Start*

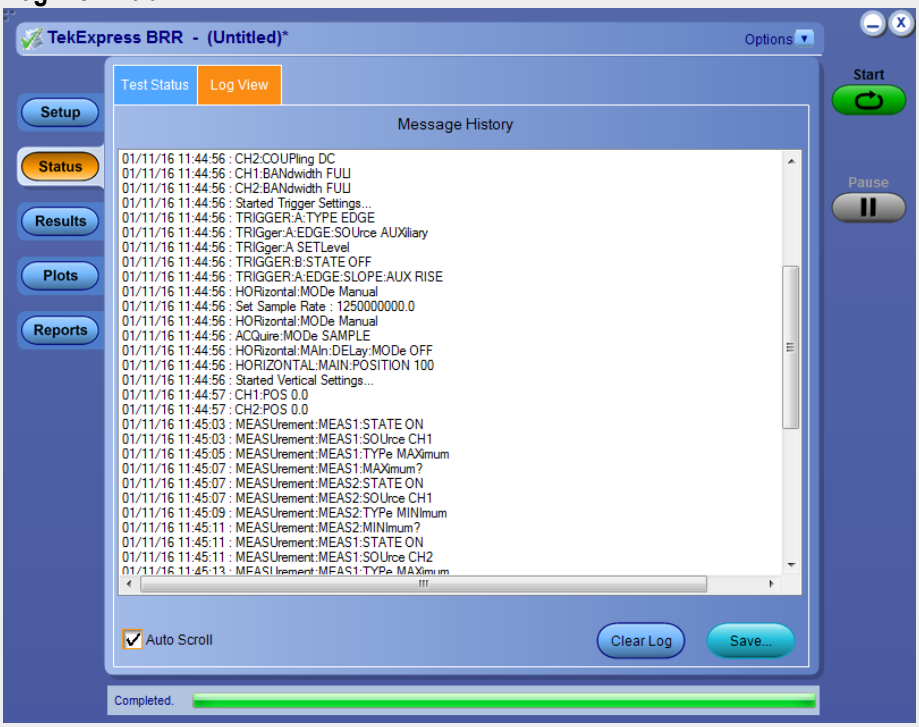
PreRun Check List

Configure Email Notification

Test Status Tab



Log View Tab



Viewing Plots

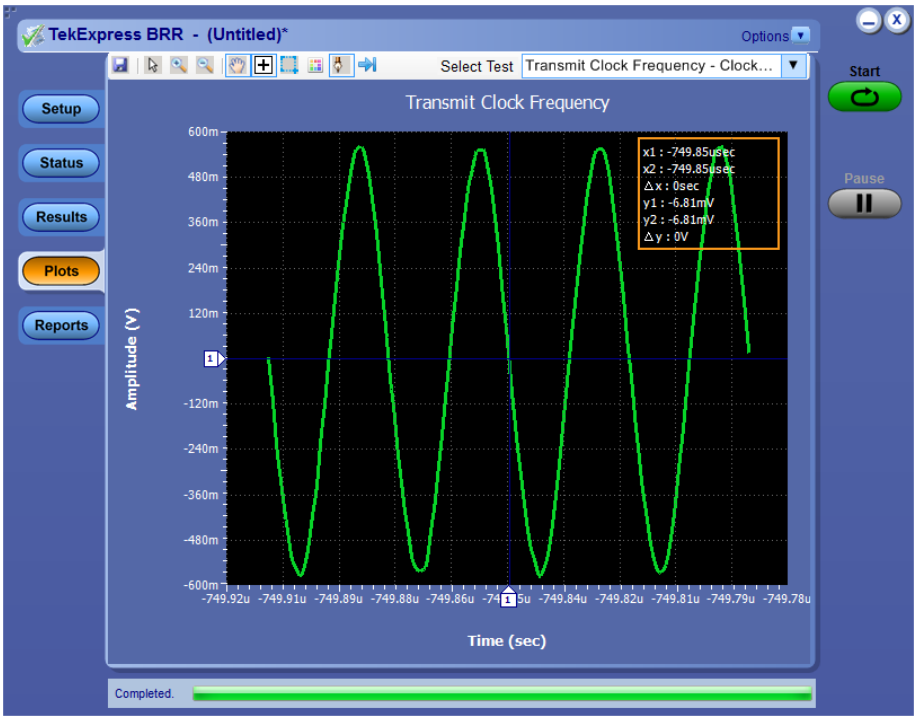
Viewing Plots

The Plots panel displays a summary of plot generated during run. The plots have zoom, cursors, save, dock/undock and select test features.

Types of Plots

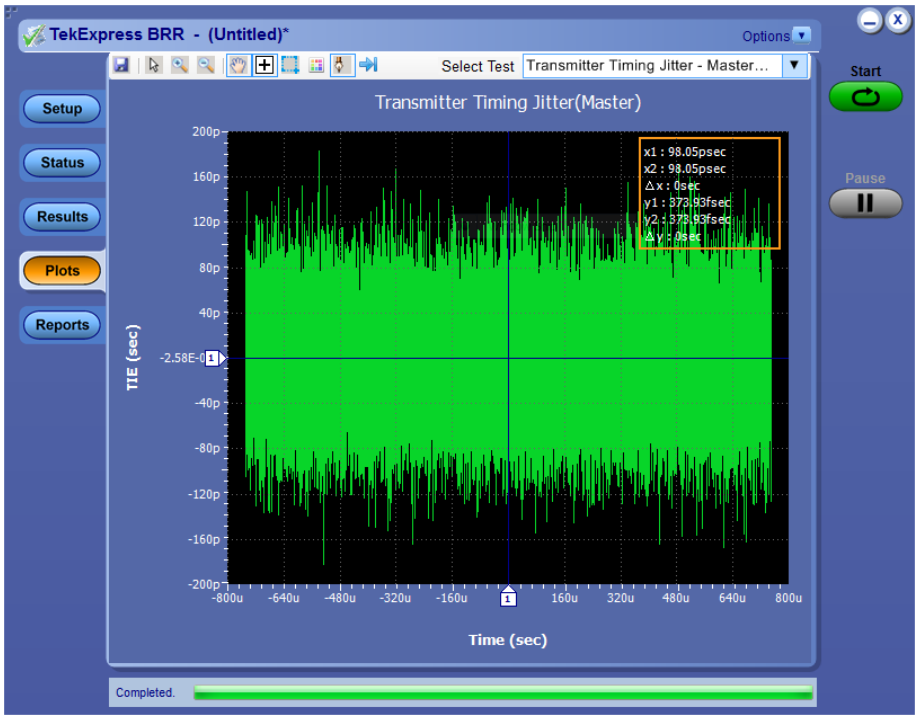
Following are different plots generated during run, based on the measurement(s) selected.

- **Transmit Clock Frequency**

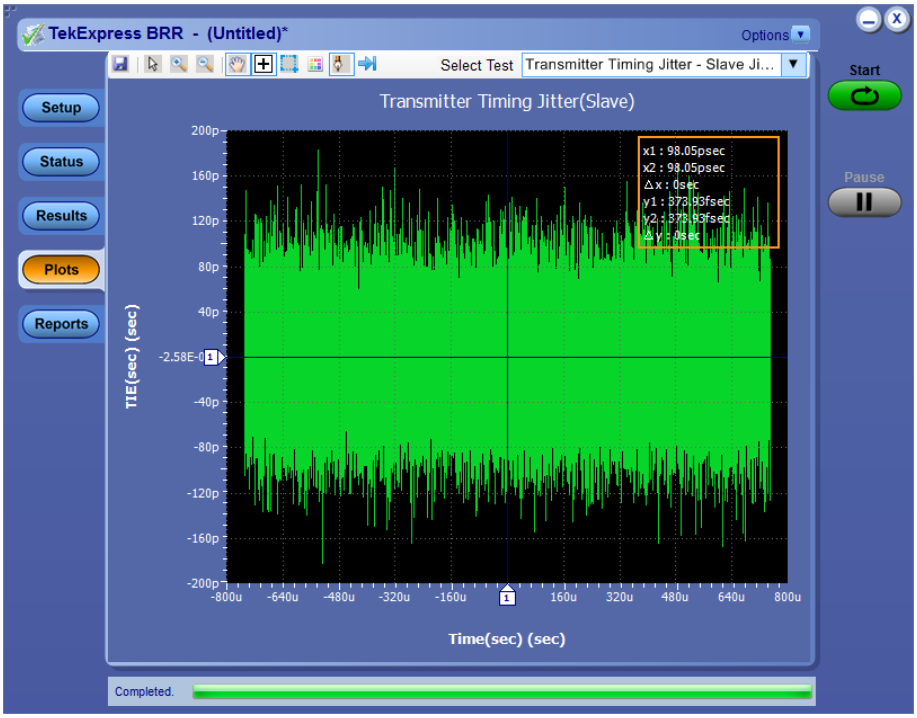


- **Transmitter Timing Jitter**

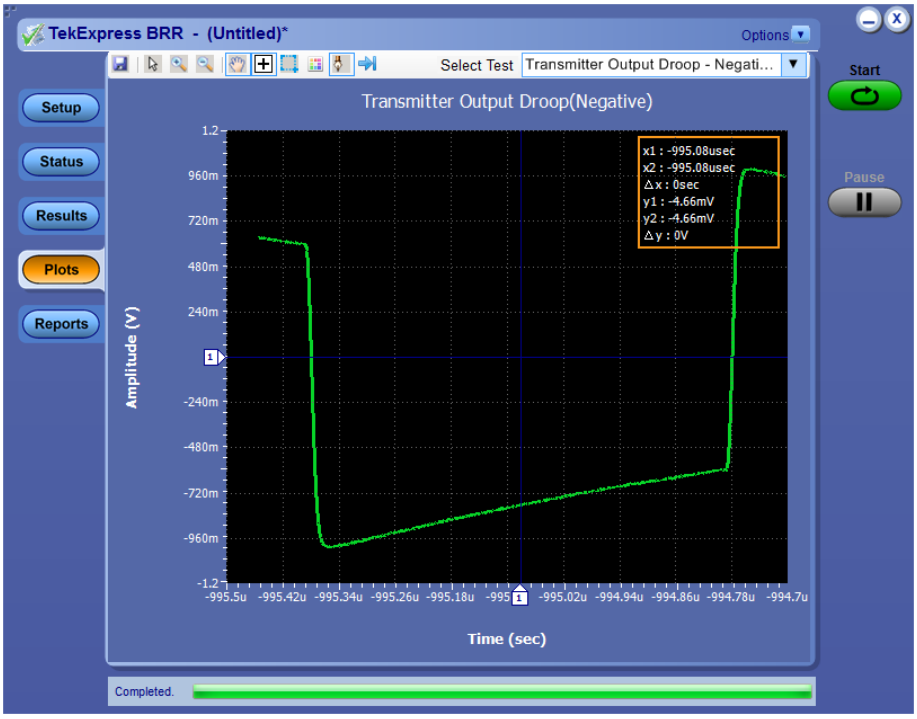
  Master Jitter



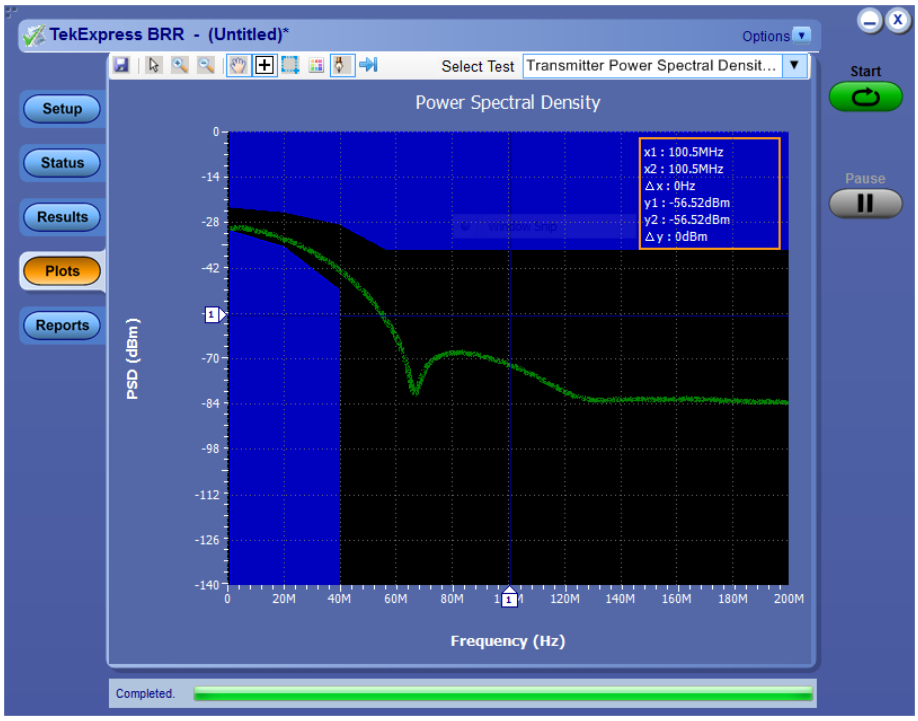
Slave Jitter



- **Transmitter Output Droop**

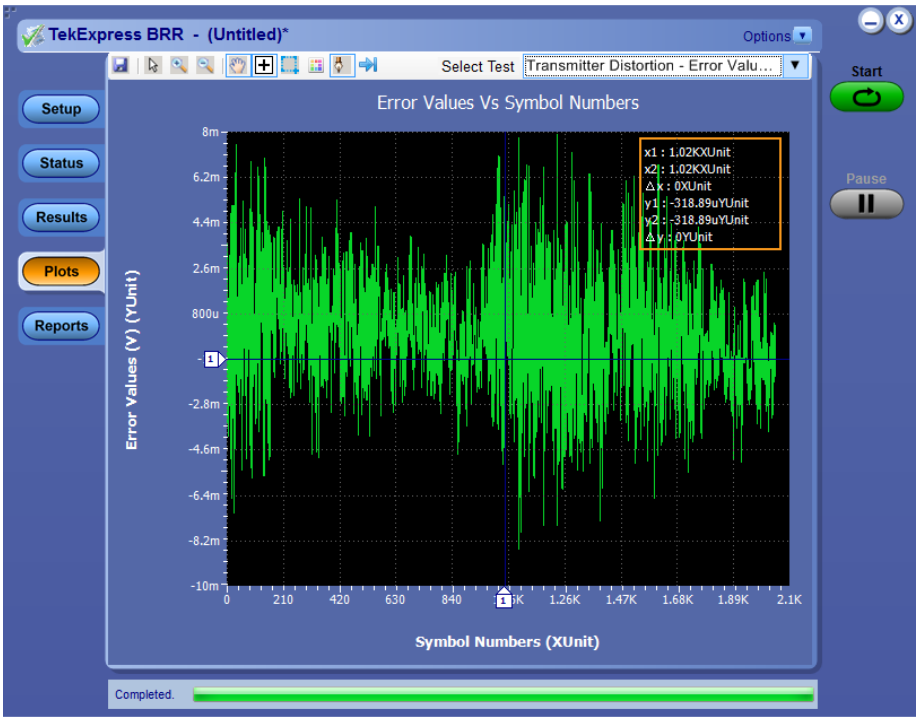


- **Transmitter Power Spectral Density**

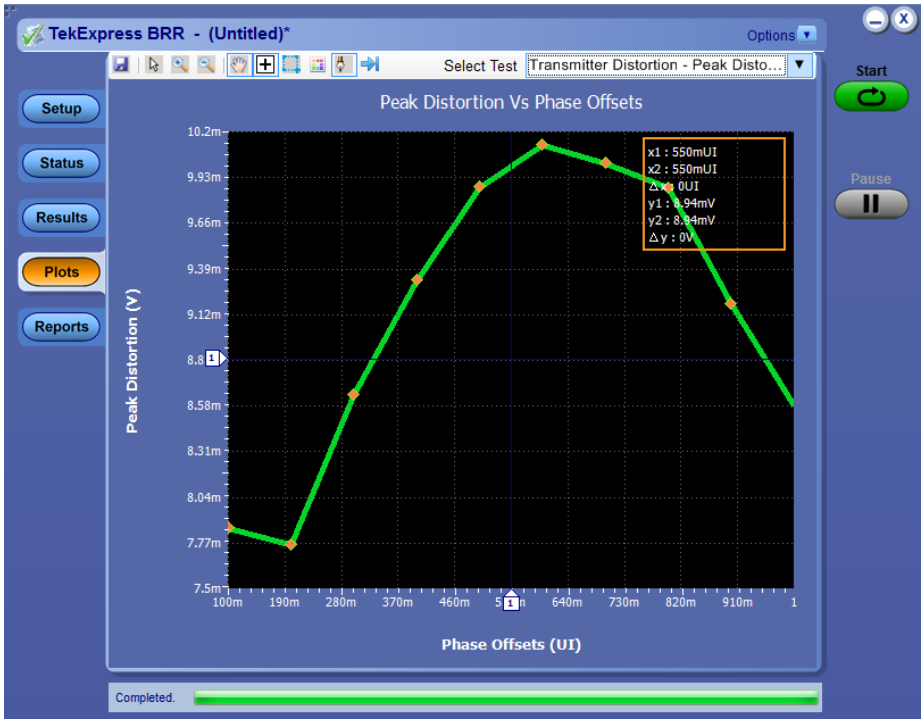


- **Transmitter Distortion**

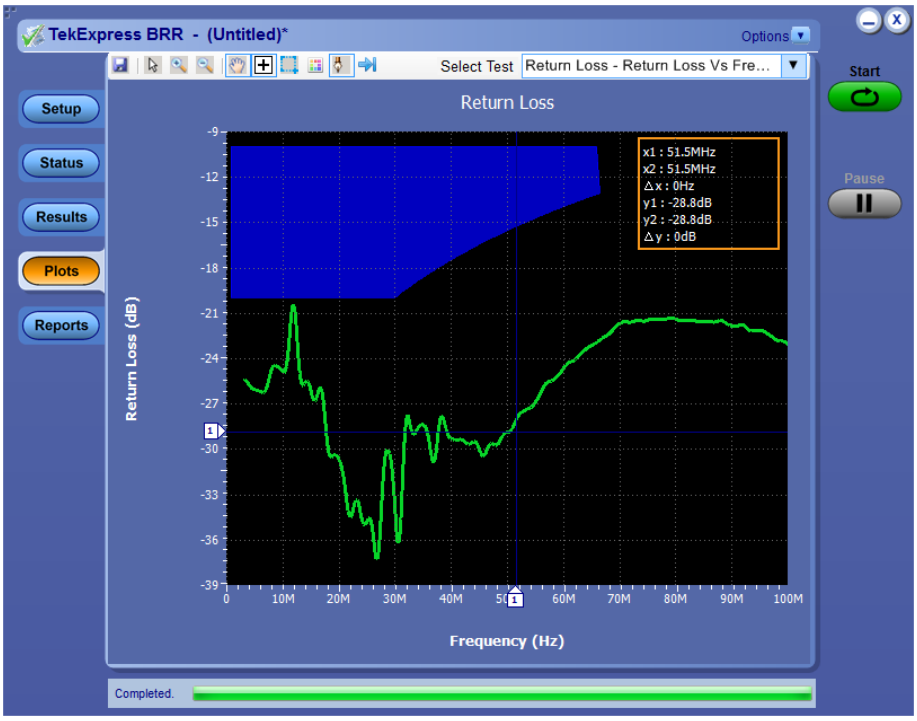
  Error value



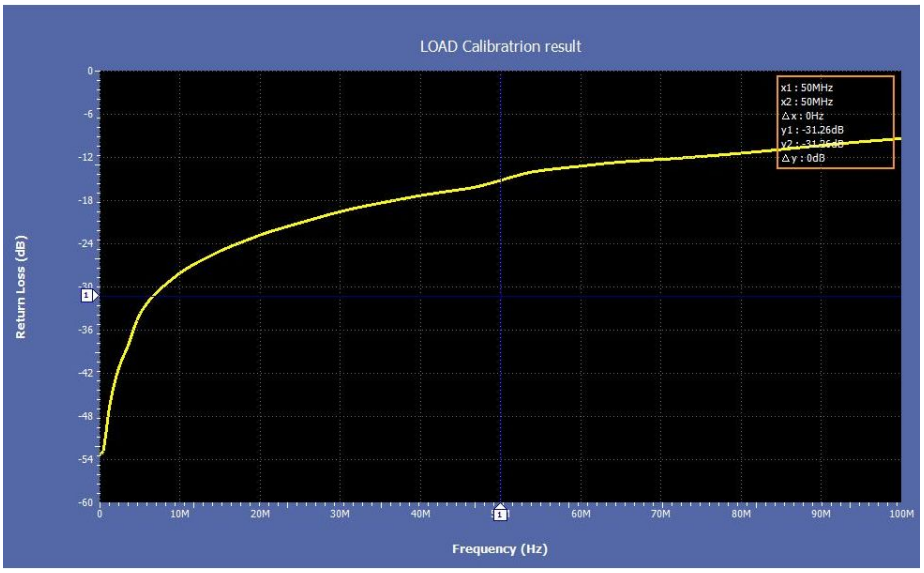
Peak Distortion


TekExpress BRR - (Untitled)*
Options
Select Test
Transmitter Distortion - Peak Disto...
Start
Pause
Setup
Status
Results
Plots
Reports
Peak Distortion Vs Phase Offsets
Peak Distortion (V)
10.2m
9.93m
9.66m
9.39m
9.12m
8.8
8.58m
8.31m
8.04m
7.77m
7.5m
100m
190m
280m
370m
460m
640m
730m
820m
910m
1
Phase Offsets (UI)
x1 : 550mUI
x2 : 550mUI
y1 : 8.94mV
y2 : 8.94mV
Completed.

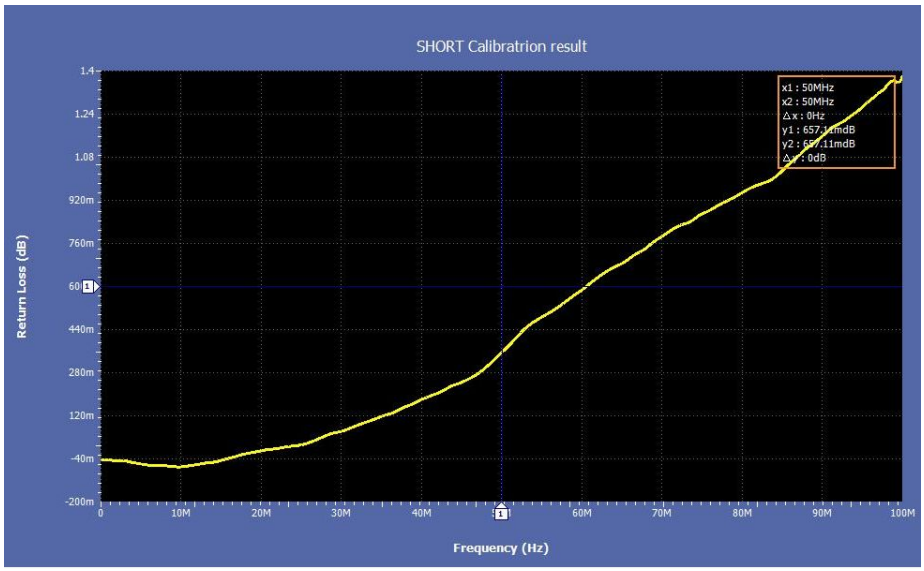
- **Return Loss**



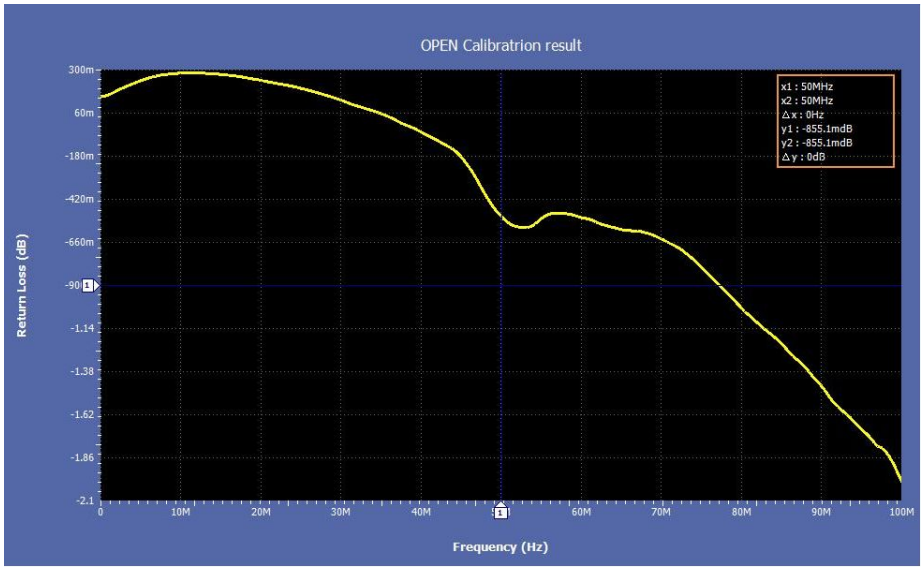
Return Loss Load



Return Loss Short



Return Loss Open



See also *Known Limitations*

Viewing Results

Viewing Test Results: the Results Panel

When a test completes running, the application switches to the Results Panel.

Set result viewing preferences from the Preferences menu in the upper right corner.

Each test result occupies a row in the Results table. By default, results are displayed in summary format, with the measurement details collapsed, and with the Pass/Fail column visible. You can change the display view.

- To expand all test rows listed, from the **Preferences** menu in the upper right corner, select **View Results Details**.
- To expand a collapsed test row, click the plus button ( ) to the left of the test row.
- To collapse all expanded test rows, select **Preferences > View Results Summary**.
- To collapse a single expanded test row, click the minus button ( ) to the left of the test row.
- To remove or restore the Pass/Fail column, select **Preferences > Show Pass/Fail**.
- To enable or disable the wordwrap feature, select **Preferences > Enable Wordwrap**.
- To expand the width of a column, place the cursor over the vertical line that separates the column from the one to the right. When the cursor changes to a double-ended arrow, hold down the mouse button and drag the column to the desired width.
- To sort the test information by column, click the column head. When sorted in ascending order, a small up arrow is displayed. When sorted in descending order, a small down arrow is displayed.
- To clear all test results displayed, click **Clear**

button.

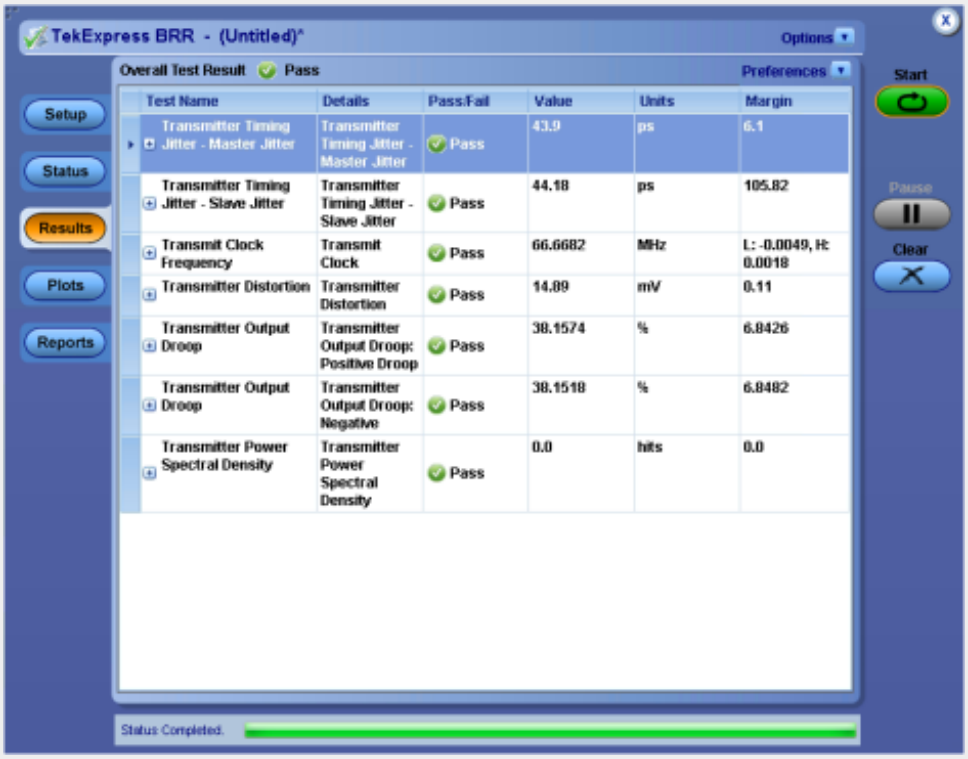
See Also

Setting Up Tests the Setup Panel

Configuring and Viewing Reports: the Reports Panel

Running the Tests and Viewing their Progress: the Status Panel

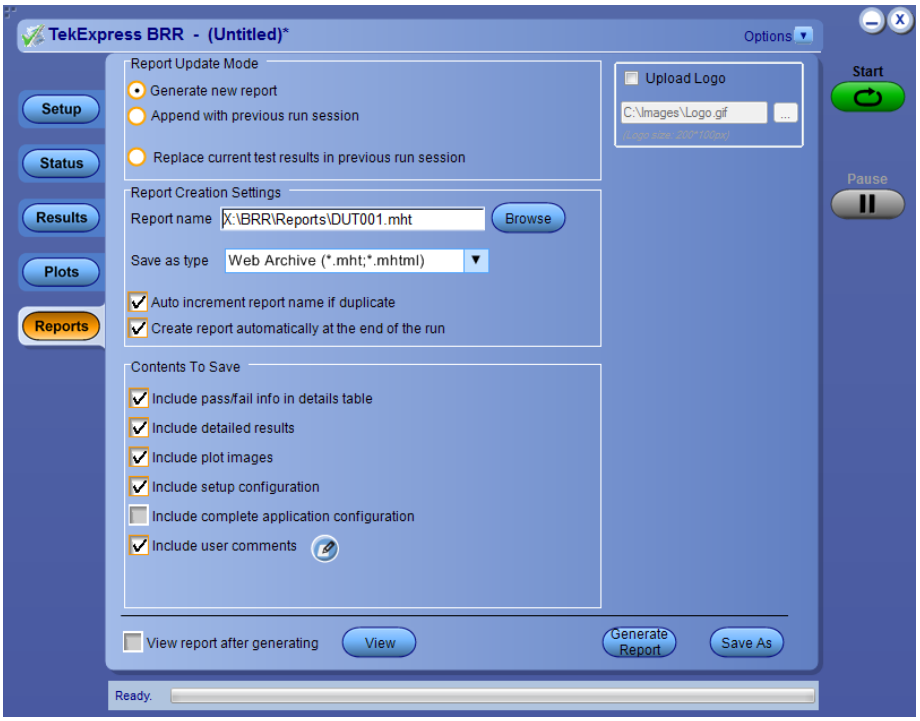
Results Preferences



Configuring and Viewing Reports

Configuring and Viewing Reports: the Reports Panel

Use the Report panel to browse for, name, and save reports, select report viewing options, and to view reports.



Naming a Report

Use the Reports Panel to select the naming convention to use for the report. By default, the test report file is located in the session folder to which it belongs, and gets overwritten each time you run the test under the same device name if you don't change the test report name before running the test.

If you do not want your test results to be overwritten each time you run any test, always give the report a unique name, or select to have the name increment each time you run a particular test. Generally, you would select report options before running a test, or when creating and saving test setups. Report settings are included in saved test setups.

Selecting Report Options

- Report Name

Displays the default name and location where the report will be saved when generated.

To change the report name or location, type over the current folder path and name and then save the test setup. Be sure to include the entire folder path, the file name, and the file type. For example: C:\Documents and Settings\your user name \My Documents\My TekExpress\BRR\DUT001_group1.mht.

To open an existing report, click **Browse** button, locate the report file and then click **View** button.

NOTE. *You cannot change the file location using the Browse button.*

- Save As Type

Saves a report in a file type different from the default. Lists supported file types to choose from (Web Archive (*mht,*mhtm), PDF (*pdf)) .

NOTE. *If you select a file type different from the default, be sure to change the report file name extension in the Report Name field to match.*

- Auto increment report name if duplicate

If the application finds a report with the same name as the one being generated, the application automatically increments the name of the report. For example: DUT001, DUT002, DUT003

- Include User Comments

Select to include any comments about the test that you or another user added in the DUT tab of the Setup panel. Comments appear in the Comments section under the summary box at the beginning of each report.

- Append Reports

This option adds new report data to the end of an existing report of the same name. This option is deselected by default.

- View Report After Generating

Automatically opens the report in your Web browser when the test completes. This option is selected by default.

NOTE. *If you unchecked this option before running a test but now would like to view the report, then when analysis is complete, click the* ***Browse*** *button at the top of the Reports panel and navigate to the report file.*

Viewing a Report

You can view any report by locating and opening the report file, which ends in .mht unless you changed the file type before running the report.

NOTE. *The application automatically zooms the test report on the scope along with waveform details, after completion of test.*

The top of the report displays information about the instruments and probes used, the duration of the test, software versions, and some summary test information. Below that is a table that shows the test name, measurement details, various measurements, test result (pass/fail), Compliance Mode status (Yes/No), and analysis time. Additional test parameter information that does not fall into the other columns is put in the Comments column.

Report example

Tektronix

TekExpress BroadR-Reach

Transmitter Test Report

Setup Information			
DUT ID	DUT001	TekExpress BroadR-Reach	1.0.2.163 (EVALUATION)
Pre-Recorded Mode	True	FrameWork Version	4.0.6.264
Date/Time	2016-03-04 14:17:06	Scope Model	DPO7254C
Overall Execution Time	0:00:34	FirmWare Version	7.6.1 Build 2
Overall Analysis Time	0:00:05		
Overall Test Result	Pass		
DUT COMMENT:	General Comment - BroadR-Reach DUT		

Test Name Summary Table	
Transmit Clock Frequency	Pass

Transmit Clock Frequency							
Measurement Details	Iteration	Measured Value	Unit	Test Result	Margin	Low Limit	High Limit
Transmit Clock Frequency		66.668	MHz	Pass	L: -0.0077, H: 0.0056	66.6603	66.6736
COMMENTS	Number of unit Intervals : 50000						

Back to Summary Table

See Also

Setting Up Tests: the Setup Panel

Running the Tests and Viewing their Progress: the Status Panel

Viewing Test Results: the Results Panel

About Saving and Recalling Test Setups

Saving and Recalling Test Setups

About Saving and Recalling Test Setups

TekExpress BRR opens with the default test setup selected. Any time you want to create a new test setup, you can select the default test setup to clear the previous test setup selections and take the settings back to their defaults.

You can run a test before or after saving a setup. When you save a setup, the selected Oscilloscope, general parameters, acquisition parameters, measurement limits, prerecorded waveform files (if applicable), test, and other configuration settings are all saved under the setup name. You can open a setup and click **Start** button without having to do any other setting up except ensuring that the Oscilloscope is detected and ready. For details, see *Before You Click Start*.

See Also

Save a Test Setup

Recall a Saved Test Setup

Create a New Test Setup Based on an Existing One

Delete a Test Setup

Setting Up Tests: the Setup Panel

Saving a Test Setup

Save a test setup before or after running a test using the parameters you want saved. You can create a new setup from any setup you have open or from the default setup. When you select the default test setup, all parameters are returned to their defaults. The following instructions start from the default setup:

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

See Also

About Test Setups

Recall a Saved Test Setup

Create a New Test Setup Based on an Existing One

Delete a Test Setup

Recalling a Saved Test Setup

These instructions are for recalling saved test setups.

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

See also

About Test Setups

Save a Test Setup

Create a New Test Setup Based on an Existing One

Delete a Test Setup

Creating a New Test Setup Based on an Existing One

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

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

See also

About Test Setups

Save a Test Setup

Recall a Saved Test Setup

Delete a Test Setup

Deleting a Test Setup

If you no longer need a test setup, delete it from the test setup list in the Options menu using these instructions.

Each saved test setup consists of two main parts, the test setup file and the test setup folder, both named for the test session.

1. Make sure the setup you want to delete is not currently selected in TekExpress BRR.
2. Navigate to the BRR folder where test setup files are stored. For example, X:\BRR\(test setup name).
3. Locate the test setup file and then delete it. This removes the setup from the list in the **Options** menu.
4. Locate the test setup folder. If you want to keep any of the session files, move them out of the test setup folder and then delete the test setup folder.

See also

About Test Setups

Save a Test Setup

Recall a Saved Test Setup

Create a New Test Setup Based on an Existing One

BroadR-Reach measurements

Transmitter Clock Frequency

Test Setup

Specification

- BroadR-Reach Physical Layer Transceiver Specification for Automotive Applications, version 3.2, section 5.1.3 or IEEE P802d3bwTM (100 BASE T1) section 96.5.4 transmitter specifications.
- BroadR-Reach Physical Media Attachment Test Suite version 2.0, section 5.5

Required test equipment

In addition to the DUT and Oscilloscope, you will need the following:

- One supported differential probe.
- Short RJ45 connector
- Test Fixture: TF-GBE-BTP

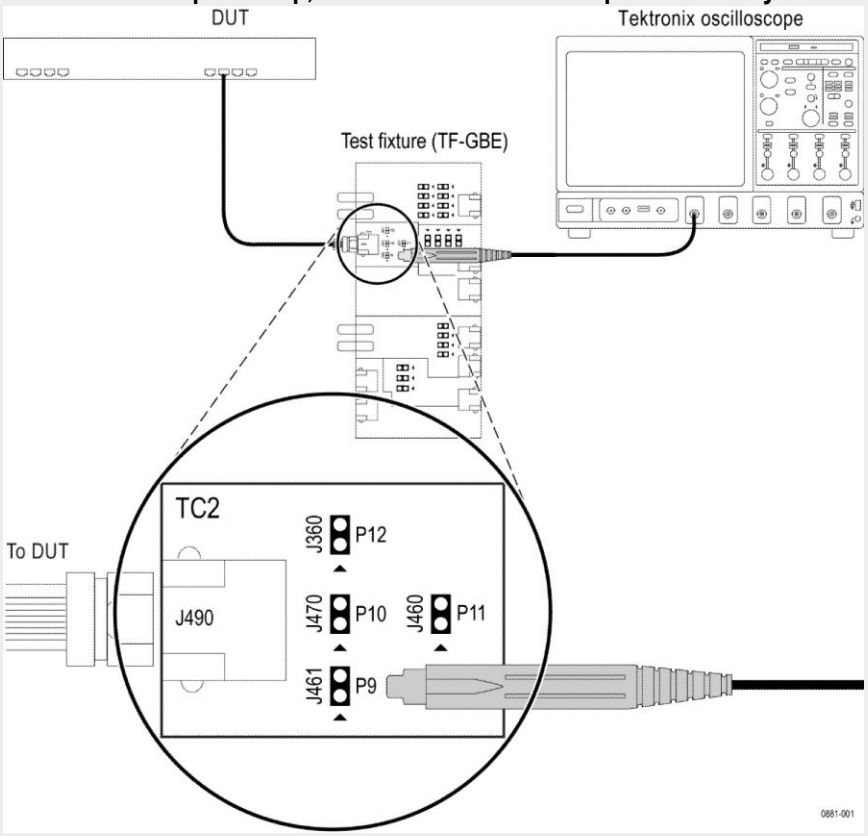
Test setup procedure for Transmitter Clock Frequency

The Transmitter Clock Frequency measurement setup involves the DUT, test fixture, and Oscilloscope.

Connect the equipment as shown in this connection diagram and as explained in the following procedure.

1. Set the DUT to generate and transmit test mode 1 signal.
2. Connect the Ethernet cable to J490 and the test port of the DUT. If the DUT does not have an RJ45 connector, then strip the RJ45 cable on one side and solder it to the appropriate pins on the DUT.
3. Connect the differential probe to P9 and the configured channel of the Oscilloscope.

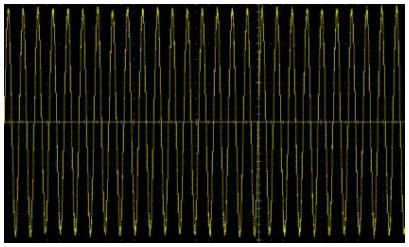
Connection diagram for Test setup procedure for Transmitter Clock Frequency, Transmitter Timing Jitter-Master Jitter, Transmitter Timing Jitter-Slave Jitter, Transmittter Output Droop, and Transmitter Power Spectral Density



NOTE.

Make sure "+" on the probe tip aligns with "<" on the text fixture board. This takes care of the polarity being not reversed.

Measurement Setup and Algorithm

Item	Requirements
Configuration parameters	1. Record Length
Signal type	Test mode 2 

Item	Requirements
Measurement algorithm outputs	1. Clock frequency 2. Number of unit intervals
Measurement algorithm inputs	Test mode 2 signal captured in differential form
Limits	Lower limit: 66.6603 MHz Upper limit: 66.6736 MHz
Plots	*Plot* showing only 4 cycles of clock frequency

Measurement Algorithm

BroadR-Reach Automated Compliance Solution automatically executes the calculations described below for the Transmit Clock Frequency measurement.

1. Check if the input waveform pattern is a test mode 2 or not. The test mode 2 signal is a clock signal where the PHY shall transmit the data symbol sequence (+1, -1) repeatedly on all channels.
2. Compute and verify the clock frequency between the High and Low limits as mentioned in the above table.

See also

Plots

Transmitter Timing Jitter Master and Slave

This test confirms that the transmitter timing jitter of the PMA is within conformance limits.

Test Setup (Master Jitter)

Specification

- BroadR-Reach Physical Layer Transceiver Specification for Automotive Applications, version 3.2, section 5.4.3 or IEEE P802d3bwTM (100 BASE T1) section 96.5.4 transmitter specifications.
- BroadR-Reach Physical Media Attachment Test Suite version 2.0, section 5.3

Required test equipment

In addition to the DUT and Oscilloscope, you will need the following:

- One supported differential probe.
- Short RJ45 connector
- Test Fixture: TF-GBE-BTP

Test setup procedure for Timing Jitter (Master) test

The Timing Jitter (Master) measurement setup involves the DUT, test fixture, and Oscilloscope.

Connect the equipment as shown in this connection diagram and as explained in the following procedure.

1. Set the DUT to generate and transmit a test mode 2 signal.
2. Connect the Ethernet cable to J490 and the test port of the DUT. If the DUT does not have a RJ45 connector, then strip the RJ45 cable on one side and solder it to the appropriate pins on the DUT.
3. Connect the differential probe to P9 and the configured channel of the Oscilloscope.

Test Setup (Slave Jitter)

Specifications

- BroadR-Reach Physical Layer Transceiver Specification for Automotive Applications, version 3.2, section 5.4.3 or IEEE P802d3bwTM (100 BASE T1) section 96.5.4 transmitter specifications.
- BroadR-Reach Physical Media Attachment Test Suite version 2.0, section 5.3.

Required test equipment

In addition to the DUT and Oscilloscope, you will need the following:

- One supported differential probe.
- Short RJ45 connector
- Test Fixture: TF-GBE-BTP
- Link Partner (This is needed to put the DUT to slave mode.)

Test setup procedure for Timing Jitter (Slave) test

The Timing Jitter (Slave) measurement setup involves the DUT, test fixture, link partner, and Oscilloscope.

Connect the equipment as shown in this connection diagram and as explained in the following procedure.

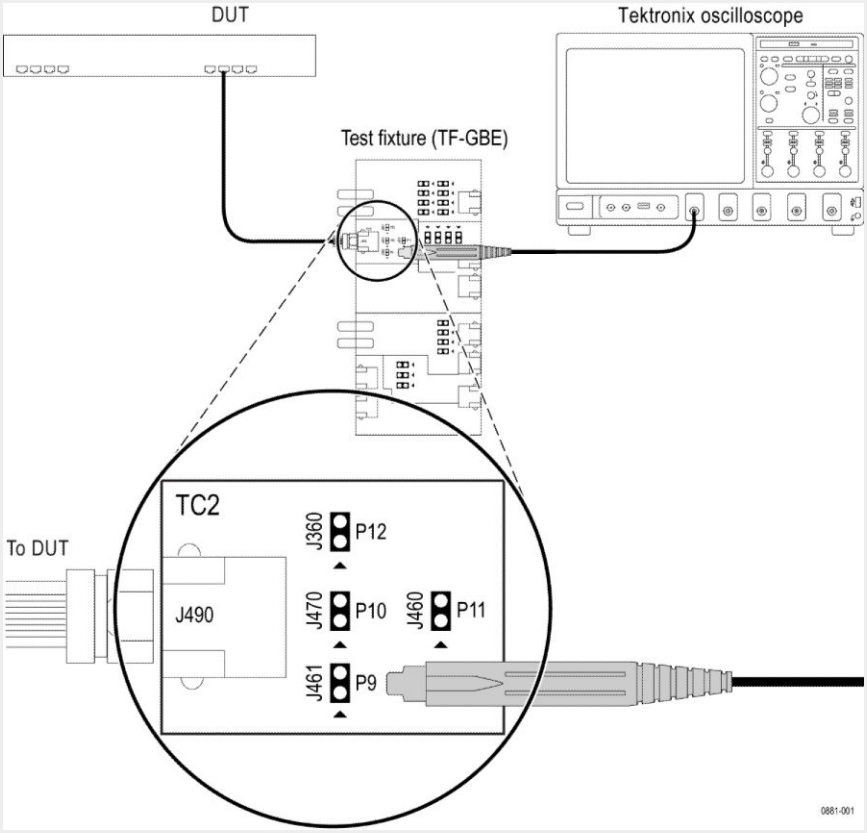
1. Set the DUT to operate in normal mode.
2. Connect the DUT to the link partner and establish a normal Ethernet connection.

3. Use the appropriate cable or probe to bring the DUT TX_TCLK to the configured channel of the Oscilloscope.

NOTE. *You will need to choose either an SMA cable, a BNC cable, or a differential probe based on the interface available on the DUT for TX_TCLK. .*

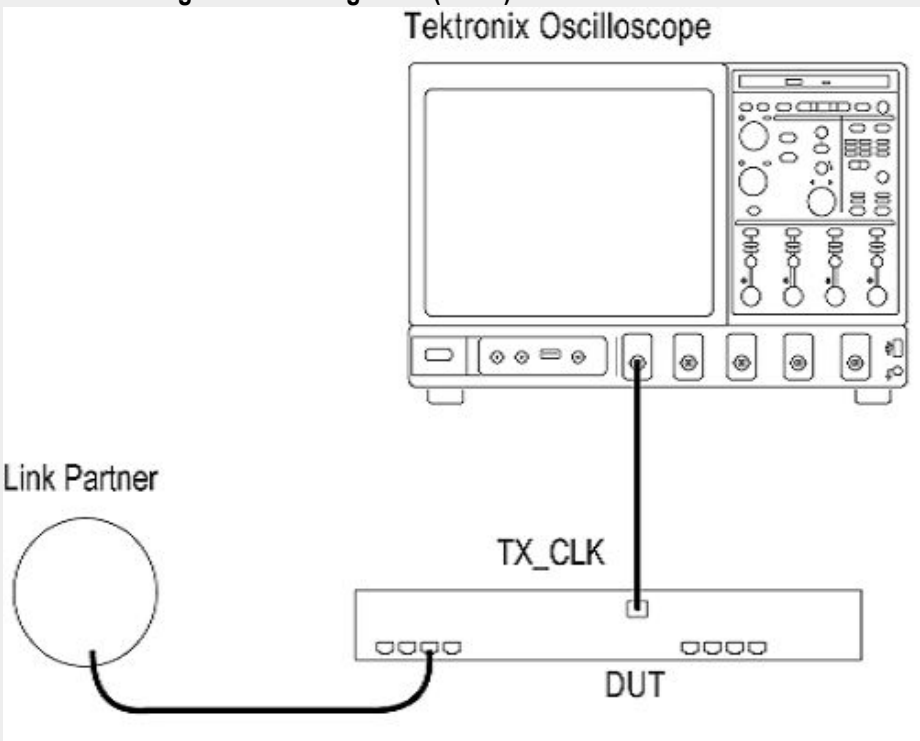
Slave transmitter timing jitter can only performed when the DUTs TX_TCLK is exposed and accessible.

Test setup procedure for Transmitter Clock Frequency, Transmitter Timing Jitter-Master Jitter, Transmitter Timing Jitter-Slave Jitter, Transmittter Output Droop, and Transmitter Power Spectral Density

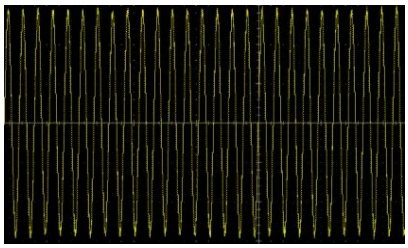
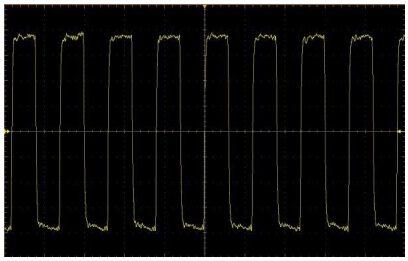


NOTE. *Make sure "+" on the probe tip aligns with "<" on the text fixture board. This takes care of the polarity being not reversed*

Connection diagram for Timing Jitter (Slave) test



Measurement setup and algorithm

Item		Requirements
Configuration parameters		a. Hysteresis b. Edge Type
Signal type		Test mode 2 (master) and TK_TCLK (slave)  
Measurement algorithm outputs		
	MASTER	RMS TIE between test mode 2 waveform and unjittered reference
	SLAVE	RMS TIE between TX_TCLK waveform and unjittered reference Number of edges considered for computation
Measurement algorithm inputs		
	MASTER	Minimum 1 ms duration test mode 2 signal captured in differential form. Hysteresis Edge Type
	SLAVE	Minimum 1 ms duration TX_TCLK signal captured in differential form. Hysteresis Edge Type
Limits		
	MASTER	Lower limit: NA Upper limit: 50 ps
	SLAVE	Lower limit: NA Upper limit: 150 ps
Plots		Time trend of TIE

Measurement Algorithm (MASTER)

BroadR-Reach Automated Compliance Solution automatically executes the calculations described below.

1. Edge locations in test mode 2 waveform (captured in differential form) are determined with middle level percentage as 50%. Hysteresis and EdgeType are entered by user in GUI.
2. The least square method is used for straight line fit, with Y as the edge locations and X as the edge indices. Only edges that match the EdgeType entered by the user in the GUI are considered for fitting. Slope and intercept are determined as the output of the least square method. Using the intercept and slope, reconstructed edge locations are determined (un-jittered reference).

   reconstructedTime(I) = intercept + I * slope, I = 0 to num_edges(num_edges depend on EdgeType)
3. Compute TIE on EdgeType entered by user in GUI.

   TIE(I) = reconstructedTime(I) – EdgePosition(I), I = 0 to num_edges(num_edges depend on EdgeType).

   Compute rms value of TIE, which is reported as the result.

Measurement Algorithm (SLAVE)

BroadR-Reach Automated Compliance Solution automatically executes the calculations described below.

1. Edge locations in TX_TCLK waveform (captured in differential form) are determined with middle level percentage as 50%. Hysteresis and EdgeType are entered by user in GUI.
2. The least square method is used for straight line fit, with Y as the edge locations and X as the edge indices. Only edges that match the EdgeType entered by the user in the GUI are considered for fitting. Slope and intercept are determined as the output of the least square method. Using the intercept and slope, reconstructed edge locations are determined (un-jittered reference).

   reconstructedTime(I) = intercept + I * slope, I = 0 to num_edges(num_edges depend on EdgeType)
3. Compute TIE on EdgeType entered by user in GUI.

   TIE(I) = reconstructedTime(I) – EdgePosition(I), I = 0 to num_edges(num_edges depend on EdgeType).

   Compute rms value of TIE, which is reported as the result.

See also *Plots*

Transmitter Output Droop

This test confirms that the transmitter output level does not decay faster than the maximum specified rate.

Test Setup

Specification

- BroadR-Reach Physical Layer Transceiver Specification for Automotive Applications, version 3.2, section 5.4.1 or IEEE P802d3bwTM (100 BASE T1) section 96.5.4 transmitter specifications.
- BroadR-Reach Physical Media Attachment Test Suite version 2.0, section 5.1

Required test equipment

In addition to the DUT and Oscilloscope, you will need the following:

- One supported differential probe.
- Short RJ45 connector
- Test Fixture: TF-GBE-BTP

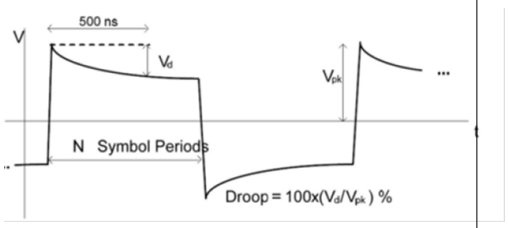
Test setup procedure for Transmitter Output Droop

The Transmitter Output Droop measurement setup involves the DUT, test fixture, and Oscilloscope.

Connect the equipment as shown in this connection diagram and as explained in the following procedure.

1. Set the DUT to generate and transmit test mode 1 signal.
2. Connect the Ethernet cable to J490 and the test port of the DUT. If the DUT does not have an RJ45 connector, then strip the RJ45 cable on one side and solder it to the appropriate pins on the DUT.
3. Connect the differential probe to P9 and the configured channel of the Oscilloscope.

Measurement Setup and Algorithm

Item	Requirements
Configuration parameters	1. Averages
Signal type	Test mode 1 
Measurement algorithm outputs	1. Positive Droop in % 2. Number of positive droops 3. Negative Droop in % 4. Number of negative droops
Measurement algorithm inputs	Test mode 1 signal captured in differential form
Limits	Lower limit: NA Upper limit: 45%

Measurement Algorithm

BroadR-Reach Automated Compliance Solution automatically executes the calculations described below for the transmitter output droop measurement.

1. Check if the input waveform pattern is a square wave test pattern or not.
2. If it is square wave test pattern, then check whether positive and negative widths are greater than 33 unit intervals (500 nS).
3. On each rising edge, compute the maximum voltage ($V_{p_positive}$) and compute the voltage ($V_{droop_positive}$) 500 ns after that maximum voltage index.

   $V_d = (V_{p_positive} - V_{droop_positive})$

   Positive Droop = $(V_d/V_{p_positive}) * 100$
4. On each falling edge, compute the minimum voltage ($V_{p_negative}$) and compute the voltage ($V_{droop_negative}$) 500 ns after that minimum voltage index.

   $V_d = (V_{p_negative} - V_{droop_negative})$

   Negative Droop = $(V_d/V_{p_negative}) * 100$

See also *Plots*

Transmitter Power Spectral Density

This test confirms that the transmitter power spectral density is within conformance limits.

Test Setup

Specification

- BroadR-Reach Physical Layer Transceiver Specification for Automotive Applications, version 3.2, section 5.4.4 or IEEE P802d3bwTM (100 BASE T1) section 96.5.4 transmitter specifications.
- BroadR-Reach Physical Media Attachment Test Suite version 2.0, section 5.4

Required test equipment

In addition to the DUT and oscilloscope, you will need the following:

- One supported differential probe.
- Short RJ45 connector
- Test Fixture: TF-GBE-BTP

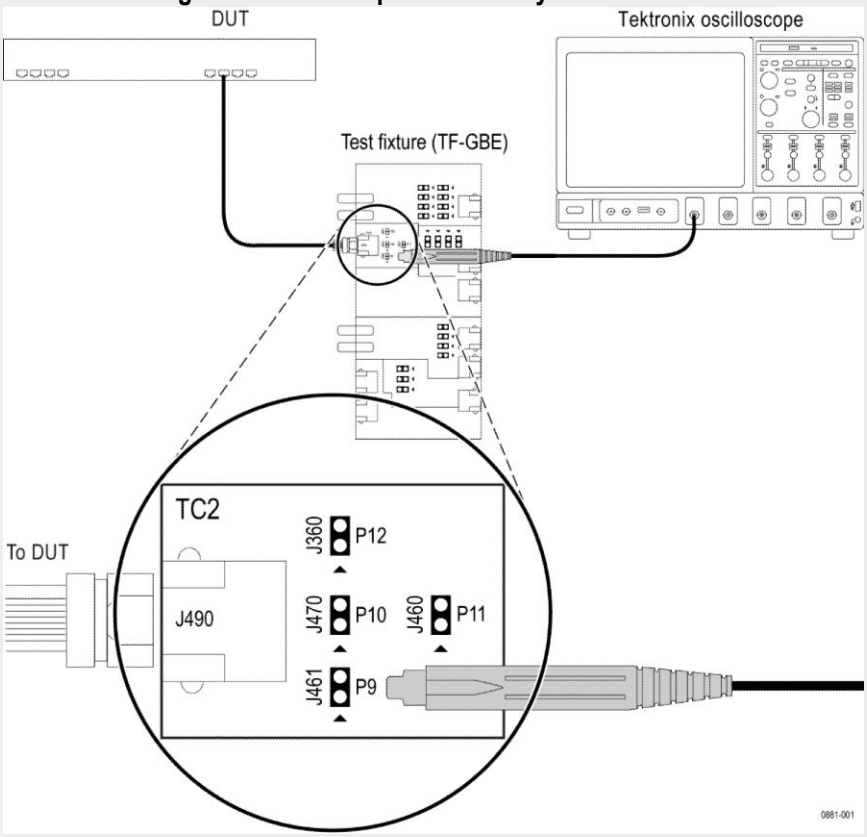
Test setup procedure for Power Spectral Density test

The Power Spectral Density test involves the DUT, test fixture, and Oscilloscope.

Connect the equipment as shown in this connection diagram and as explained in the following procedure.

1. Set the DUT to generate and transmit test mode 5 signal.
2. Connect the Ethernet cable to J490 and the test port of the DUT. If the DUT does not have an RJ45 connector, then strip the RJ45 cable on one side and solder it to the appropriate pins on the DUT.
3. Connect the differential probe to P9 and the configured channel of the Oscilloscope.
4. Make sure "+" point of differential probe should be near |>.

Connection diagram for Power Spectral Density

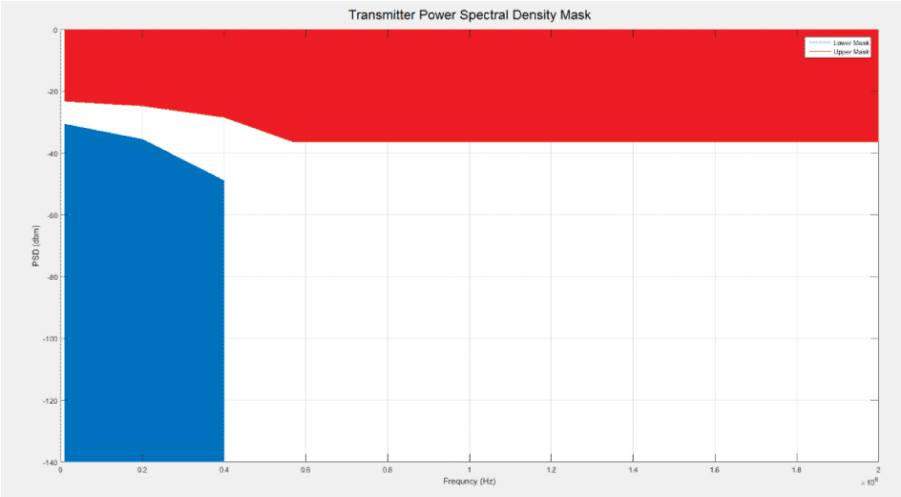


NOTE. *Make sure "+" on the probe tip aligns with "<" on the text fixture board. This takes care of the polarity being not reversed.*

Power Spectrum Curve Mask Information

Frequency	PSD Upper Bound (dBm)	PSD Lower Bound (dBm)
at 1 MHz	–23.3	–30.9
at 20 MHz	–24.8	–35.8
at 40 MHz	–28.5	–40.2
57 MHz - 200 MHz	–36.5	– NA

Upper and lower limits/masks are piece-wise linear masks connecting points given in the above table.



Settings. RBW = 10 KHz, VBW = 30 KHz, sweep time > 1 min, RMS detector, sweep time 3.275 seconds

Measurement Algorithm

BroadR-Reach Automated Compliance Solution automatically executes the calculations described below for the power spectral density measurement.

1. The test mode 5 signal feed as input is filtered using a band limiting filter with cut off frequency of 600 MHz.
2. Spectral functions (SpectralMag) present in the Oscilloscope MATH sub system are used for computing the spectrum of the filtered test mode 5 signal.
3. The output of “SpectralMag” is averaged over “Averages” times. “Averages” is an acquisition parameter present in the GUI. The default value of this parameter is 64.
4. ”SpectralMag” is executed with the following configuration settings:
   - Center frequency: 100.5 MHz
   - Frequency span: 201 MHz
   - Window type: Gaussian
   - Vertical axis: scale —> Linear
   - Gating duration: 200 micro seconds
   - Gating duration = Record length * sampling rate
     - Mega samples / 12.5 GS/sec = 200 micro seconds
   - Gating position: 0
   - R = 100 Ohm
5. Averaged SpectralMag is saved and used in postprocessing.
6. In postprocessing, the averaged spectral output is filtered using a moving average filter (smoothing).
7. Filtered result is X(k). From X(k), we compute the power in dBm as given below.

Power = (V_rms^2)/R

Power in frequency domain = |X(k)|^2/R where k = 0 to N-1 are frequency bins

Power in dB = 20*log10(|X(k)|) – 10*log10(R)

Power in dBm = Power in dB + 10*log10(1000)

Power in dBm = Power in dB + 10*log10(1000)

20*log10(|X(k)|) – 10*log10(R) + 10*log10(1000)

If R = 100 Ohm,

Power in dBm = 20*log10(|X(k)|) – 20 + 30

Power in dBm = 20*log10(|X(k)|) – (20 - 30)

Power in dBm = 20*log10(|X(k)|) – Offset

Where Offset = (20 – 30)

8. Computed power in dBm is compared with limits given in the specification, which gives limits only at certain frequencies. Linear interpolation is used to find values for intermediate frequencies.

See also *Plots*

Transmitter Distortion

This test confirms that the peak transmitter distortion is less than 15 mV for all 10 UIs within the eye opening.

Test Setup

Specification

- BroadR-Reach Physical Layer Transceiver Specification for Automotive Applications, version 3.2, section 5.4.2 or IEEE P802d3bwTM (100 BASE T1) section 96.5.4 transmitter specifications.
- BroadR-Reach Physical Media Attachment Test Suite version 2.0, section 5.2

Required test equipment

In addition to the DUT and Oscilloscope, you will need the following:

NOTE. *You will need to complete the calibrate procedure before doing the Transmitter Distortion measurement with disturbing signal. The calibrate procedure is used to effectively remove the disturbing signal and compensate for nonlinearity in the disturber and test fixture.*

- One supported differential probe.
- Two BNC cables (for connecting AFG or AWG5K to fixture) or two SMA cables with two BNC to SMA connectors (for connecting AWG7K to fixture)
- Short RJ45 connector
- GPIB cable (required if you use *AWG automation*, connects AWG and Oscilloscope)
- Test Fixtures: TF-GBE-BTP
- TF-BRR-CFD (Clock Frequency Divider Unit): This is used to synchronize Oscilloscope and signal source with the DUT Transmit _CLK.

  NOTE. *If you are using R&S Clock divider fixture, then the output from the R&S fixture must be terminated with 50 ohms, which goes to Oscilloscope as external REF Clock input.*

Connect the equipment as shown in the below connection diagram.

Test Setup Procedure for Transmitter Distortion

This measurement is divided into the following two parts. Do them in this order:

1. **Calibration**
2. **Measurement with disturbing signal**

Calibration is used to effectively remove the disturbing signal and compensate for nonlinearity in the disturber and test fixture. Calibration is used only when measurements are executed with a disturbing signal.

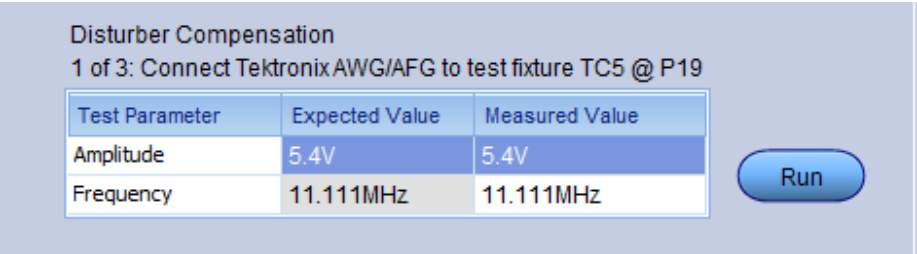
Design of the transmitter to tolerate the presence of the remotely driven signal with acceptable distortion or other changes in performance is a critical issue and must be addressed by the implementer. A disturbing signal is used to simulate the presence of a remote transmitter. The disturbing signal is defined as a sine wave generator that simulates the potential interfering effect of another transmitter.

BroadR-Reach (100BASE T1) measurements that require test mode 4 have to be done with a disturbing signal. Characteristics of disturbing signal are given in the following table.

Table 9: Characteristics of disturbing signal

Test mode	Waveform	Characteristics of the waveform		Purity
		Amplitude (p-p)	Frequency (MHz)	
4	Sinusoidal	5.4 V	11.111	All harmonics > 40 dB below fundamental

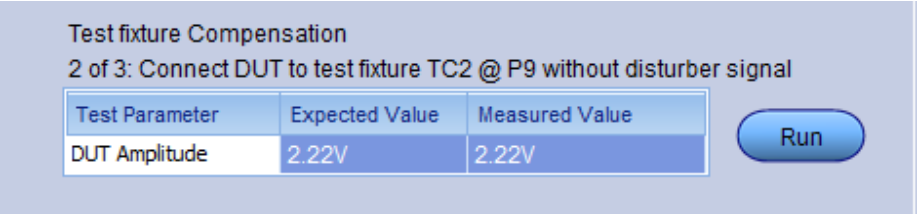
Disturber compensation

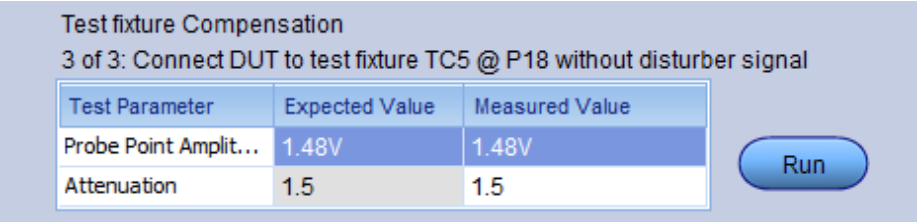


Disturbing signal is fed from AWG/AFG. As part of disturbing signal compensation, do the following:

1. Measure peak amplitude and frequency of disturbing signal.
2. Calibrate/adjust the AWG so that measured amplitude and frequency are as close as possible to default/expected values as mentioned in Table 8.
3. Use the final measured amplitude and frequency values as initial values for disturbing signal removal.

Text fixture compensation





1. Measure the peak amplitude of the signal which is acquired when both DUT signal (test mode 4) and disturbing signal are ON. Let this peak amplitude be Amp1 (Refer, B- Test Fixture Compensation diagram).
2. Measure the peak amplitude of the signal which is acquired when only DUT signal (test mode 4) is ON and disturbing signal is OFF. Let this be Amp2. An attenuation factor Amp1/Amp2 is computed. This is used as attenuation compensation factor.

AFG Configuration

The AFG setup happens automatically when you click the **Run** button at Step-1 of Disturber Compensation on Calibration tab and also sets the Signal Generator on Global Settings tab.

Following parameters are set for AFG:

1. Resets the AFG.
2. Sets amplitude to 5.4 Vpp.
3. Sets frequency to 11.111 MHz.
4. Sets the external Ref clock.
5. Turns on CH1 and CH2.

Point 1, 2, and 3 are applicable for both CH1 and CH2 and the phase set to CH1 is 0 degrees and for CH2, it is 180 degrees.

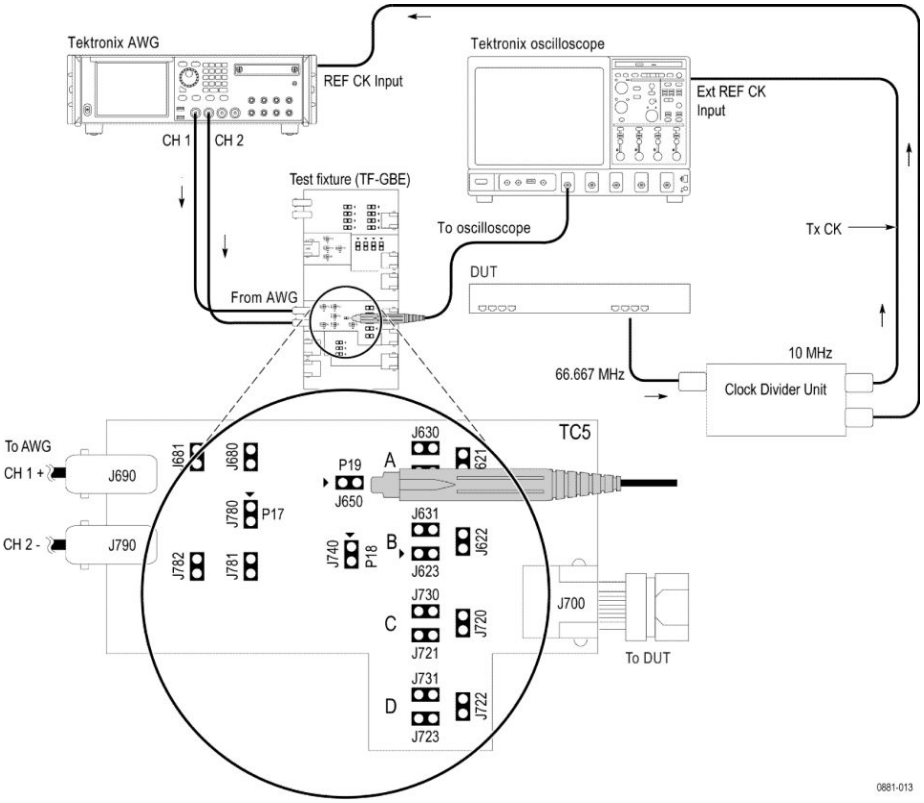
NOTE.

- *Make sure that before clicking the **Start** button the calibration steps are performed.*
- *During calibration, the signal source AFG/AWG will be setup automatically from the application, by loading the disturber pattern.*
- *When you click the **Start** button, the AFG/AWG does not setup, only Oscilloscope will setup because the application will not disturb the calibration done using AFG/AWG.*

The application automatically controls the AWG 5K and 7K similarly to that of AFG as described above.

Calibration Compensation Connection Diagram

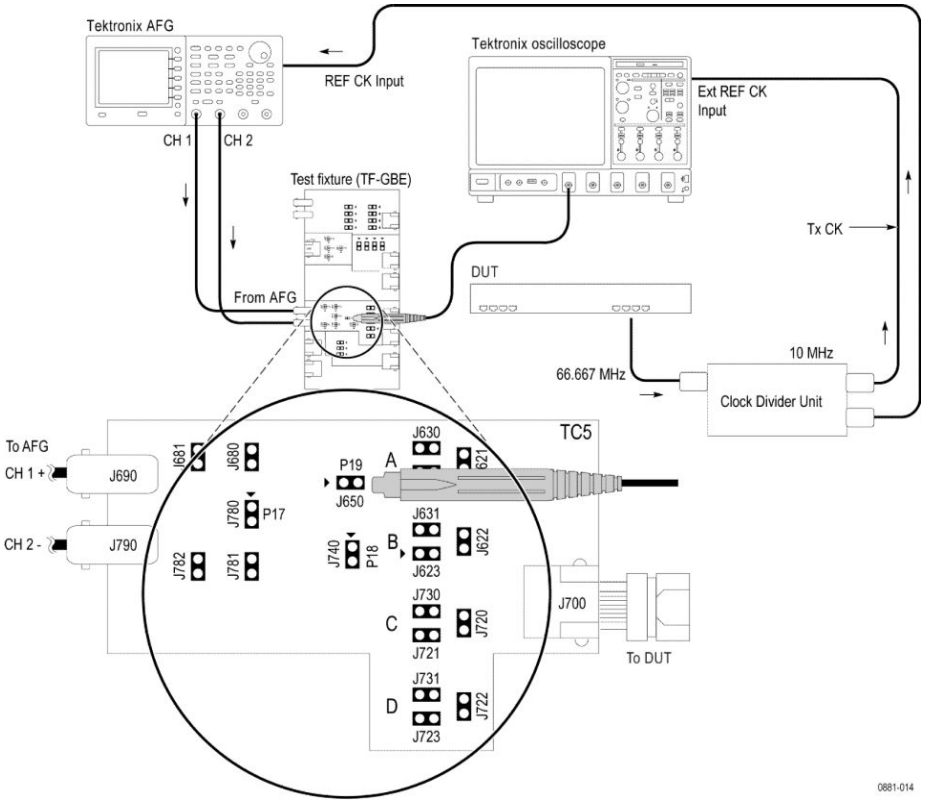
A. Disturbing signal compensation (AWG)



NOTE.

- *Clock Divider Unit is not used during Disturber signal compensation. Application will setup the reference clock input to internal.*
- *Without an amplifier, the AWG7000 series generates a maximum voltage of 2 Volt peak-to-peak signal, which cannot meet the 5.4 Volt requirement for the disturber signal for Transmitter Distortion test.*
- *Make sure "+" on the probe tip aligns with "<" on the text fixture board. This takes care of the polarity being not reversed.*

Disturbing signal compensation (AFG)



Connection diagram for disturbing signal compensation (part of calibration)

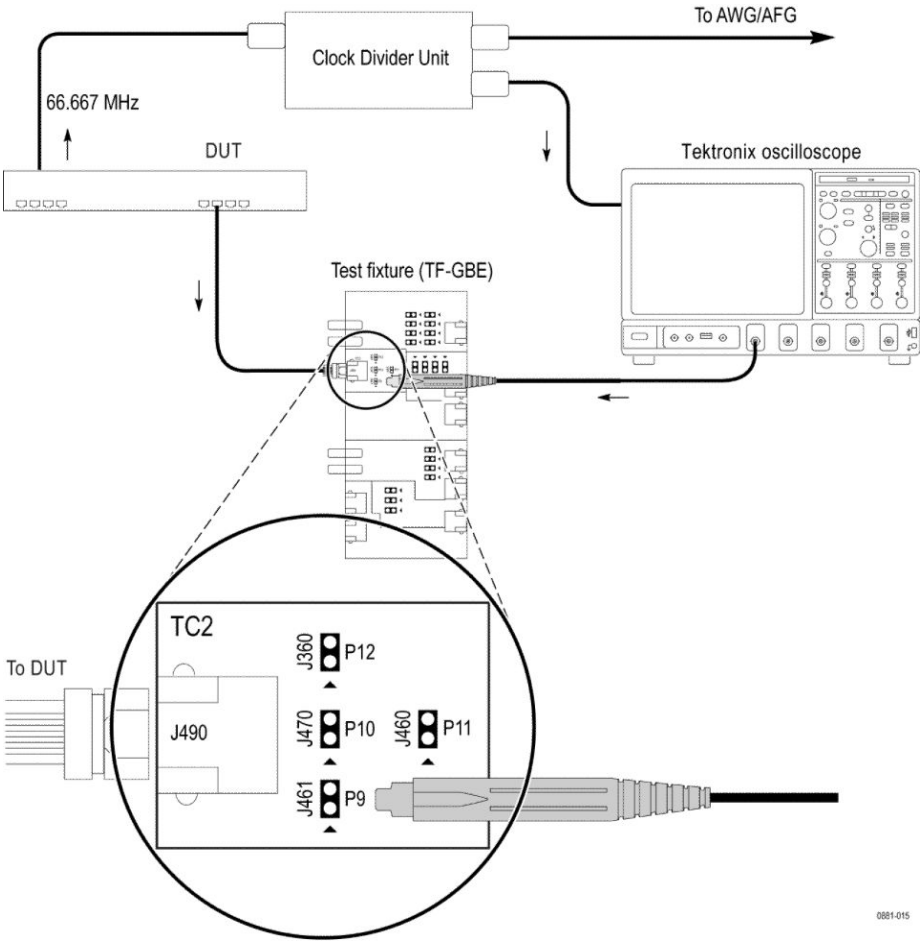
Make the connection as shown above:

1. Connect a BNC Cable to (AWG/AFG)"+" and Channel 1 of Arbitrary Waveform Generator/Arbitrary Function Generator to J690.
2. Connect a BNC Cable to (AWG/AFG)"-" and Channel 2 (CH1_inverted) of Arbitrary Waveform Generator/Arbitrary Function Generator to J790.
3. Short the jumpers J621, J630, J623, J721, J723, J680,and J781.
4. Connect the Differential Probe to P19 and configured channel of the Oscilloscope.
5. In the Calibration tab, select the **Run** button in the Disturber Compensation grid.
6. If the Measured Value is not approximately equal to the Expected Value, modify the amplitude and clock frequency settings of the Arbitrary Waveform Generator/Arbitrary Function Generator; then, click Measure and compare the values to be approximately equal.
7. Select the **Run** button to compensate for disturber signal to meet expected values of 5.4 Vpp and 11.111 MHz

NOTE.

- *Set the attenuator factor to 10X on the probe, if you are using P6247\6248. This is applicable for Disturbing Signal Compensation, Test Fixture Compensation and Measuring amplitude sections.*
- *Make sure "+" on the probe tip aligns with "<" on the text fixture board. This takes care of the polarity being not reversed.*
- *At the calibration level, application uses the internal reference clock.*

B. Test Fixture Compensation



Measuring amplitude with DUT signal ON

Connection diagram for Test Fixture Compensation (DUT signal)

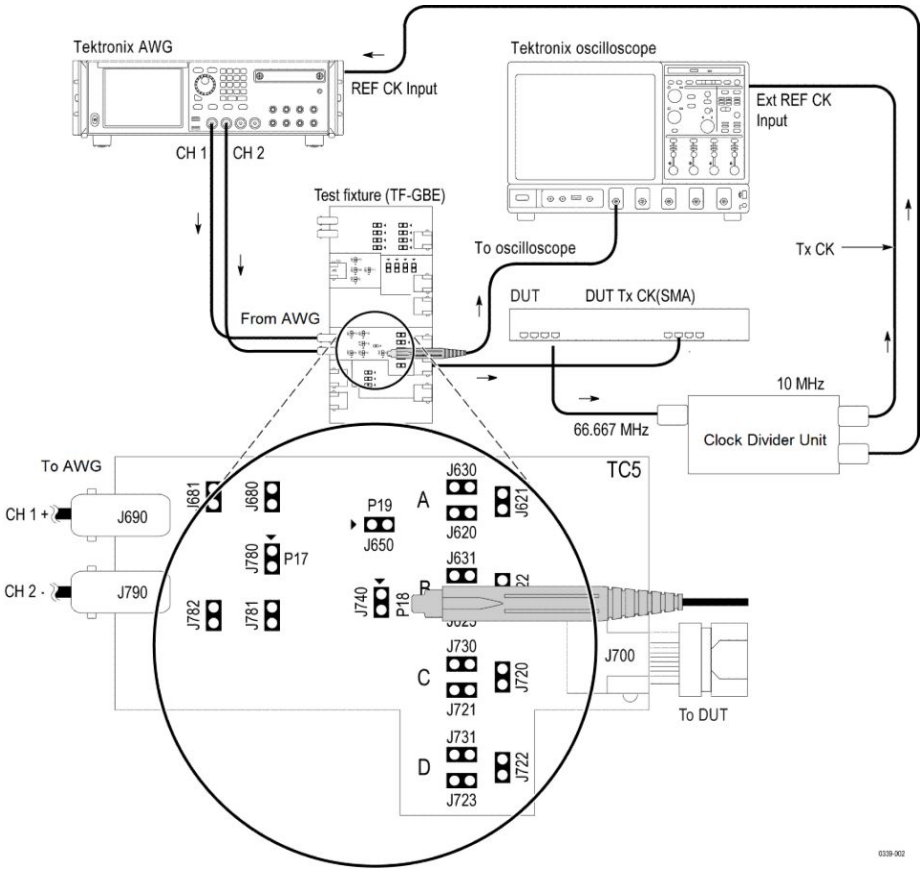
Make the connections as shown above:

1. Set the DUT to generate Test Mode 4 signal.
2. Connect the Ethernet cable to J490 and the test port of the DUT.
3. Connect the Differential Probe to P9 and configured channel of the Oscilloscope.
4. In the Calibration tab, select **Run** button in Step-2 in the Test Fixture Compensation pane.

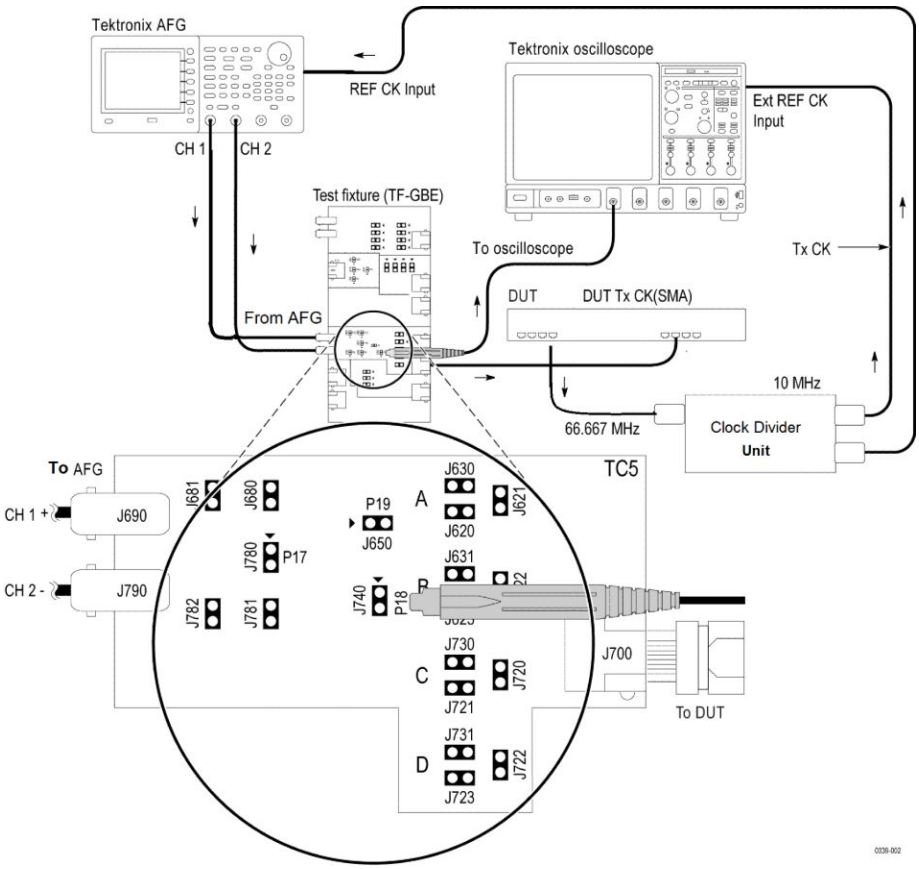
NOTE. *Make sure "+" on the probe tip aligns with " < " on the text fixture board. This takes care of the polarity being not reversed.*

C. Measuring amplitude with disturbing signal OFF and DUT signal ON

AWG



AFG



Connection diagram for Test Fixture Compensation (DUT signal is ON and disturbing signal is OFF)

Connect the equipment as explained in the following procedure.

1. Set the DUT to generate and transmit test mode 4 signal.
2. Connect the Ethernet cable to J700 and the test port of the DUT.
3. Disconnect the BNC cable to (AWG/AFG) at J690 and J790 point on text fixture.
4. If AWG/AFG is already connected, switch off the disturber signal generator.
5. Short jumpers J621, J630, J623, J721, J723, J680 and J781.
6. Connect the differential probe to P18 and the configured channel of the Oscilloscope.
7. In the Calibration tab, select the**Run** button in Step-3 in test fixture compensation pane.

NOTE. *Make sure "+" on the probe tip aligns with " < " on the text fixture board. This takes care of the polarity being not reversed.*

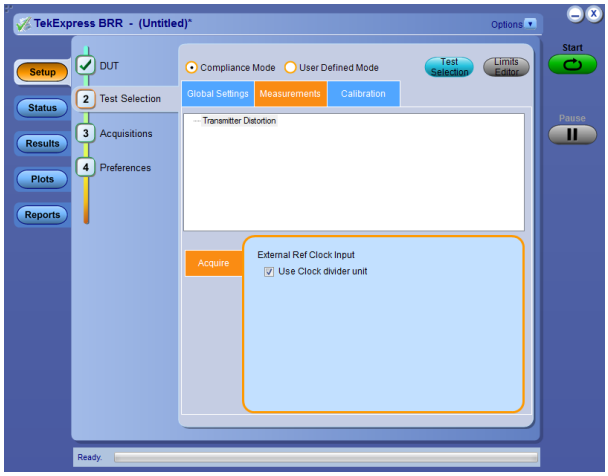
Measurement for Transmitter Distortion

1. Click **Apply** button after completion of Calibration.

2. Make the connection setup with clock divider as shown in the C diagram. The clock divider synchronizes the Oscilloscope and the disturber source.

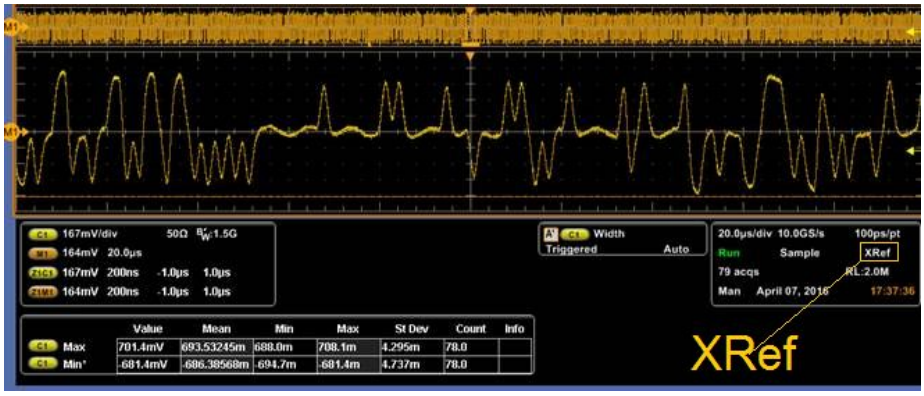
   In case the automaton with AFG/AWG is selected in the Global setting tab, the disturber pattern is automatically loaded during the disturber compensation process.

   The "Use Clock divider unit" option is selected by default.

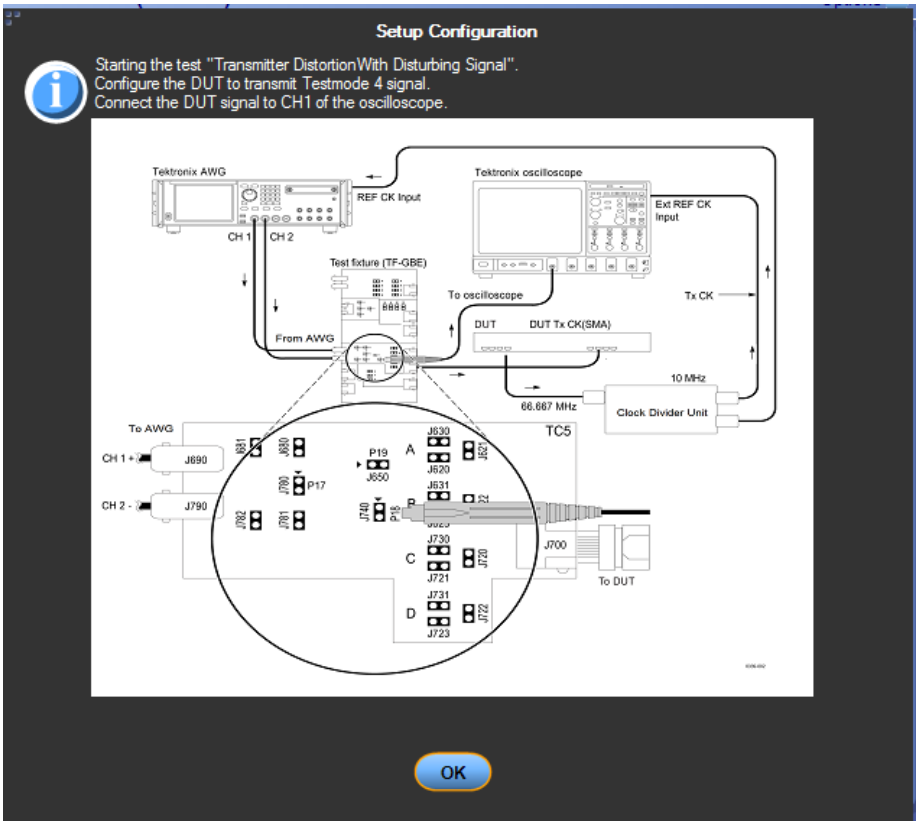


NOTE.

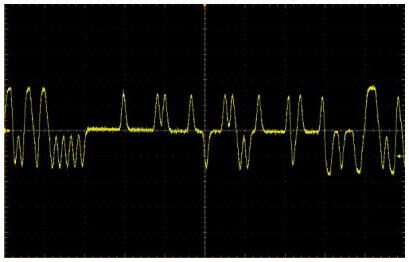
- *In case of unstable 10MHz clock output from the Clock Divider Unit, uncheck the "Use Clock divider unit" option.*
- *Check the amplitude of 10MHz clock coming from Clock Divider Unit, which should not exist the limit mentioned on the Oscilloscope and AFG/ AFG for 10MHz reference input signal.*
- *If External Reference Clock is stable 10MHz, Oscilloscope will phase lock with external reference clock. This can be seen in below image as "XRef", which indicates acquired signal is properly locked. Oscilloscope displays "NoRef", in case of unlocked phase. It indicates External Reference Clock is not proper and can result in incorrect results.*



3. Click the **Start** button to start the test and do the connections as per the below schematics, and then **OK** button to continue the measurement.



Measurement setup and algorithm

Item	Requirements
Configuration parameters	Use Clock divider unit
Signal type	Test mode 4 
Measurement algorithm outputs	All the computed 10 peaks must be less than 15mV.
Measurement algorithm inputs	Test mode 4 signal captured in differential form Disturbing Signal
Limits	Lower limit: NA Upper limit: 15 mV

BroadR-Reach Automated Compliance Solution automatically executes the calculations described below for the transmitter distortion measurement.

1. Compute the unit interval of the acquired test mode 4 signal.
2. Determine the number of frames and the start and end of the frames (each frame is 2047 bits long) in the acquired signal. Neglect the residual signal present at both the start and the end of the acquired signal. Make sure that all further processing happens from start of the kth frame to end of nth frame.
3. Compute the DC offset and remove the DC offset from the signal. Filter the signal using an HPF with a cutoff frequency of 1.068 MHz If LPF is enabled, then filter the signal using an LPF with a cutoff frequency of 33 1/3 MHz.
4. Using the start and end frames information, do averaging across frames and finally get an averaged frame containing 2047 symbols or (2047*k) samples where *k* is number of samples per symbol.
5. Do single acquisition and collect 2047 samples corresponding to a phase offset. These 2047 samples are picked from 2047 symbols and so each sample represents a symbol.

   The application removes the disturber signal as per the IEEE 802.3 MATLAB code.
6. Repeat the previous two steps for phase offsets 0 to 1 UI in steps of 0.1 UI (10 phase offsets). A total of 10 peak errors/peak distortions are obtained.

See also

Plots

Know limitations

Return Loss

This test confirms that the Return Loss of the DUT is within conformance limits.

Test setup: Return Loss Calibration and Return Loss Measurement

Specification

- BroadR-Reach Physical Layer Transceiver Specification for Automotive Applications, version 3.2, section 8.2.2
- BroadR-Reach Physical Media Attachment Test Suite version 2.0, section 5.6

Required test equipment

In addition to the DUT and Oscilloscope, you will need the following:

- Two supported differential probes.
- Three BNC cables (for connecting AFG or AWG5K to fixture) or three SMA cables with two BNC to SMA connectors (for connecting AWG7K to fixture) and TCA-BNC or TCA-SMA adapters (auxiliary)
- Four Inch Short RJ45 connector
- GPIB cable (required if you use *AWG automation*, connects AWG and Oscilloscope)
- Test Fixtures: TF-GBE-BTP and fixture with loads (Open, Short, and Load)

Test setup procedure for Return Loss test

The Return Loss measurement setup involves the DUT, test fixture, AWG/AFG, and Oscilloscope.

Connect the equipment as shown in the below connection diagram and as explained in the following procedure.

1. Make the connections as shown in the below connection diagram.
2. Perform the Load, Open, and Short*Calibration*.
3. Set the DUT to generate a test mode signal 4.
4. Connect a BNC cable/SMA cable to (AWG/AFG)+ terminal on TC1 and Channel 1 of AWG/AFG.

5. Connect a BNC cable/SMA cable to (AWG/AFG)– and $\overline{CH1}$ (CH1 inverted) of the AWG/AFG *3*.

NOTE. *Make sure Channel 1 is connected to P1 of TC1 of TF-GBE-BTP and Channel 2 to P2 of TC1. Do not reverse Channel 1 to P2 and Channel 2 to P1. This will affect the machine.*

---

6. Connect the marker1 to the auxiliary channel of Oscilloscope.
7. Connect one differential probe to P1 of TC1 fixture and another differential probe to P2 of TC1 fixture.
8. Connect the GPIB/LAN/USB cable between AWG/AFG and the Oscilloscope.
9. Select the Signal Generator option from Global Settings Tab.

NOTE. *List of signal generators are shows in Global Settings Tab based on TekExpress Instrument Control Settings. Here you can select the connected signal source in Search Criteria and then click* ***Refresh*** *button. The connected signal sources will automatically display in Retrieved Instruments list and Global Settings's Tab under 'Signal Generator' list.*

NOTE. *To setup the signal source manually, copy the related pattern from C:\Program Files\Tektronix\TekExpress\TekExpress BroadR-Reach\AWG Waveforms\Return Loss to USB memory stick. Recall this setup file on connected signal source.*

---

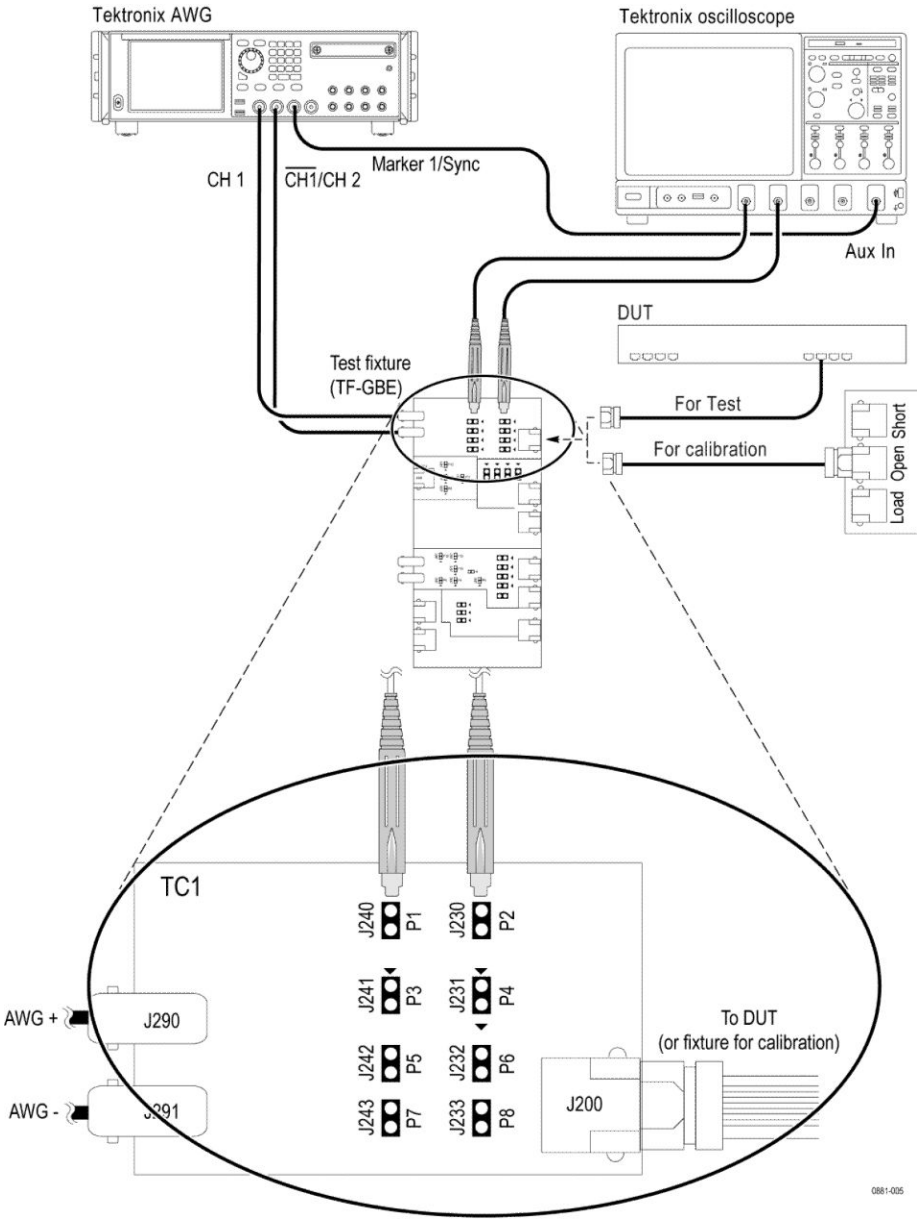
10. Click **RUN** button.

    It will automatically transfer the Return loss pattern data to AWG/AFG, whichever is been selected, in the Signal generator of Global setting tab.

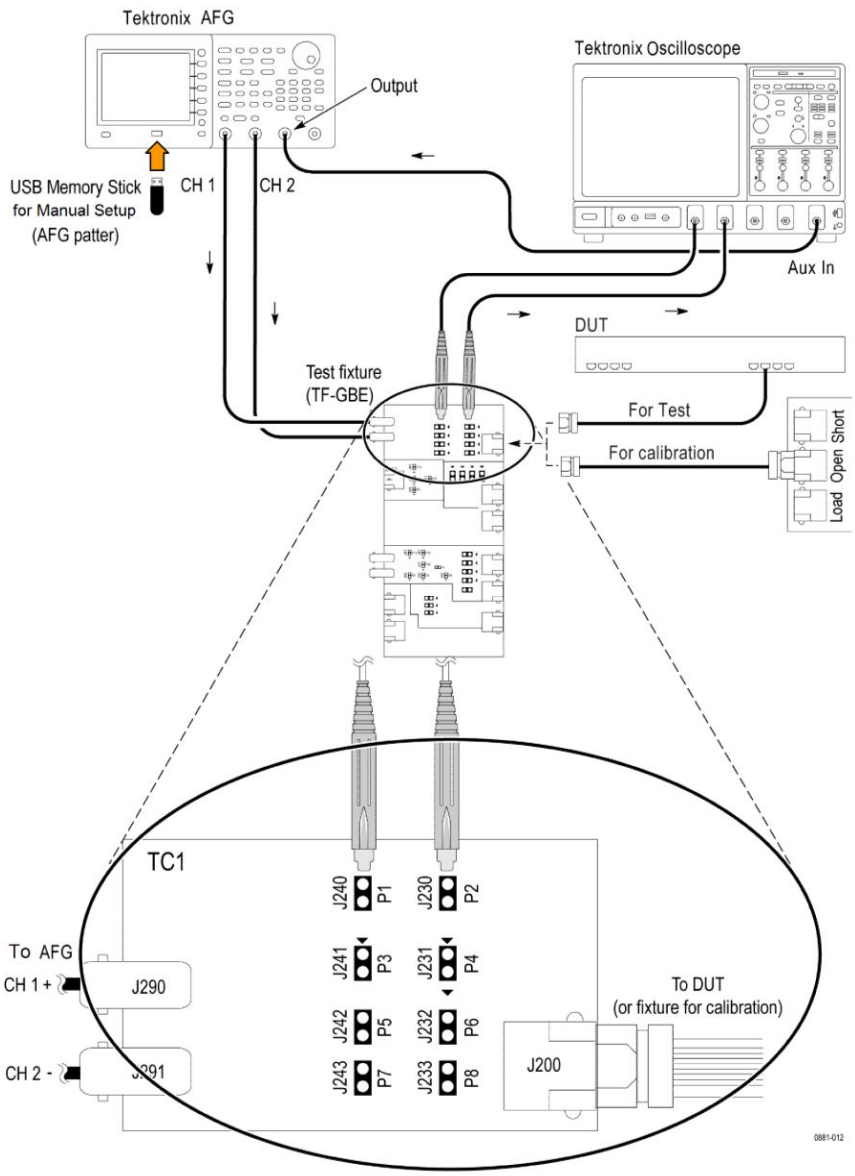
NOTE.

- *Make sure "+" on the probe tip aligns with " < " on the text fixture board. This takes care of the polarity being not reversed.*
- *Set the attenuator factor to 1X on the probe.*
- *Step 8 and 9 are applicable and followed for Return Loss Load, Short and Open.*
- *Performing Calibration is recommended before executing the Return Loss measurement, and make sure that you click the* ***Apply*** *button on the Calibration tab for execution of Return Loss measurement.*
- *Make sure differential high impedance probe heads and probe wires should not touch each other as it effects the Calibration results in Return Loss measurement.*
- *During calibration, the signal source AFG/AWG will be setup automatically from the application, by loading the disturber pattern.*
- *When you click on the Start button, the AFG/AWG does not setup, only Oscilloscope will setup because the application will not disturb the calibration done using AFG/AGW.*

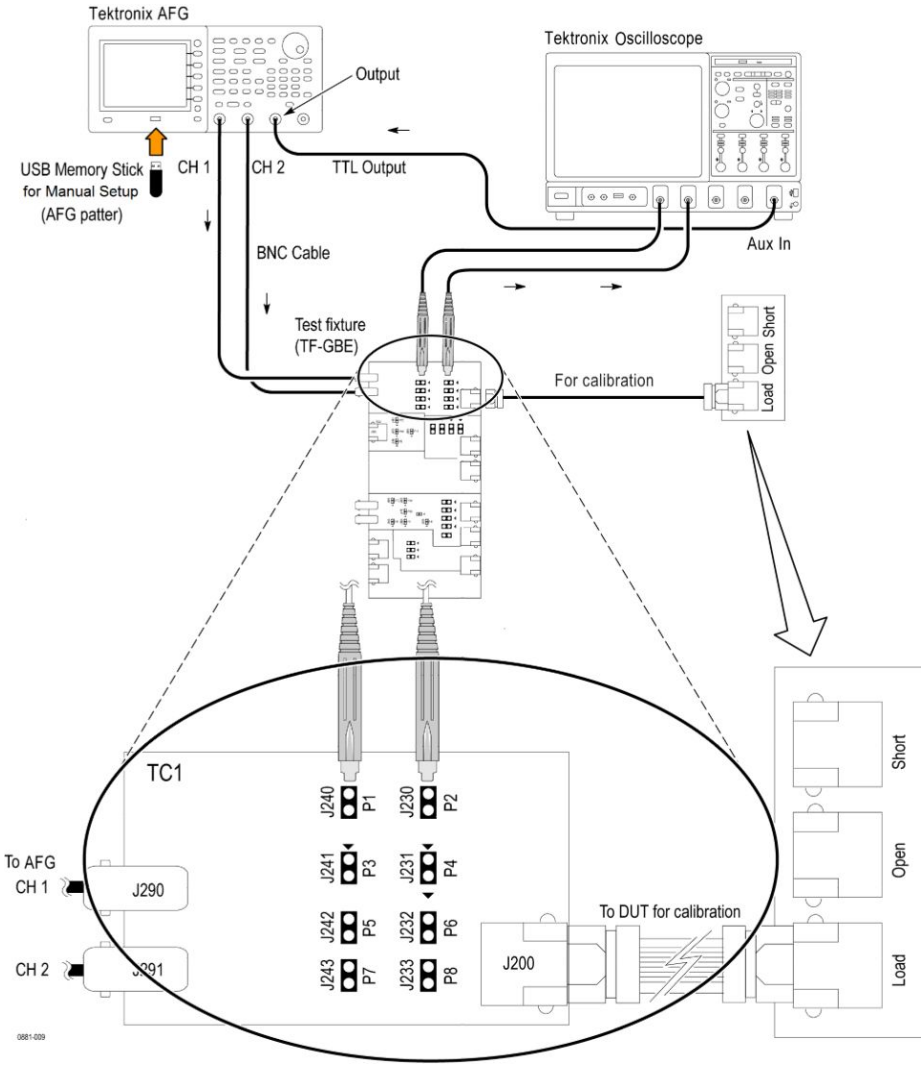
Connection diagram for Return Loss test (AWG)



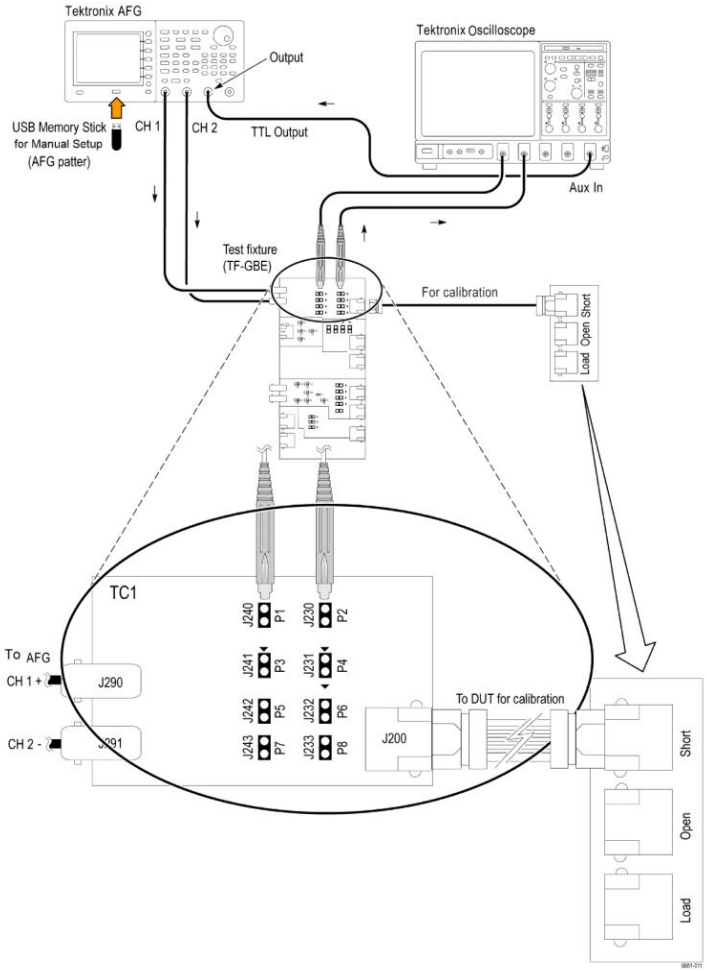
Connection diagram for Return Loss test (AFG)


Tektronix AFG
Output
Tektronix Oscilloscope
USB Memory Stick
for Manual Setup
(AFG patter)
CH 1
CH 2
Aux In
DUT
Test fixture
(TF-GBE)
For Test
For calibration
Open
Short
Load
TC1
J240
P1
J230
P2
J241
P3
J231
P4
J242
P5
J232
P6
J243
P7
J233
P8
To AFG
CH 1 +
J290
CH 2 -
J291
J200
To DUT
(or fixture for calibration)
0881-012

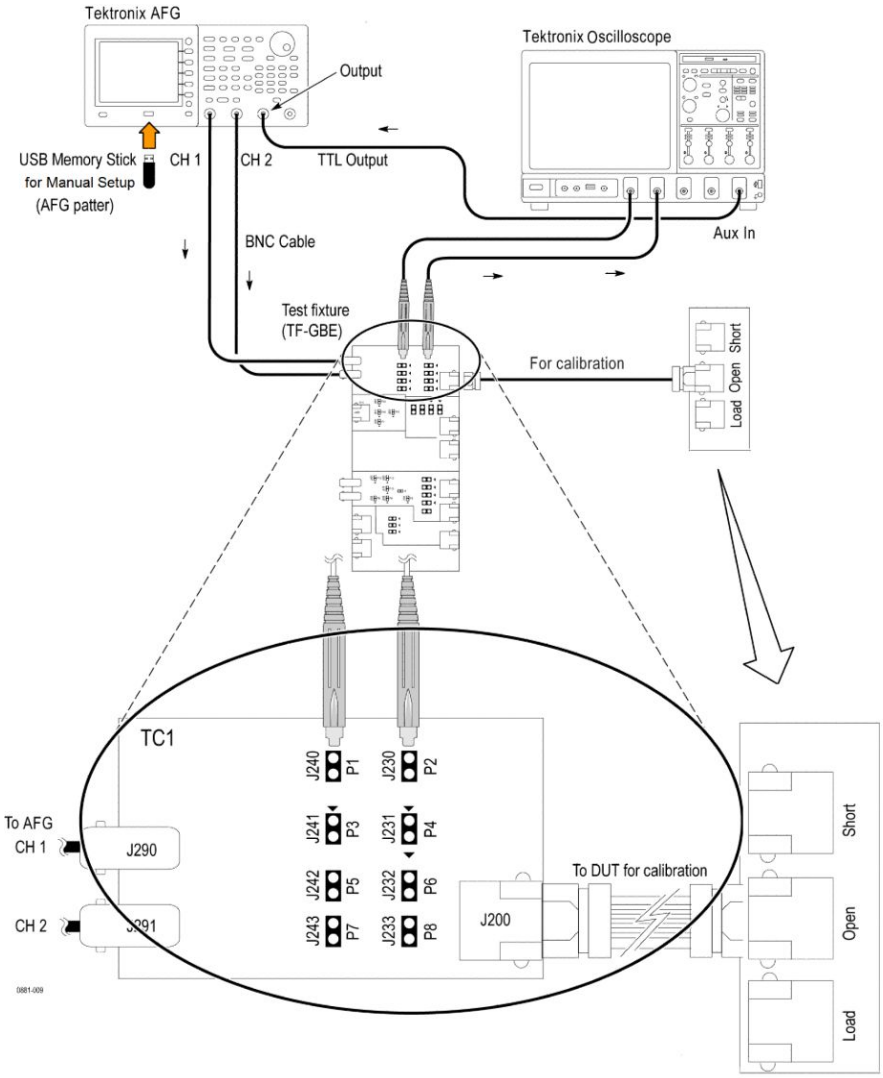
Return Loss Calibration-Load



Return Loss Calibration-Short



Return Loss Calibration-Open



AFG configuration

In Return loss, there is no automatic control of AFG. You have to manually load the required pattern from C:\Program Files\Tektronix \TekExpress\TekExpress BroadR-Reach\AWG Waveforms\Return Loss\AFG3000.

Following parameters are set in AFG, when you recall the setup file:

1. Go to the Calibration tab of the Return Loss and click on the link given at the bottom of the *Calibration* tab.
2. Click **RUN** button, which automatically transfer the Return Loss pattern to AFG/AWG.

   The pattern is set to arbitrary.
3. There are three *Calibration* Runs mainly Load, Open, and Short . For each of these Runs, make sure that the AFG is loaded with the pattern as mentioned in the above path before clicking the

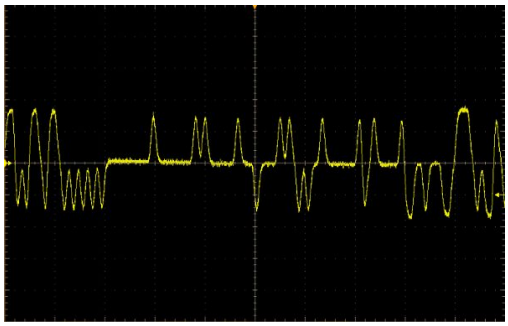
button.

NOTE. *For manual setup of AFG, Copy the RL1000_AFG3000.TFS and RL1000_AFG3000.tfw (AFG setup file) files to the USB memory and recall on AFG.*

Measurement setup and algorithm

The Return Loss measurement is divided into the following three steps:

1. Return loss calibration
2. Computation of error coefficients
3. Computation of cable return loss

Item		Requirements
Configuration parameters		1. Smoothing factor
Signal type		Test mode 4 
Measurement algorithm outputs		Return loss in dB for frequency ranging from 3 MHz to 100 MHz
	Calibration	Gamma_Open, Gamma_Short and Gamma_Load corresponding to open, short and load terminations
	Computation of coefficients	Error coefficients a, b, and c

Item		Requirements
	Computation of cable loss return	1. Return_loss_100 Ohm loads
Measurement algorithm inputs		
	Calibration	Differential waveforms captured at probe point P1 and P2 Load Type
	Computation of coefficients	Gamma_Open, Gamma_Short and Gamma_Load computed during calibration
	Computation of cable loss return	Differential waveforms captured at probe point P1 and P2 Load Type
Limits		Lower limit: NA Upper limit: Return Loss (f) = 20 (in dB) for f = 1 to 30 MHz = 20 – 20 * $\log_{10}$(f/30) (in dB) for f = 30 to 66 MHz
Plots		A plot containing return loss versus frequency and mask obtained with above limits

Measurement Algorithm

Calibration. The calibration procedure is required to correct for probe and fixture loading in the final measured result. Calibration is done using three loads: open, short, and 100 Ω load.

BroadR-Reach Automated Compliance Solution automatically executes the calculations described below for the return loss calibration.

1. Multiply the acquired signals using Gaussian window (convolving in frequency domain) and then compute the reflected and incident voltages.

   Reflected voltage = (2*P1 – P2)

   Incident voltage = P1

2. Return loss is computed as follows:

   Return loss = FFT(Reflected voltage)/ FFT(Incident voltage)

   This return loss is computed for three types of loads: Open, Short, 100 Ω load. Let the return loss computed be Gamma_Open, Gamma_Short, and Gamma_Load for open, short, and 100 Ω load, respectively. The Gamma_Open, Gamma_Short,and Gamma_Load are dumped in csv files.

3. For plotting purpose, Gamma_Open, Gamma_Short, and Gamma_Load are converted to dB scale, interpolated, and smoothed. Interpolated and smoothed Gamma_Open, Gamma_Short, and Gamma_Load are plotted for visual representation.

Computing error coefficients. BroadR-Reach Automated Compliance Solution automatically executes the calculations described below for the return loss error coefficient calculation.

1. Gamma_Open, Gamma_Short, and Gamma_Load are used for computing the following three coefficient that are required for computing cable return loss:
   a. b – Gamma_Short + c * Gamma_Short
   b. Gamma_Load
   c. (Gamma_Short + Gamma_Open – 2 * b) / (Gamma_Short – Gamma_Open)

Computing return loss. BroadR-Reach Automated Compliance Solution automatically executes the calculations described below for the return loss.

1. Multiply the acquired signals using Gaussian window (convolving in frequency domain) and then compute the reflected and incident voltages.

   Reflected voltage = (2*P1 – P2)

   Incident voltage = P1
2. Return loss is computed as follows:

   Return loss = FFT(Reflected voltage)/ FFT(Incident voltage)

   Let the computed return loss be GammaValue.
3. Read the error coefficients a, b, and c from the dumped csv files. Compute the corrected return loss using a, b, and c.

   Corrected Return loss = (GammaValue – b)/(a – c * GammaValue)

   This return loss is for 100 Ω load. Let this be return_loss_100.
4. For plotting purposes, Return_Loss_100 is converted to dB scale, interpolated, and smoothed. Interpolated and smoothed Return_Loss_100 is plotted for visual representation.

   Return_loss_100 is compared with the specification given limit.

Hints when Return Loss fails

Make sure you:

- Performed Scope SPC calibration, made sure it passed.
- Used 6" Ethernet interconnect cable, a different one than earlier tests.
- Ran with default configuration (like 200 avg count).

See also *Plots*

Know limitations

Known Limitations

The following are the know limitations:

1. Return Loss and Transmitter Distortion measurements needs Calibration before RUN. Calibration is recommended when there is a change in the setup.
2. In Return Loss measurement, when you select 'Use pre-recorded waveform' option for Calibration for the first time, then an error message is displayed as 'Calibration files not present'. It is recommended to RUN the live Calibration before you select the pre-recorded option.
3. When you recall the Return Loss measurement session file, the Calibration files are not recalled. You need to set/load the Calibration files manually.
4. You cannot stop or pause Calibration RUN during its execution. (This is applicable for both Return loss and Transmitter Distortion measurement)
5. The Plots are different when you RUN the Return Loss in pre-recorded mode for the first time, when compared to subsequent RUNS.

   Wait (max of 3-4 minutes) till the **RUN** button is enable to use. Do not press **RUN** button till application is ready.
6. When the application is not able to work with LAN connection for AFG, use USB-GPIB cable to connect AFG to scope. Make sure instrument refresh is performed in the TekExpress application, before configuring signal source in the Global settings tab.

Measurement Error Messages

Measurement Error Messages

The following table lists all of the error messages associated with BRR measurements and their definitions.

Table 10: Measurement error messages

Error message	Description
Not Enough Edges in the Waveform. Acquire the waveform for a longer duration.	This error will occur If the acquired signal does not have enough rise to fall and fall to rise transitions.
Data points acquired is insufficient for moving average filtering. Set the start frequency, stop frequency and RBW such that (stop frequency - start frequency)/RBW is greater than 3.	User should check the start frequency, stop frequency and RBW setting. This error will occur If the waveform does not have enough frequencies between the start and stop frequency.
Captured Signal has less than 3 segments, Please increase Record Length.	This error will occur If the acquired signal has less than three segments of test mode 4 signal (2047 bits).
Signal Validation failed for Test mode 2. Make sure that Unit interval/frequency of the signal does not deviate beyond ± 500 ppm from 66(2/3) MHz.	This error will occur if the frequency of the acquired test mode 2 signal is offset by ± 500 ppm from 66(2/3) MHz.
Signal Validation failed for Test mode 2. Make sure that input signal has Rise to rise period deviation of less than 10%.	This error will occur if the acquired signal does not meet the requirements given in specification for test mode 2 signal.
Signal Validation failed for Test mode 2. Make sure that input signal has Fall to fall period deviation of less than 10%.	This error will occur if the acquired signal does not meet the requirements given in specification for test mode 2 signal.
Signal Validation failed for TX_TCLK signal. Make sure that input signal has Rise to rise period deviation of less than 10%.	This error will occur if the acquired signal does not meet the requirements given in specification for TX_TCLK signal.
Signal Validation failed for TX_TCLK signal. Make sure that input signal has Fall to fall period deviation of less than 10%.	This error will occur if the acquired signal does not meet the requirements given in specification for TX_TCLK signal
Signal Validation failed for Test mode 1. Make sure that input signal has 1. Edge to edge period deviation less than 10% 2. At least 2 Unit Intervals	This error will occur if the acquired signal does not meet the requirements given in specification for test mode 1 signal.

Error message	Description
Signal Validation failed for test mode 4. Make sure that input signal 1. Is PAM3 modulated 2. Has at least 3 Test mode 4 frames (2047 bits). 3. The common possible cause could be polarity of the probing points being reversed. Make sure Probe tip '+' aligns with '<' on the test fixture board.	This error will occur if the acquired signal does not meet the requirements given in specification for test mode 4 signal.
Signal Validation failed for Test mode 4. Make sure that Unit interval/frequency of the signal does not deviate beyond ± 500 ppm from 66(2/3) MHz.	This error will occur if the acquired signal does not meet the requirements given in specification for test mode 4 signal.
Input signal is an invalid TX_TCLK signal. Make sure that edge to edge deviation is less than 10%.	This error will occur if the acquired signal does not meet the requirements given in specification for TX_TCLK signal.
Unable to run because return loss measurement is not calibrated. Calibrate and apply the return loss measurement and then click on start.	This error will occur if user tries to execute return loss measurement without applying calibration (computation of error coefficients).
Invalid Signal at CH1. Please check DUT connections and re-run the test.	This error will occur if an invalid signal is fed as input to return loss measurement (wrong connection).
Calibration files are not present for return loss measurement. Either Deselect the Measurement or Stop the Execution, Perform Calibration and Re-Run the test.	This error will occur if return loss measurement is executed without calibration files. User has to always run the calibration and then execute return loss measurement. If calibration is done, return loss measurement uses latest available calibration files.

TekExpress Programmatic Interface

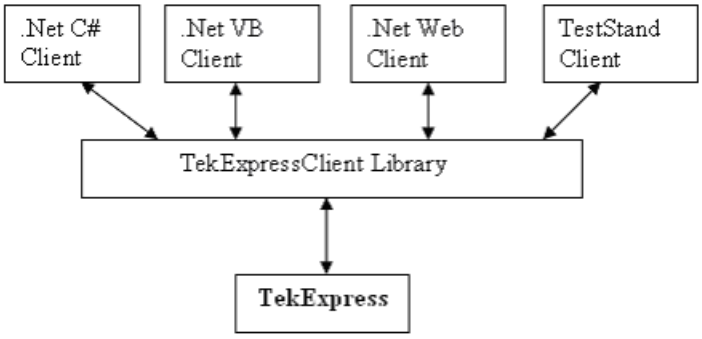
About the Programmatic Interface

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

References Required

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

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

Required Steps for a Client

The following are the steps that a client needs to follow to use TekExpressClient.dll to programmatically control the server:

A client UI must be developed to access the interfaces exposed through the server. This client needs to load 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 must do the following to run a remote operation:

1. The client needs to provide the IP address of the PC at which the server is running in order to connect to the server.
2. The client needs to lock 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. Note that this does not lock the UI.

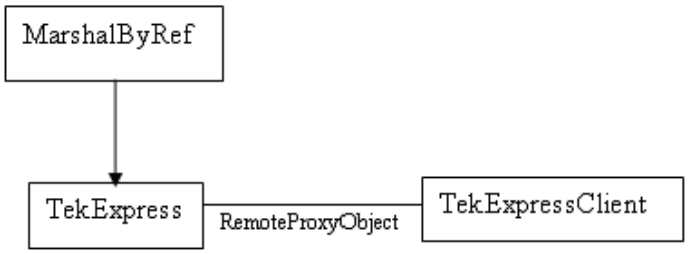
   If any other client tries to access a server that is locked, it will get a notification that the server is locked by another client.
3. When the client has connected to and locked the server, the client can access any of the programmatic controls to run the remote automations.
4. After the client operations are completed, the server needs to be unlocked by the client.

See Also

About BRR Application Commands

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.

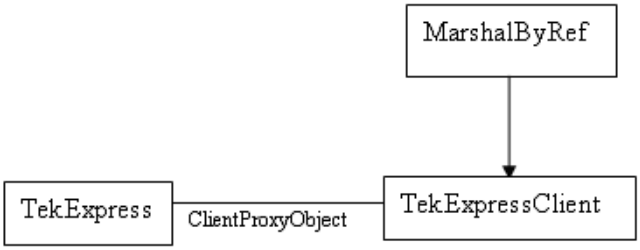
For example,

//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.

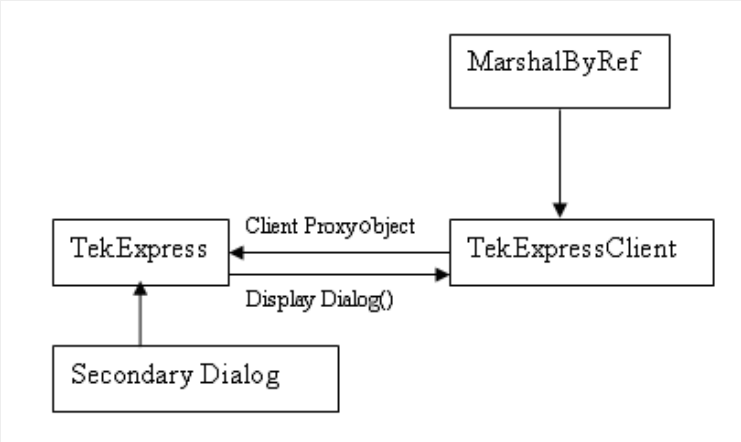
Examples

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

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

For more information, click the topic links listed below.

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.

Program Examples

The following program examples show how to communicate between a PC and TekExpress BRR remotely, using typical steps.

For detailed information about each command, see the *BRR Application Commands* section.

Table 11: Remote access code example 1

Task	Code
Start the application	
Connect through an IP address.	{'Set String Details string devicename = ”BRR” string suitename = ”BRR” m_Client.Connect("localhost")'True or False clientID = m_Client.getClientID }
Lock the server	m_Client.LockServer(clientID)
Disable the Popups	m_Client.SetVerboseMode(clientID, false)
Set the DUT ID	m_Client.SetDutId(clientID, "DUT_Name")

Task	Code

Select channels	m_Client.SetGeneralParameter(clientID, devicename, suitename, Testname, "Source Data$CH1");
	m_Client.SetGeneralParameter(clientID, devicename, suitename, Testname, "Source Data$CH2");
	m_Client.SetGeneralParameter(clientID, devicename, suitename, Testname, "Probe2$C2");"This Probe2 is for selecting Source 2 for return Loss"

Select a measurement	m_Client.SelectTest(clientID, devicename, suitename, "Return Loss", True)

Configure the selected measurement (Acquire Parameters)	m_client.SetAcquireParameter(clientID, devicename, suitename,"Transmit Clock Frequency","Average$60");
	m_client.SetAcquireParameter(clientID, devicename, suitename," Transmit Clock Frequency","Record Length$10");
	m_client.SetAcquireParameter(clientID, devicename, suitename," Transmitter Distortion","TX_TCLK$Included");
	m_client.SetAcquireParameter(clientID, devicename, suitename," Transmitter Distortion","Average$100");
	m_client.SetAcquireParameter(clientID, devicename, suitename," Transmitter Distortion","Disturbing Signal$False");
	m_client.SetAcquireParameter(clientID, devicename, suitename," Transmitter Distortion","High Resolution$50");

Configure the selected measurement (Analyze Parameters)	m_client.SetAnalyzeParameter(clientID, devicename, suitename," Transmitter Distortion","LP Filter$Included") m_client.SetAnalyzeParameter(clientID, devicename, suitename," Transmitter Distortion","RBW$50") m_client.SetAnalyzeParameter(clientID, devicename, suitename," Transmitter Distortion","Center Frequency$75") m_client.SetAnalyzeParameter(clientID, devicename, suitename," Transmitter Distortion","Frequency Span$200")
Run with set configurations	m_Client.Run(clientID)

Task	Code
Wait for the test to complete	Do Thread.Sleep(500) m_Client.Application_Status(clientID) Select Case status Case "Wait" 'Get the Current State Information mClient.GetCurrentStateInfo(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxButtontexts) 'Send the Response mClient.SendResponse(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxResponse) End Select Loop Until status = "Ready"
After the test is complete	'Save all results values from folder for current run m_Client.TransferResult(clientID, logDirname) 'Save all waveforms from folder for current run m_Client.TransferWaveforms(clientID, logDirname) 'Save all images from folder for current run m_Client.TransferImages(clientID, logDirname)
Unlock the server	m_Client.UnlockServer(clientID)
Disconnect from server	m_Client.Disconnect()
Exit the application	

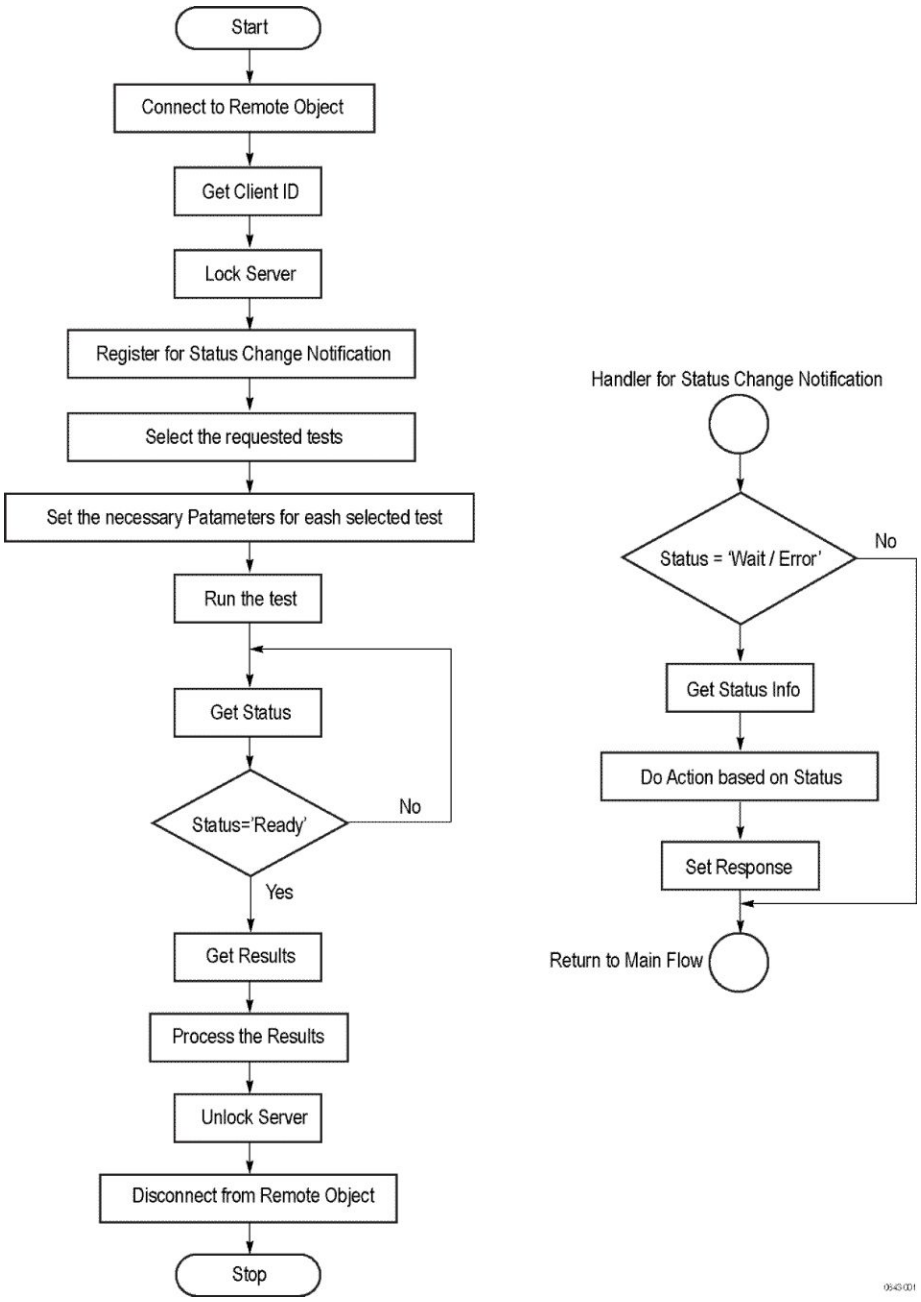
Table 12: Remote access code example 2

Task	Code
Start the application	
Connect through an IP address.	m_Client.Connect("localhost") 'True or False clientID = m_Client.getClientID
Lock the server	m_Client.LockServer(clientID)
Disable the Popups	m_Client.SetVerboseMode(clientID, false)
Set the DUT ID	m_Client.SetDutId(clientID, "DUT_Name")
Select a test	mClient.SelectsingleTest(clientID, "BRR", "BRR", “Spec 1.0”, "Transmitter Distortion", true)
Set Disturbing Signal	mClient.SetAcquireParameter(clientID, "BRR", "BRR", "Transmitter Distortion","Disturbing Signal$False")
Set Record Length	mClient.SetAcquireParameter(clientID, "BRR", "BRR", "Transmitter Distortion","Record Length $10")
Run with set configurations	m_Client.Run(clientID)

Task	Code
Wait for the test to complete.	Do Thread.Sleep(500) m_Client.Application_Status(clientID) Select Case status Case "Wait"
Get the current state information	mClient.GetCurrentStateInfo(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxButtontexts)
Send the response	mClient.SendResponse(clientID, WaitingMsbBxCaption, WaitingMsbBxMessage, WaitingMsbBxResponse) End Select Loop Until status = "Ready"
Save results	Save all results values from folder for current run m_Client.TransferResult(clientID, logDirname)
Unlock the server	m_Client.UnlockServer(clientID)

Client Programmatic Interface Example

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



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

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

---

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

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

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

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

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

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

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

Handler of Status Change Notification

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

See Also

About BRR Application Commands

Program Example

BRR Application Commands

About TekExpress BRR Application Commands

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

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 measurement

Select a suite

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 check if a session is saved

Unlock the server

Disconnect from server

Connect Through an IP Address

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

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

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

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

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

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

string ipAddress

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

out string clientID

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

NOTE.

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.

Lock the Server

NOTE. *This method does not lock the UI, but you need this method to set the value that gives the status of the operation after it has been performed.*

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

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

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

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

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

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

out string clientID

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

NOTE. *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.*

Disable the Popups

Command name	Parameters	Description	Return value	Example

SetVerboseMode()	string clientID
	bool _verbose
		This method sets the verbose mode to either True or False.
		When the value is set to True, any of the message boxes appearing during the application run will be routed to the client machine that is controlling TekExpress.
		When the value is set to False, then all the message boxes are shown on the server machine.
			String that displays the status of the operation after it has been performed.
			When Verbose mode is set to True, the return value is “Verbose mode turned on. All dialog boxes will be shown to client ...”.
			When Verbose mode is set to False, the return value is “Verbose mode turned off. All dialog boxes will be shown to server ...”.
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				Verbose mode is turned on
				return=m_Client.SetVerboseMode(clientID, True)
				Verbose mode is turned off
				returnval=m_Client.SetVerboseMode(clientID, False)

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...".

out string clientID

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

bool_verbose

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

Set or Get the DUT ID

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

out string clientID

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

string dutName

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

string dutId

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

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

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

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

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

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

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

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

Set the Configuration Parameters for a Suite or Measurement

Command name	Parameters	Description	Return value	Example

SetGeneralParameter	string clientID
	string device
	string suite
	string test
	string parameterString
		This method sets the general parameters.	String that displays the status of the operation after it has been performed.
			The return value is "" (an empty String) on success.
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				returnval = m_Client.SetGeneralParameter(clientID, devicename, suitename, string.Empty, parameterstring);

SetAnalyzeParameter()	string clientID
	string device
	string suite
	string test
	string parameterString
		This method sets the parameter values in the Ref Levels and Clock Settings tabs in the test configuration section.	String that displays the status of the operation after it has been performed.
			The return value is "" (an empty String) on success.
				mClient = new Client() //m_Client is a reference to the Client class in the Client DLL
				returnval as string
				Select Analyze parameter:
				returnval = mClient.SetAnalyzeParameter(clientID, devicename, suitename, test, parameterstring)

SetAcquireParameter()	string clientID
	string device
	string suite
	string test
	string parameterString
		This method sets the parameter values in the Vertical Setup and the Scope Settings tabs in the test configuration section.	String that displays the status of the operation after it has been performed.
			The return value is ” (an empty string) on success.
				mClient = new Client() //m_Client is a reference to the Client class in the Client DLL
				returnval as string
				Set Acquire Parameter:
				returnval = mClient.SetAcquireParameter(clientID, devicename, suitename, test, parameterstring)

out string clientID

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

string device

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

string suite

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

string test

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

string parameterString

Name	Type	Direction	Description
parameterString	string	IN	Selects or deselects a test

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 the Configuration Parameters for a Suite or Measurement

Command name	Parameters	Description	Return value	Example

GetGeneralParameter()	string clientID
	string device
	string suite
	string test
	string parameterString
		This method gets the general configuration parameters for a given suite or measurement.	The return value is the general configuration parameter for a given suite or measurement that is set.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string

GetAnalyzeParameter()	string clientID
	string device
	string suite
	string test
	string parameterString
		This method queries the parameter values in the Ref Level and Clock Settings tabs in the test configuration section.	The return value is the configuration parameter for a given suite or measurement.	mClient = new Client() //m_Client is a reference to the Client class in the Client DLL
				returnval as string
				Get Analyze parameter:
				returnval = mClient.GetAnalyzeParameter(clientID, devicename, suitename, test, parameterstring)
				GetAnalyzeParameter Examples

GetAcquireParameter()	string clientID
	string device
	string suite
	string test
	string parameterString
		This method queries the parameter values in the Vertical Setup and Scope Settings tabs in the test configuration section.	The return value is the configuration parameter for a given suite or measurement.	mClient = new Client() //m_Client is a reference to the Client class in the Client DLL
				returnval as string
				Get Acquire Parameter:
				returnval = mClient.GetAcquireParameter(clientID, devicename, suitename, test, parameterstring)
				GetAcquireParameter Examples

out string clientID

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

string device

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

string suite

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

string test

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

string parameterString

Name	Type	Direction	Description
parameterString	string	IN	Selects or deselects a test

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...".

GetAcquireParameter Examples

This example uses BRR test Transmitter Distortion.

Parameter	Example
TX_TCLK	returnval = mClient.GetAcquireParameter(clientID, devicename, suitename,"Transmitter Distortion","Transmitter Distortion Acquisition $TX_TCLK")
Average	returnval = mClient.GetAcquireParameter(clientID, devicename, suitename,"Transmitter Distortion","Transmitter Distortion Acquisition $Average")
Disturbing Signal	returnval = mClient.GetAcquireParameter(clientID, devicename, suitename,"Transmitter Distortion","Transmitter Distortion Acquisition $Disturbing Signal")
Hi Resolution	returnval = mClient.GetAcquireParameter(clientID, devicename, suitename,"Transmitter Distortion","Transmitter Distortion Acquisition$Hi Resolution")
Probe Point Amplitude in V	returnval = mClient.GetAcquireParameter(clientID, devicename, suitename,"Transmitter Distortion","Transmitter Distortion Acquisition $Probe Point Amplitude in V")
Distortion Amplitude in V	returnval = mClient.GetAcquireParameter(clientID, devicename, suitename,"Transmitter Distortion","Transmitter Distortion Acquisition $Distortion Amplitude in V")
Distortion Frequency in MHz	returnval = mClient.GetAcquireParameter(clientID, devicename, suitename,"Transmitter Distortion","Transmitter Distortion Acquisition $Distortion Frequency in MHz")
DUT Amplitude in V	returnval = mClient.GetAquireParameter(clientID, devicename, suitename,"Transmitter Distortion","Transmitter Distortion Acquisition $DUT Amplitude in V")

GetAnalyzeParameter Examples

This example uses BRR test Transmitter Distortion.

Parameter	Example
LP Filter	returnval =mClient.GetAnalyzeParameter(clientID, devicename, suitename,"Transmitter Distortion","LP Filter")

Select a Measurement

Command name	Parameters	Description	Return value	Example
SelectTest()	string clientID string device string suite string test bool isSelected	This method selects or deselects a given test.	String that displays the status of the operation after it has been performed. The return value is "" (an empty String) on success.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL returnval as string **Select test: (test name)** returnval=mClient. SelectTest(clientID , devicename, suitename, testname, isSelected);) Select a specific test.

Select a Specific Test

Test	Command

Transmit Clock Frequency	For selecting the test:
	returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmit Clock Frequency", True);
	For deselecting the test:
	returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmit Clock Frequency", False);

Transmitter Master Timing Jitter	For selecting the test: returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmitter Timing Jitter - Master Jitter", True); For deselecting the test: returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmitter Timing Jitter - Master Jitter", False)
Transmitter Slave Timing Jitter	For selecting the test: returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmitter Timing Jitter - Slave Jitter", True); For deselecting the test: returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmitter Timing Jitter - Slave Jitter", False)

Test	Command
Transmitter Output Droop	For selecting the test: returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmitter Output Droop", True); For deselecting the test: returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmitter Output Droop", False)

Transmitter Power Spectral Density	For selecting the test:
	returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmitter Power Spectral Density", True);
	For deselecting the test:
	returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmitter Power Spectral Density", False);

Transmitter Distortion	For selecting the test:
	returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmitter Distortion", True);
	For deselecting the test:
	returnval=m_Client.SelectTest(clientID, devicename, suitename, "Transmitter Distortion", False);

Return Loss	For selecting the test:
	returnval=m_Client.SelectTest(clientID, devicename, suitename, "Return Loss", True);
	For deselecting the test:
	returnval=m_Client.SelectTest(clientID, devicename, suitename, "Return Loss", False);

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...".

out string clientID

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

string device

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

string suite

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

string test

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

bool isSelected

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

Select a Suite

Command name	Parameters	Description	Return value	Example

SelectSuite()	string clientID
	string device
	string suite
	bool isSelected
		This method selects or deselects a given suite.
		Setting parameter is selected to True, you can select a suite.
		Setting parameter is selected to False, you can deselect a suite.
			String that gives the status of the operation after it has been performed.
			The return value is "" (an empty String) on success.
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				Select Suite (Default):
				returnval=m_Client.SelectTest(clientID, "BRR", "BRR", True)

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...".

out string clientID

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

string device

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

string suite

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

bool isSelected

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

Run with Set Configurations or Stop the Run Operation

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

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

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

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

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

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

out string clientID

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

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

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

Get or Set the Timeout Value

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

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

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

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

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

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

out string clientID

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

string time

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

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

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

Wait for the Test to Complete

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

Command name	Parameters	Description	Return value	Example
ApplicationStatus()	string clientID	This method gets the status of the server application. The states at a given time are Ready, Running, Paused, Wait, or Error.	String value that gives the status of the server application.	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)

QueryStatus()	string clientID
	out string[] status
		This is an interface for the user to transfer Analyze panel status messages from the server to the client.	String that gives the status of the operation after it has been performed.
			On success the return value is “Transferred...”.
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnVal=m_Client.QueryStatus(clientID, out statusMessages)
				if ((OP_STATUS)returnVal == OP_STATUS.SUCCESS)
				return “Status updated...”
				else
				return CommandFailed(returnVal)

GetCurrentStateInfo() ***NOTE.*** *This command is used when the application is running and is in the wait or error state.*	string clientID out string caption out string message out string[] buttonTexts	This method gets the additional information of the states when the application is in Wait or Error state. Except client ID, all the others are out parameters.	This command does not return any value. This function fills up the out parameters that are passed when invoking this function.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. m_Client.GetCurrentStateInfo(clientID, caption,message, buttonTexts)

Command name	Parameters	Description	Return value	Example
SendResponse() ***NOTE.*** *This command is used when the application is running and is in the wait or error state.*	string clientID out string caption out string message string response	After receiving the additional information using the method GetCurrentStateInfo(), the client can decide on the response to send and send the response to the application using this function. The response should be one of the strings that was earlier received 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.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. m_Client.SendResponse(clientID, caption,message, response)

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 NOT FOUND and the message displayed is "Server not found...Disconnect!".

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

Ready: Test configured and ready to start

Running: Test running

Paused: Test paused

Wait: A popup that needs your inputs

Error: An error has occurred

out string clientID

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

out string[] status

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

out string caption

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

out string message

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

out string[] buttonTexts

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

string response

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

After the Test is Complete

Command name	Parameters	Description	Return value	Example
GetPassFailStatus()	string clientID string device string suite string test	This method gets the pass or fail status of the measurement after test completion. ***NOTE.*** *Execute this command after completing the measurement.*	String that gives the status of the operation after it has been performed. Returns the pass or fail status in the form of a string.	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, ”Transmitter Output Droop”) // Pass or Fail
GetResultsValue()	string clientID string device string suite string test string parameterString	This method gets the result values of the measurement after the run.	String that gives the status of the operation after it has been performed. Returns the result value in the form of a string.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as srting returnval=m_Client.GetResultsValue(clientID, "BRR", "BRR", "Transmitter Output Droop", "Measured Value")

Command name	Parameters	Description	Return value	Example

GetResultsValueForSubMeasurements()	string clientID
	string device
	string suite
	string test
	string parameterString
	int rowNr
		This method gets the result values for individual sub-measurements, after the run.	String that gives the status of the operation after it has been performed. Returns the result value in the form of a string.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				Single-ended Low Level Voltages DP (nS)
				returnval=m_Client.GetResultsValueForSubMeasurements(clientID, "BRR", "BRR", "Transmitter Output Droop", "Transmitter Output Droop", "Measured Value", 0) //For DP wfm
				returnval=m_Client.GetResultsValueForSubMeasurements(clientID, "BRR", "BRR", "Transmitter Output Droop", "Measured Value", 1) //For DN wfm

Command name	Parameters	Description	Return value	Example

GetReportParameter()	string clientID
	string device
	string suite
	string test
	string parameterString
		This method gets the general report details such as Oscilloscope model, TekExpress version, and BRR version.	The return value is the Oscilloscope model, TekExpress version, and BRR version.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				Oscilloscope Model
				returnval=m_Client.GetReportParameter(clientID,"Scope Model")
				TekExpress Version
				returnval=m_Client.GetReportParameter(clientID,"TekExpress Version")
				BRR Version
				returnval=m_Client.GetReportParameter(clientID,"Application Version")

TransferReport()	string clientID
	string filePath
		This method transfers the report generated after the run.
		The report contains the summary of the run.
		The client must provide the location where the report is to be saved at the client-end.
			String that gives the status of the operation after it has been performed.
			Transfers all the result values in the form of a string.
				m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL.
				returnval as string
				returnval=m_Client.TransferReport(clientID,"C:\Report”)

Command name	Parameters	Description	Return value	Example
TransferWaveforms()	string clientID string filePath	This method transfers all the waveforms from the folder for the current run. ***NOTE.*** *For each click of Run button, a folder is created in the X: drive. Transfer the waveforms before clicking the Run button.*	String that gives the status of the operation after it has been performed. Transfers all the waveforms in the form of a string. On success the return value is “Transferred...”.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.TransferWaveforms(clientID,"C:\Waveforms”)
TransferImages()	string clientID string filePath	This method transfers all the images (screenshots) from the folder for the current run (for a given suite or measurement). ***NOTE.*** *For each click of Run button, a folder is created in the X: drive. Transfer the waveforms before clicking the Run button.*	String that gives the status of the operation after it has been performed. Transfers all the images in the form of a string.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.TransferImages(clientID, "C:\Waveforms”)

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

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

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

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

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

out string clientID

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

string device

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

string suite

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

string test

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

string parameterString

Name	Type	Direction	Description
parameterString	string	IN	Selects or deselects a test

int rowNr

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

string filePath

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

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

Save Recall or Check if a Session is Saved

Command name	Parameters	Description	Return value	Example
CheckSessionSaved()	string clientID out bool saved	This method is called when a check is to be made to know if the current session is saved.	Return value is either True or False.	m_Client = new Client() //m_Client is a reference to the Client class in the Client DLL. returnval as string returnval=m_Client.CheckSessionSaved(m_clientID, out savedStatus)
RecallSession()	string clientID string name	Recalls a saved session. The name of the session is provided by the client.	String that gives the status of the operation after it has been performed. The return value is “Session Recalled...”.	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)
SaveSession()	string clientID string name	Saves the current session. The name of the session is provided by the client.	String that gives the status of the operation after it has been performed. The return value is "Session Saved..."/"Failed...".	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)
SaveSessionAs()	string clientID string name	Saves the current session in a different name every time this method is called. The name of the session is provided by the client.	String that gives the status of the operation after it has been performed. The return value is "Session Saved...".	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)

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...".

out string clientID

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

string name

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

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

string name

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

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

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

string name

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

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

out bool saved

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

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

Unlock the Server

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

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

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

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

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

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

out string clientID

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

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

Disconnect from the Server

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

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

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

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

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

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

out string clientID

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

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

Handle Error Codes

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

Value	Code	Description
FAIL	−1	The operation failed.
SUCCESS	1	The operation succeeded.
NOTFOUND	2	Server not found
LOCKED	3	The server is locked by another client, so the operation cannot be performed.
UNLOCK	4	The server is not locked. Lock the server before performing the operation.
NULL	0	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...".

Index

A

Acquire Status, 49
Acquisition source
    selecting live waveforms as, 29
Acquisitions, 26
Activating the license, 11
Analysis Status, 49
Appending reports, 61
ApplicationStatus(), 137
Auto Scroll, 49
AWG/AFGs supported, 7

B

BRR Application Commands, 119

C

Calibration, 40
CheckSessionSaved(), 146
Clear Log, 49
Clearing test results displayed, 59
Client Proxy Object, 112
Clock Settings, 37
Commands
    ApplicationStatus(), 137
    GetCurrentStateInfo(), 137
    QueryStatus(), 137
    selecting a channel, 113
    SendResponse(), 137
Commands List, 119
Compliance Mode, 33, 36
Configuring tests, 33
Connect(), 119
Connected instruments
    selecting, 33
    viewing, 15

D

Default directory, 3
Default test setup, 65
Directories, 3
Disconnect(), 149
DSP Filter
    8 GHz, 33
DUT settings, 21

E

Edge Trigger Level, 26
Email Notifications, 47
Email Settings, 14
Exiting the application, 17

F

File name extensions, 6
Fixture supported, 8
Fixtures, 8
Flowchart for Client Programmatic Interface, 116

G

GetAcquireParameter(), 127
GetAnalyzeParameter(), 127
GetCurrentStateInfo(), 137
GetDutId(), 123
GetGeneralParameter(), 127
GetPassFailStatus(), 140
GetReportParameter(), 142
GetResultsValue(), 140
GetResultsValueForSubMeasurements(), 141
GetTimeOut(), 135

I

Installing the software, 10
Instrument Control Settings, 14
Instrument Control Settings dialog box, 15
Instruments
    searching for, 14
    selecting, 33
    viewing connected, 15
Interface, 109
Interface error codes, 150

K

Keep On Top, 12

L

License, 12
License activation, 11
Limits Editor, 33
Live waveforms, 29
LockSession(), 121
Log View, 48

M

Measurement parameters, 37
Menus
    Options, 16
    Preferences, 59
Message History, 49
Mode
    selecting, 33
My TekExpress folder, 5

O

Options Menu, 12, 16

P

Panels
    Setup, 21
Parameters, 37, 40
Population Count, 26
Preferences Menu, 59
Probe parameters, 26, 37
Probes, 8
Probes supported, 8
Program example, 113
Programmatic Interface, 109

Q

QueryStatus(), 137

R

RecallSession(), 146
Record Length, 26, 37
Ref Levels, 37
Related documentation, 3
Remote Proxy Object, 111
Report Options, 61
Report sections, 61
Reports
    appending, 61
    naming, 61
    viewing, 61
Results Panel
    using, 59
Run(), 134

S

Sample Rate, 26, 37
SaveSession(), 146
SaveSessionAs(), 146
Saving Log files, 49
Saving reports, 61
Scope settings, 37
SelectSuite(), 132
SelectTest(), 130
SendResponse(), 138
Server, 110
Session files, 5
Session folders, 5
SetDutId(), 123
SetGeneralParameter(), 125
SetTimeOut(), 135
Setup Panel, 21
SetVerboseMode(), 122
Signal Path Compensation (SPC), 46
Status Panel
    using, 48
Stop(), 134
System requirements, 8

T

TekExpress Client, 109
TekExpress Client requirements, 110
TekExpress Server, 109
Test fixture supported, 8

Test groups, 24
Test limits
    changing, 33
Test Reports
    selecting options, 61
    viewing, 61
Test results
    viewing, 59
Test Results Summary, 59
Test setup files, 5
Test setups
    creating based on existing, 67
    deleting, 67
    recalling, 66
    saving, 66
Test Status, 48
Tests
    configuring, 33
    selecting, 24
TF-GBE-BTP test fixture, 8
TransferImages(), 143
TransferReport(), 142
TransferWaveforms(), 143

U

UnlockSession(), 148
User Comments
    including in test reports, 61
    viewing in reports, 61
User Defined Mode, 33, 36

V

Version, 12
Vertical Scale, 26
Vertical Setup, 37

W

Waveforms
    acquiring live, 29