

VISIQ version 1.0.6

[illegible]

Copyright © 2024 Quantifi Photonics Ltd All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form, be it electronically, mechanically, or by any other means such as photocopying, recording or otherwise, without the prior written permission of Quantifi Photonics Ltd (Quantifi Photonics).

Information provided by Quantifi Photonics is believed to be accurate and reliable.

However, no responsibility is assumed by Quantifi Photonics for its use nor for any infringements of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent rights of Quantifi Photonics.

The information contained in this publication is subject to change without notice.

Trademarks

Quantifi Photonics' trademarks have been identified as such. However, the presence or absence of such identification does not affect the legal status of any trademark. All third party product and company names are trademarks™ or registered® trademarks of their respective holders. Neither Quantifi Photonics nor any software programs or other goods or services offered by Quantifi Photonics are affiliated with, endorsed by, or sponsored by the third parties listed below.

LabVIEW™ is a trademark of National Instruments. MATLAB® is a trademark of The MathWorks Inc. Python® is a trademark of the Python Software Foundation. Microsoft Windows® is a trademark of the Microsoft Corporation. Tektronix® is a trademark of Tektronix, Inc. Teledyne LeCroy® is a trademark of Teledyne Lecroy, Inc. Keysight is a trademark of Keysight Technologies, Inc.

Units of Measurement

Units of measurement in this publication conform to SI standards and practices.

User manual version: 1.06

Table of contents

1 Conventions	6
2 VISIQ overview	7
2.1 Programming interfaces	8
2.2 DSP pipeline and nodes	8
3 Setting up hardware	10
4 Installing the VISIQ software package	12
5 Working with VISIQ	13
5.1 Select the data source	14
5.2 Connect VISIQ with the coherent receiver	15
5.3 Set channels	16
5.4 Configure signal properties	17
5.5 Configure the DSP pipeline	20
5.5.1 Pipeline configuration	20
5.5.2 Node: CoRx	21
5.5.3 Node: CoRx auto calibrate	22
5.5.4 Node: Filter	24
5.5.5 Node: Retimer	25
5.5.6 Node: PolDeMux	26
5.5.7 Node: Equalizer	27
5.5.8 Node: Phase estimator	28
5.5.9 Node: MATLAB	29
5.5.10 Node: Nonlinear correction	30
5.5.11 Node: Find Baud rate	31
5.5.12 Node: Chromatic dispersion	32
5.5.13 Node: IQ offset removal	32
5.5.14 Node: Carrier offset removal	32

5.6 Run the analysis	33
5.7 Run tests	34
5.8 Visualize data	36
5.8.1 Plot details	37
5.9 Select measurements	41
5.9.1 Measurement details	42
5.10 Load and save data	47
5.11 System menu functions	49
5.11.1 Load and save data	49
5.11.2 Message log	50
5.11.3 VISIQ information in About window	51
6 Working with SCPI commands	52
6.1 SCPI overview	52
6.1.1 Programming conventions	52
6.1.2 Status and event registers	53
6.2 SCPI commands summary	56
6.2.1 Common commands	56
6.2.2 System menu functions	57
6.2.3 Select the data source	58
6.2.4 Connect VISIQ with the coherent receiver	60
6.2.5 Set channels	62
6.2.6 Configure signal properties	63
6.2.7 Configure the DSP pipeline	65
6.2.8 Run the analysis	87
6.2.9 Run tests	88
6.2.10 Select measurements	90
6.2.11 Load and save data	96

6.3 SCPI commands descriptions	97
6.3.1 Common commands	97
6.3.2 System menu functions	99
6.3.3 Select the data source	100
6.3.4 Connect VISIQ with the coherent receiver	104
6.3.5 Set channels	108
6.3.6 Configure signal properties	110
6.3.7 Configure the DSP pipeline	117
6.3.8 Run the analysis	150
6.3.9 Run tests	151
6.3.10 Select measurements	157
6.3.11 Load and save data	164
6.4 SCPI programming examples	166
6.4.1 Example A	166
6.5 Programming applications	169
6.5.1 Setting up NI-MAX application	170
6.5.2 Setting up NI-VISA application	171
6.5.3 Python® 2.7 code example	172
6.5.4 MATLAB® code example	173
7 System requirements	174
8 Technical Support	175
8.1 Contacting the Technical Support Group	175

1 Conventions

Please make yourself familiar with these conventions; we use them throughout this user manual:

WARNING

Indicates a potentially hazardous situation which, if not avoided, could result in **death or serious injury**.

Do not proceed unless the required conditions are met and understood.

CAUTION

Indicates a potentially hazardous situation which, if not avoided, may result in **minor or moderate injury** or **component damage**.

Do not proceed unless the required conditions are met and understood.

NOTE

Indicates relevant information that requires your attention.

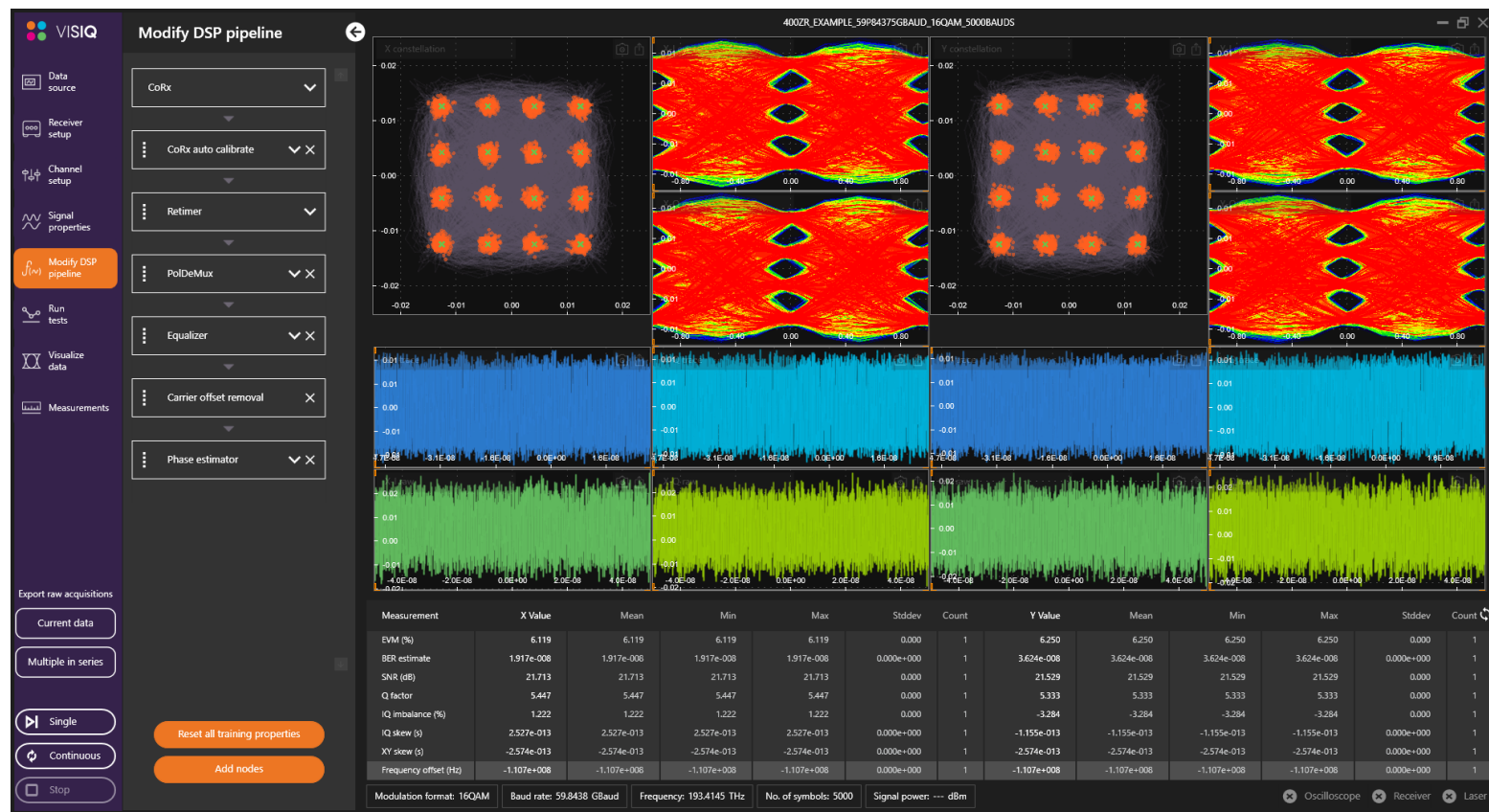
2 VISIQ overview

Quantifi Photonics' coherent signal analysis software VISIQ is designed to make the complex process of coherent signal measurement and DSP optimization as simple as possible. The software performs three main functions: Control of the OMA hardware, digital signal processing and the presentation of the signal properties.

VISIQ works with all popular models of high-performance real-time oscilloscopes and allows an easy configuration of hardware calibration parameters to ensure your hardware impairments are compensated for.

The intuitive and easy-to-use DSP pipeline uses reconfigurable functional nodes that you can drag-and-drop in the graphical sequencer. This innovative approach enables you to try different DSP configurations with immediate feedback, so you can quickly learn how to optimize your coherent DSP. Coherent receiver auto calibration, auto Baud rate detection, adaptive equalizer and nonlinear correction algorithms are available in VISIQ to significantly reduce the set-up time.

For repetitive testing, test automation is available using SCPI and gRPC.



2.1 Programming interfaces

Through its programming interfaces you can take advantage of both gRPC and the SCPI-compliant command language, and choose from programming tools such as LabView, C++, Python, or other programming languages used to control automatic test equipment (ATE).

2.2 DSP pipeline and nodes

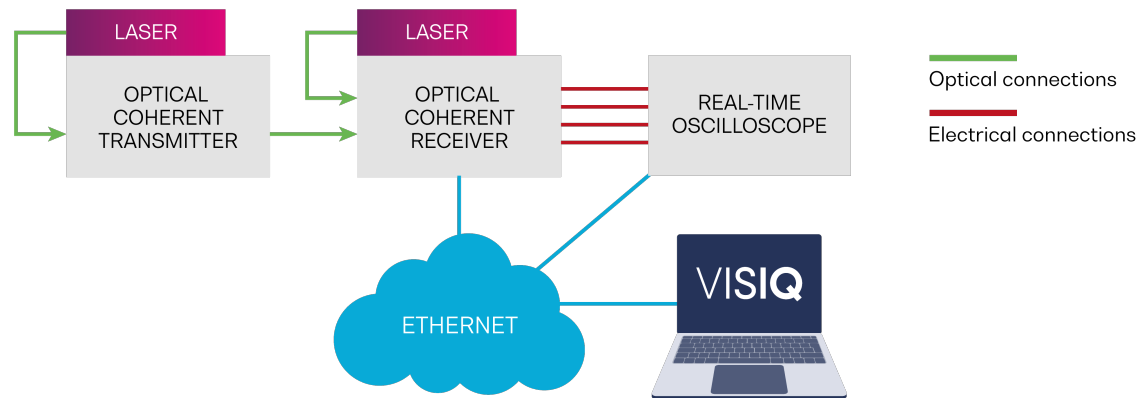
When configuring your pipeline, you can select from the following nodes:

Node	Restrictions	Description
CoRx	☒ always the first node in the pipeline ☒ cannot be removed	This node compensates coherent receiver imperfections such as skew, frequency response, and IQ phase error. Relevant receiver settings are stored in a settings file (.crs format). Example settings files are included in the VISIQ software package, locate them in ...\\QuantifiPhotonics\\VisIQ\\Examples\\Configurations.
Carrier offset removal		This node estimates and compensates the residual frequency offset between the carrier and the local oscillator that may be present in your signal. The residual carrier must be $< \pm 10\%$ of the Baud rate to work effectively.
Chromatic dispersion		This node compensates for chromatic dispersion due to fiber propagation.
CoRx auto calibrate		This node manages automatic calibration of the coherent receiver. It calculates and compensates for the DC offsets and the gain for the XI/XQ pair and the YI/YQ pair. It also calculates residual skew between the XI/XQ pair and the YI/YQ pair in the system.
Equalizer		This node calculates the complex (real and imaginary) Finite Impulse Response (FIR) taps. It attempts to correct the frequency response of the signal by training over a defined number over symbols. The training goal is to match the ideal reference symbols.
Filter		This node works as a digital filter that modifies the frequency components present in the signal.
Find Baud rate	☒ only 1 per pipeline ☒ insert before Retimer node (if used)	This node detects the Baud rate of the signal and replaces the Baud rate setting in the Stream properties with the discovered Baud rate.

Node	Restrictions	Description
IQ offset removal		This node corrects IQ offsets that may be present on the signal.
MATLAB		This node enables you to add a MATLAB script to the pipeline. The node is only enabled if you have a MATLAB license and an installed MATLAB version 2014a or newer. In the MATLAB node, define the location of the MATLAB file. Example scripts are included in the VISIQ software package, locate them in ...\\Quantifi Photonics\\VisIQ\\Examples\\Scripts.
Nonlinear correction		This node can correct gain imbalances, IQ offsets as well as non-linear distortion that may present on the signal.
Phase estimator		This node calculates and removes the phase errors over time that may be present in the signal. The phase error occurs as a result of the phase noise difference between the carrier and the local oscillator laser.
PolDeMux		This node demultiplexes the two polarizations of the carrier.
Retimer	☒ only 1 per pipeline ☒ insert after Find Baud rate node (if used) ☒ cannot be removed	This node retimes the input to a particular sample per Baud. Interpolation is used to create traces of samples per Baud higher than the input. The algorithm also calculates the skew between x pol and y pol that may be present.

3 Setting up hardware

For details on hardware requirements, please refer to [System requirements](#).



► Set up hardware:

1. **Connect the coherent optical receiver to the oscilloscope.**
Use a set of four phase-matched, high-bandwidth RF cables.
Phase matching of $<1\text{ps}$ is strongly recommended.
2. **Connect the oscilloscope to the computer running VISIQ.**
Use an Ethernet connection. If the Ethernet environment does not support DHCP automatic IP assignment, set up static IP addresses on the oscilloscope and the computer to ensure that they can ping each other.
3. **Coherent optical receiver: If using Quantifi Photonics' IQRX, connect it to the computer running VISIQ.**
Use either an Ethernet or USB connection.
If using USB, install Cohesion Operator on the computer first, as it will install the USB driver for IQRX. For details, refer to the IQRX user manual.
4. **Connect the optical signal from the coherent optical transmitter to the coherent optical receiver signal input.**

WARNING

Practice safe optical handling to protect yourself from harm caused by optical radiation:

- Turn the laser power OFF when making optical connections.
- Clean the optical end-face each time before making connections.
- Once connected, turn the TX laser/signal back on.

► Configure the hardware setup:

1. **Oscilloscope:** Perform channel deskew. Oscilloscopes typically offer a routine for self-deskew.
2. **IQ transmitter hardware: Set it up to generate a reference signal.**
A 32GBaud 16QAM signal is recommended as a starting point.
An optical signal power of -3dBm to 0 dBm is recommended.
3. **Local Oscillator (receiver laser): Set the frequency/wavelength to match that of the signal transmitter and turn it on.**
It is recommended that the LO power be set to the maximum value, not exceeding the input power rating of the coherent receiver.
4. **Oscilloscope:** Check and enable four input channels on the oscilloscope and run oscilloscope acquisitions to check that an appropriate level of signal amplitude is seen. If using Quantifi Photonics' IQRX, a signal amplitude level of 100mVpp or higher is normal and recommended.



WARNING

It is important that you do NOT exceed the maximum voltage rating of the oscilloscope input. The maximum input voltage is related to the vertical scale setting. Ensure that the oscilloscope is set to a large vertical scale setting such as 100mV/Div, before connecting the coherent receiver or turning on the optical signal input to the receiver. Exceeding the full scale voltage can damage the oscilloscope.

4 Installing the VISIQ software package

Your Quantifi Photonics sales manager will provide a link and password to access the downloadable software package.

For software updates, renewal of the license and transfer of license to a different computer, please contact your Quantifi Photonics sales manager or email sales@quantifiphotonics.com.

Install and activate the software package on the computer that will run VISIQ as follows.

▶ To install VISIQ:

1. Download the VISIQ software package and save it on the computer that will run VISIQ.
2. Run **VISIQ_<version>.exe** and follow the on-screen installation prompts.

The installation wizard will install required drivers, applications, and desktop icons on the computer.

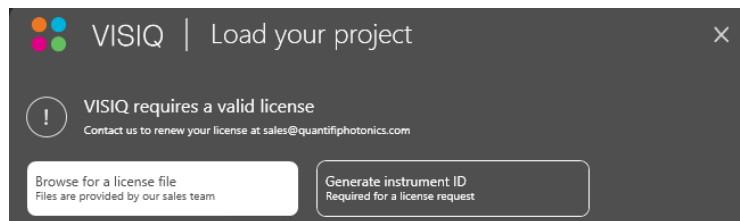
▶ To activate VISIQ:

1. Open the VISIQ application. In the startup screen, VISIQ will ask for a valid license.
2. To request your license, click **Generate instrument ID**.
3. VISIQ will generate a **hardwareid.hid** file; please email the file to your Quantifi Photonics sales manager.

Note: The license is not transferable. It is only valid for the computer that you have used to generate the instrument ID and cannot be used on (or transferred to) another computer.

4. We will email your license to you; please save it anywhere on the computer.
5. In the startup screen, browse for the license and select it.

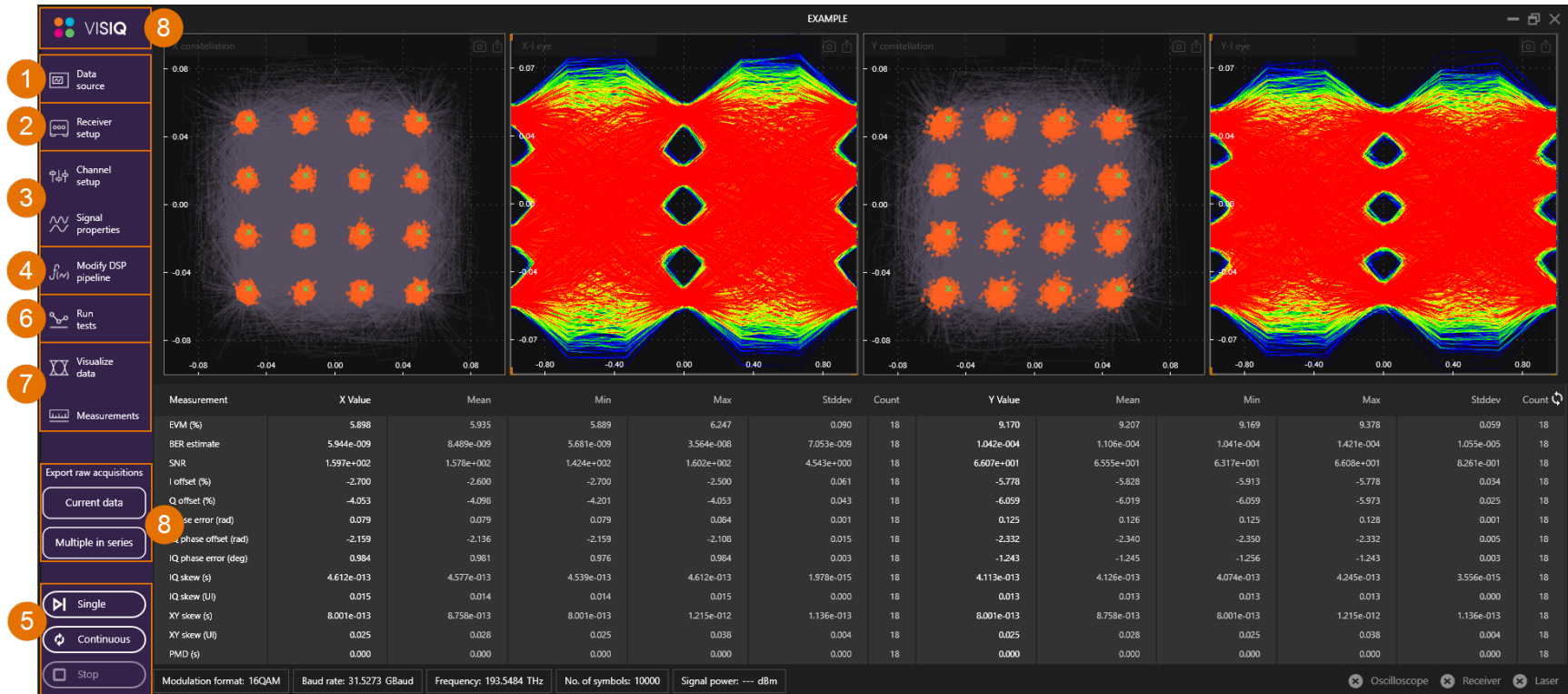
Once the license is verified, VISIQ will be ready for use. Should you encounter any difficulties or have any questions about the license activation process, please contact your Quantifi Photonics sales manager.



5 Working with VISIQ

VISIQ enables you to analyze and visualize your signal:

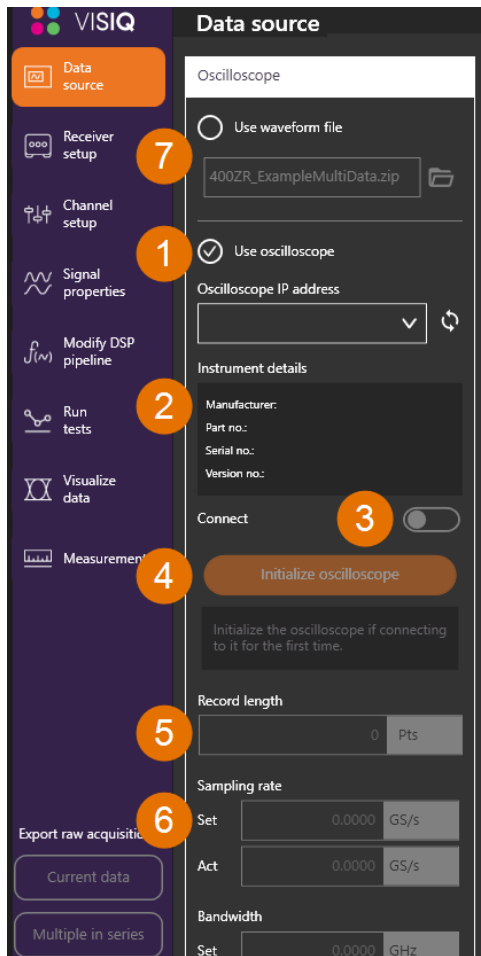
1. Select the **data source**: Connect VISIQ with an oscilloscope of your test setup, or select a previously acquired waveform file.
2. Set up the **coherent receiver**.
3. Set **channels** and configure **signal properties**.
4. Configure the **DSP pipeline**.
5. **Run the analysis**.
6. Run error detection tests
7. Select plots and measurements to display.
8. Load/Save data for further analysis at a later stage, and access system menu functions.



5.1 Select the data source

Connect VISIQ with an oscilloscope of your test setup to analyze live data, or load a previously acquired waveform file.

Parameter	Description
Record length	The number of data points to be captured by the oscilloscope for a single acquisition.
Sampling rate	The sampling rate of the oscilloscope. If the Set value is not accepted by the oscilloscope, it will snap to an acceptable value and be displayed in Act (actual value).
Bandwidth	Sets the bandwidth of all of the oscilloscope channels in use. If the Set value is not accepted by the oscilloscope, it will snap to an acceptable value and be displayed in Act (actual value).



► To connect with an oscilloscope:

1. In **Data source > Oscilloscope**, select **Use oscilloscope**.
Select the **Oscilloscope IP address** from the dropdown, or enter the IP address. Refresh the list of available devices if required. If you are using another OMA software product with the oscilloscope, please close it down.
2. VISIQ will display details of the oscilloscope.
3. **Connect** the oscilloscope.
4. If connecting with the oscilloscope for the first time or after a reboot of the oscilloscope, click **Initialize oscilloscope** to configure it to a state compatible with VISIQ.
5. VISIQ will display **Record length**.
6. Enter oscilloscope settings: **Sampling rate** and **Bandwidth**.
Repeat these steps to connect with each oscilloscope of your test setup as required.

► To load a waveform file:

7. In **Data source > Oscilloscope**: Select **Use waveform file**.

Browse and select the file.

Example files are included in the VISIQ software package in folder:
...\Quantifi Photonics\VisIQ\Examples\Projects.

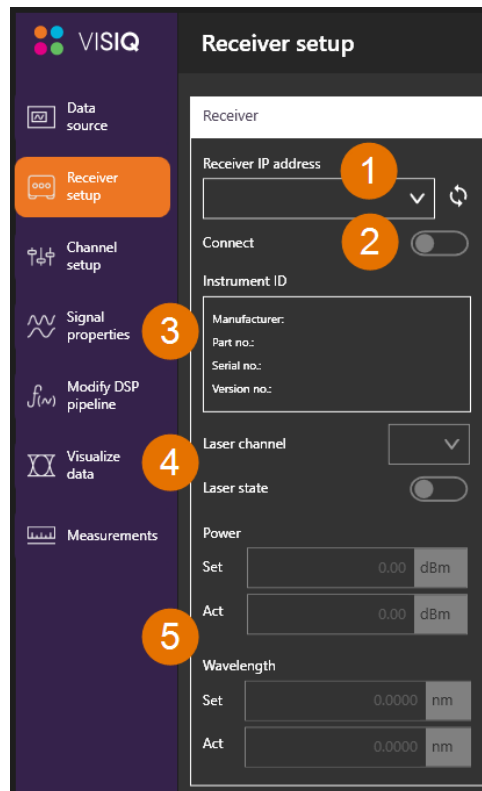
5.2 Connect VISIQ with the coherent receiver

This is required if you have selected an oscilloscope as your data source to acquire live data.

This section describes how to connect a Quantifi Photonics IQRX receiver.

If you're using a different, non-Quantifi Photonics coherent receiver, connect the receiver in the CoRx node of the DSP pipeline instead. For details, refer to **Configure the DSP pipeline > Node: CoRx**.

Parameter	Description
Power	Output power of the selected laser.
Wavelength	Output wavelength of the selected laser.



► To connect with a **Quantifi Photonics IQRX receiver**:

1. In **Receiver setup > Receiver**, select the **Receiver IP address** from the dropdown, or enter the IP address. Refresh the list of available devices if required.
2. **Connect** the receiver.
3. VISIQ will display identification details of the receiver.

► To control an IQRX laser:

4. Select a **Laser channel**.
Set the **Laser state** to **ON** or **OFF**.
5. Enter the laser output **Power** and **Wavelength**.
Set: Enter a value.
Act: Displays the actual value.

► To connect with a **non-Quantifi Photonics receiver**:

- Select the receiver type and settings file in **Modify DSP pipeline > CoRx** instead. For details, refer to [Node: CoRx](#).

5.3 Set channels

Set channel mapping to link the coherent receiver tributaries (XI, XQ, YI, YQ) to the oscilloscope channels.



► To set channel mapping:

1. In **Channel setup > Channel mapping**: Map the oscilloscope channels with the coherent receiver.

► To set the oscilloscope vertical scale:

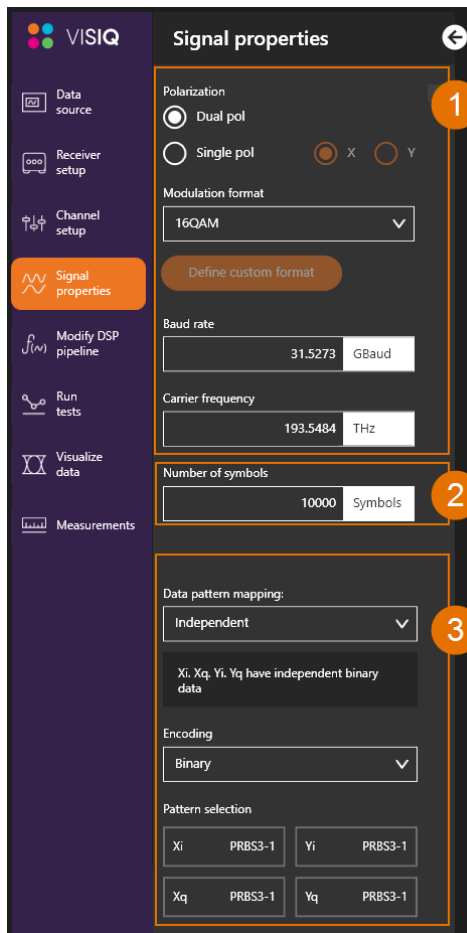
2. In **Channel setup > Oscilloscope vertical scale**: Enter the vertical scale by channel, or click **Autoscale**.

5.4 Configure signal properties

VISIQ can process single and dual polarization signals.

The pattern and encoding you set up in **Signal properties** will be copied over to **Run tests** and vice versa.

Parameter	Description
Modulation format	Modulation format of the received signal.
Baud rate	Baud rate of the received signal.
Carrier frequency	Frequency of the optical carrier
Number of symbols	The total number of symbols to capture during data acquisition. The default value of 5,000 is a good starting point, adjust it as required. A greater number results in slower processing.
Data pattern mapping	Select data pattern mapping option: Independent: The four tributaries are individually modulated with PRBS or data patterns. Multiplexed IQ: The I and Q tributaries of the same polarization are multiplexed and modulated with a single PRBS or data pattern per polarization. Multiplexed XY: All four tributaries are multiplexed and modulated with a single PRBS or data pattern.
Encoding	Select decoder settings. Binary: The symbol's I and Q bits are encoded in an incremental order on each axes. Gray: The binary values are arranged in such a way that only one bit changes between adjacent symbols.
Pattern selection	Select the test pattern.



► To set signal properties:

1. Define the properties of the received signal:

- Select **Polarization**: **Dual pol**, or **Single pol** and **X** or **Y** axis.
- Select **Modulation format**.
To define a custom format, select **custom modulation format**. For details, refer to [Custom modulation formats](#).
- Enter **Baud rate**.
- Enter **Frequency**.

2. Set the total number of symbols to capture during data acquisition:

- Enter **No. of symbols**.

3. When running error detection testing, select test parameters (for details, refer to [Run tests](#)):

- Select **Data pattern mapping**.
- Select **Encoding**.
- In **Pattern selection**: Select the test pattern by channel.

Custom modulation formats

If required, you can define a custom modulation format:

1. In **Modulation format**, select **custom modulation format**.
2. Click **Define custom format**.
3. Add or edit symbols by specifying I and Q coordinates in the first quadrant (positive values between 0 and 1).

For symmetric modulation formats, VISIQ automatically generates symbols across the remaining quadrants based on the first quadrant.

The image shows the VISIQ interface. On the left is the 'Signal properties' panel with a sidebar containing 'Data source', 'Receiver setup', 'Channel setup', 'Signal properties' (highlighted), 'Modify DSP pipeline', 'Run tests', 'Visualize data', and 'Measurements'. The 'Signal properties' panel has fields for 'Polarization' (Dual pol, Single pol), 'Modulation format' (set to 'Custom modulation format'), 'Baud rate' (10.0000 GBaud), 'Carrier frequency' (193.4145 THz), and 'Number of symbols' (5000 Symbols). An orange arrow points from the 'Define custom format' button in the 'Signal properties' panel to the 'Define custom modulation format' dialog box on the right.

The 'Define custom modulation format' dialog box has a title bar with a close button. It contains the instruction: 'Define positive I and Q coordinate (between 0 and 1) and VISIQ will generate corresponding symbols'. Below this is an 'Add symbol' button and two input fields for 'I coordinate' and 'Q coordinate', both set to '1.000'. A circled '3' is next to the 'Q coordinate' field. Below these is a table for 'Rotated symbols'.

	I coordinate	Q coordinate
Positive quadrant symbols		
Rotated symbols		
1	1.000	1.000
2	1.000	-1.000
3	-1.000	-1.000
4	-1.000	1.000

To the right of the table is a constellation diagram showing a grid with axes labeled 'I' and 'Q'. Four green '+' symbols are plotted at the corners of the unit square: (1, 1), (1, -1), (-1, -1), and (-1, 1). At the bottom of the dialog are three buttons: 'Apply', 'Reset to default', and 'Close'.

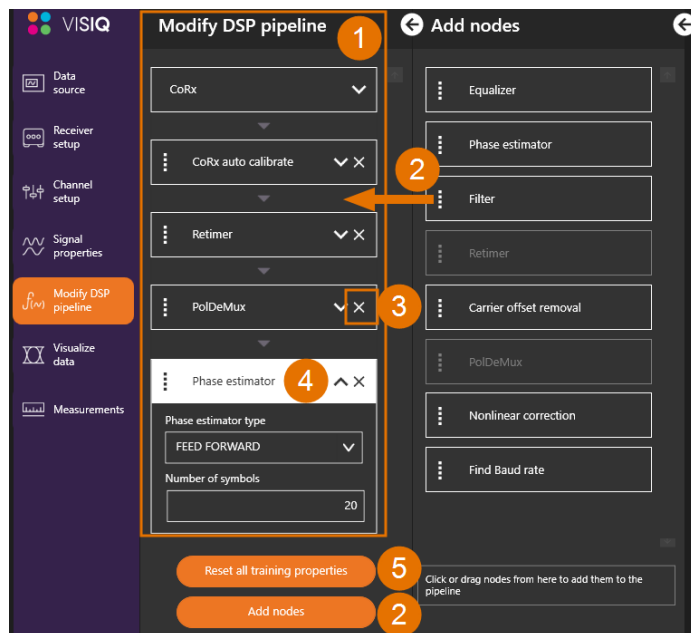
5.5 Configure the DSP pipeline

5.5.1 Pipeline configuration

You configure your DSP pipeline in VISIQ by adding, moving and/or removing configurable nodes into position as required.

- ▶ To manage your pipeline:
 1. This is your pipeline.
 2. Click **Add nodes** and drag and drop nodes into the pipeline.
 3. Click **x** to remove nodes from the pipeline.
 4. Expand a node to configure it.
 5. Click **Reset all training properties** to return training properties of all nodes that support training to default values. This applies to the nodes: **CoRx auto calibrate**, **Equalizer** and **Nonlinear correction**.

- The minimum configuration of the pipeline includes: **CoRx**, **CoRx auto calibrate**, **Retimer**, **PolDeMux**, **Carrier Offset**, **Phase estimator**.
- Some nodes can appear in the pipeline only once; they will then appear grayed out in the **Add nodes** column in the UI.

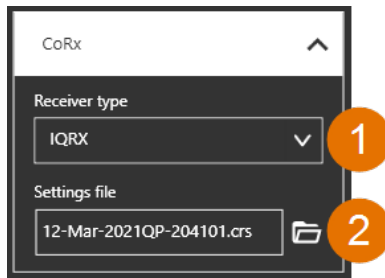


5.5.2 Node: CoRx

- ☑ always the first node in the pipeline
- ☒ cannot be removed

This node compensates coherent receiver imperfections such as skew, frequency response, and IQ phase error.

Relevant receiver settings are stored in a settings file (.crs format). Example settings files are included in the VISIQ software package, locate them in ...\\QuantifiPhotonics\\VisIQ\\Examples\\Configurations.



► To define the **CoRx** node settings:

1. In **CoRx**, select the receiver type:
 - **IQRX**: Quantifi Photonics' Coherent Optical Receiver IQRX
 - **Other**: A non-Quantifi Photonics coherent receiver. You will need to manually configure a .crs file for your receiver.
2. Select a settings file.

When creating a settings file:

- Limit the size of the arrays to 1,000.
- Limit the number of decimal places so that the maximum line length is less than 10,000 characters.
- When using an example settings file that has been supplied with the VISIQ software package: Edit a copy of the file, leave the original file unchanged.

5.5.3 Node: CoRx auto calibrate

This node manages automatic calibration of the coherent receiver. It calculates and compensates for the DC offsets and the gain for the XI/XQ pair and the YI/YQ pair. It also calculates residual skew between the XI/XQ pair and the YI/YQ pair in the system.

This node is ideal for fine-tuning coherent receiver calibration. If large adjustments are required, please correct these in the CoRx node.

Parameter	Description
Gradient	Defines the convergence rate of the adaptation. Setting values close to the minimum of 1e-6 will result in longer convergence time over the training symbols, but better results in presence of noise. Vice versa applies for values close to the maximum value of 1. If running a single iteration, set the value to 1.
Skew	An estimate of the temporal skew of the channels, measured in seconds. This is an array with a size of 4, where the index corresponds to the skew of XI/XQ/YI/YQ respectively. This only compensates for the skew between the XI/XQ pair and the YI/YQ pair.
Gain	An estimate of the gain of the channels. This is an array with a size of 4, where the index corresponds to the gain of the XI/XQ/YI/YQ pair respectively. This only compensates for the skew between the XI/XQ pair and the YI/YQ pair.
DC offset	An estimate of the DC offset of the channels. This is an array with a size of 4, where the index corresponds to the DC offset of XI/XQ/YI/YQ respectively.

The screenshot shows the 'CoRx auto calibrate' settings window. It contains several sections: 'Gradient' with a single input field; 'Skew' with four channels (Ch 1 to Ch 4) each having an input field and a unit selector; 'Gain' with four channels (Ch 1 to Ch 4) each having an input field and a unit selector; 'DC offset' with four channels (Ch 1 to Ch 4) each having an input field and a unit selector; 'Training acquisitions' with an input field; 'Training counter' showing a value of 14; and two radio buttons for 'Continuous training' and 'Hold', along with a 'Reset' button.

Numbered callouts point to the following elements:

- 1: Gradient input field (value: 0.300000)
- 2: Skew input fields (Ch 1: 0.0000E+000, Ch 2: -1.7797E-012, Ch 3: 0.0000E+000, Ch 4: 3.7518E-013)
- 3: Gain input fields (Ch 1: 1.0000E+000, Ch 2: 9.8129E-001, Ch 3: 9.1048E-001, Ch 4: 9.1167E-001)
- 4: DC offset input fields (Ch 1: 2.0876E-003, Ch 2: 5.4471E-004, Ch 3: -7.7146E-004, Ch 4: 1.7685E-003)
- 5: Training acquisitions input field (value: 20)

► To select the settings for the **CoRx auto calibrate** node:

1. Enter **Gradient**.
2. Adjust **Skew** for each channel as required.
3. Adjust **Gain** for each channel as required.
4. Adjust **DC offset** for each channel as required.
5. Enter **Training acquisitions**. The default value is a good starting point, adjust it as required.

Training optimizes the algorithm of the node by iteratively adjusting parameters. The **Training counter** counts down by one with each acquisition, from the set value to 0. When the counter reaches 0, training will hold, the node will use the latest values.

Tick **Continuous training** for training to continue indefinitely.

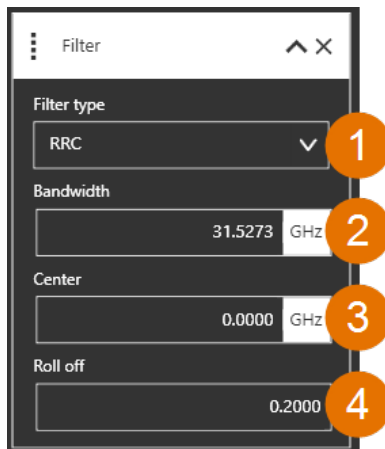
Click **Hold** to pause training.

Click **Reset** to reset the training counter, and reverse values as they were before capturing training acquisitions.

5.5.4 Node: Filter

This node works as a digital filter that modifies the frequency components present in the signal.

Parameter	Description
Filter type	The type of filter that will be applied. GAUSS : Gaussian filter RRC : Root raised cosine filter RCOSINE : Raised cosine filter BRICKWALL : Brickwall filter SINC : Sinc filter BESSEL : Bessel filter BUTTER : Butterworth filter
Bandwidth	Controls the filter's 3dB bandwidth.
Center	Sets the central frequency between the upper and the lower cutoffs.
Roll off	Controls the steepness of the frequency transfer function. Available for filter types RRC , RCOSINE , BRICKWALL , and SINC .
Order	The order of the filter. Available for filter types BESSEL and BUTTERWORTH .



► To set the filter properties:

1. Select **Filter type**.
2. Enter **Bandwidth**.
3. Enter **Center**.
4. Enter **Roll off** (available for filter types **RRC**, **RCOSINE**, **BRICKWALL**, and **SINC**).
5. Enter **Order** (available for Filter type **BESSEL** and **BUTTER**).

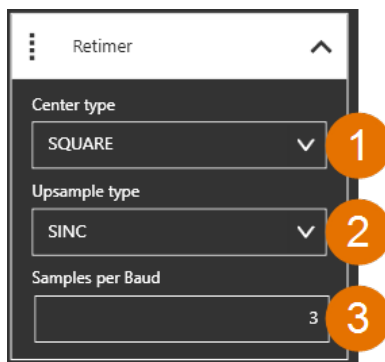
5.5.5 Node: Retimer

- ☑ only 1 per pipeline
- ☑ insert after **Find Baud rate** node (if used)
- ☒ cannot be removed

This node retimes the input to a particular sample per Baud.

Interpolation is used to create traces of samples per Baud higher than the input. The algorithm also calculates the skew between x pol and y pol that may be present.

Parameter	Description
Center type	The type of algorithm that will be used to find the symbol centers. SQUARE : Square timing recovery STDDEV : Queries the time location of the eye diagram with the largest standard deviation. CROSSPOINT : Queries the crossing points of the eye diagram and calculates the center as the distance between these crossing points. EVM : Error Vector Magnitude. The center location is the minimum EVM across the eye diagram.
Upsample type	The up-sampling technique that will be used to interpolate the inputs. LINEAR : Linear interpolation SINC : A sinc function interpolation
Samples per Baud	The number of samples per Baud to retime your signal to.



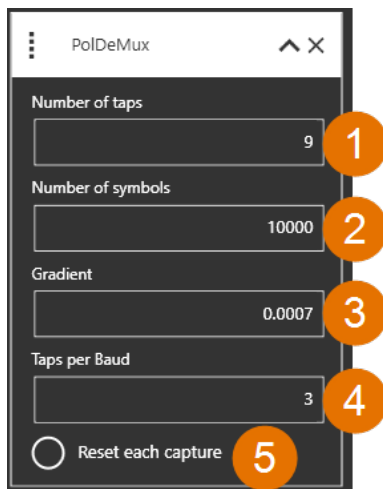
► To set the retimer properties:

1. Select **Center type**.
2. Select **Upsample type**.
3. Enter **Samples per Baud**.

5.5.6 Node: PolDeMux

This node demultiplexes the two polarizations of the carrier.

Parameter	Description
Number of taps	The number of taps. This value must be an odd value.
Number of symbols	Defines the length of the training sequence by the number of symbols. Make sure that the value is not higher than the Number of symbols in Signal properties .
Gradient	Defines the convergence rate of the adaptation. Setting values close to the minimum of 1.0e-6 will result in longer convergence time over the training symbols, but better results in presence of noise, and vice versa for values close to the maximum value of 1.0.
Taps per Baud	The number of taps per Baud.



► To set the PolDeMux properties:

1. Enter **Number of taps**.
2. Enter **Number of symbols**.
3. Enter **Gradient**.
4. Enter **Taps per Baud**.
5. Untick **Reset each capture** to use results from previous acquisitions.

5.5.7 Node: Equalizer

This node calculates the complex (real and imaginary) Finite Impulse Response (FIR) taps. It attempts to correct the frequency response of the signal by training over a defined number over symbols. The training goal is to match the ideal reference symbols.

Parameter	Description
Number of taps	The number of taps in the equalizer kernel. This value must be an odd value.
Number of symbols	Defines the length of the training sequence by the number of symbols.
Gradient	Defines the convergence rate of the adaptation. Setting values close to the minimum of $1.0e-6$ will result in longer convergence time over the training symbols, but better results in presence of noise, and vice versa for values close to the maximum value of 1.0.
Taps per Baud	The number of taps per Baud. Set this value to the number of samples per Baud of your streams at the input into the node.

The screenshot shows the 'Equalizer' node configuration window. It contains five input fields, each with a numbered orange circle next to it: 1. 'Number of taps' with a value of 99. 2. 'Number of symbols' with a value of 10000. 3. 'Gradient' with a value of 0.000100. 4. 'Taps per Baud' with a value of 3. 5. 'Training acquisitions' with a value of 18. Below these fields, there is a 'Training counter: 0' label, a radio button for 'Continuous training' (which is selected), a 'Hold' button with a checkmark, and a 'Reset' button.

► To set the equalizer properties:

1. Enter **Number of taps**.
2. Enter **Number of symbols**.
3. Enter **Gradient**.
4. Enter **Taps per Baud**.
5. Enter **Training acquisitions**. The default value is a good starting point, adjust it as required.

Training optimizes the algorithm of the node by iteratively adjusting parameters. The **Training counter** counts down by one with each acquisition, from the set value to 0. When the counter reaches 0, training will hold, the node will use the latest values.

Tick **Continuous training** for training to continue indefinitely.

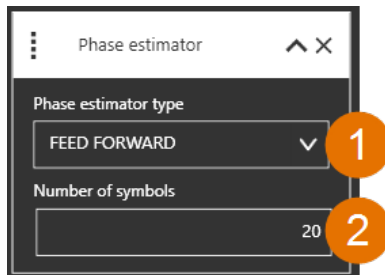
Click **Hold** to pause training.

Click **Reset** to reset the training counter, and reverse values as they were before capturing training acquisitions.

5.5.8 Node: Phase estimator

This node calculates and removes the phase errors over time that may be present in the signal. The phase error occurs as a result of the phase noise difference between the carrier and the local oscillator laser.

Parameter	Description
Phase estimator type	The type of the algorithm used. FEED FORWARD : Square timing recovery VITERBI-VITERBI : Queries the time location of the eye diagram with the largest standard deviation. IMPROVED VITERBI-VITERBI : Queries the crossing points of the eye diagram and calculates the center as the distance between these crossing points.
Number of symbols	Defines the length of the training sequence by the number of symbols.



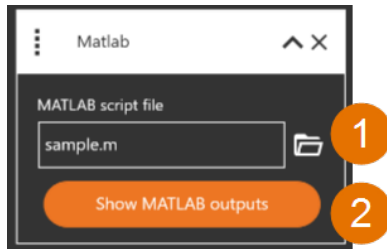
► To set the phase estimator properties:

1. Select **Phase estimator type**.
2. Enter **Number of symbols**.

5.5.9 Node: MATLAB

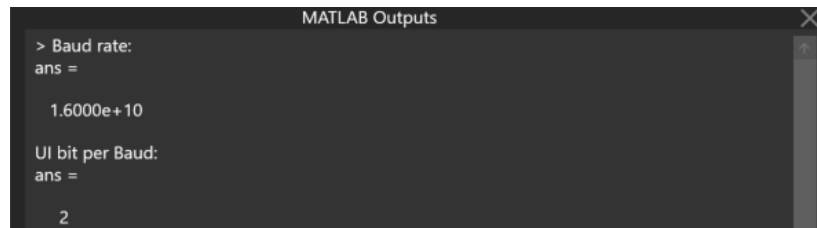
This node enables you to add a MATLAB script to the pipeline.

The node is **only** enabled if you have a MATLAB license and an installed MATLAB version 2014a or newer. In the MATLAB node, define the location of the MATLAB file. Example scripts are included in the VISIQ software package, locate them in ...\\Quantifi Photonics\\VisIQ\\Examples\\Scripts.



► To select a MATLAB script:

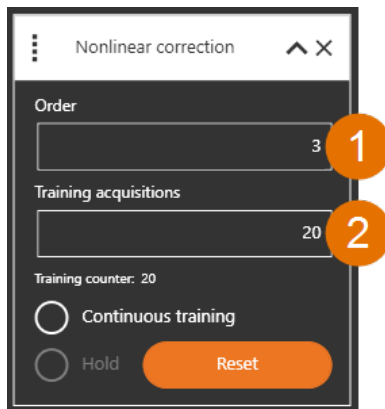
1. Browse and select the MATLAB script file.
2. Click **Show MATLAB outputs** to display command output.



5.5.10 Node: Nonlinear correction

This node can correct gain imbalances, IQ offsets as well as non-linear distortion that may present on the signal.

Parameter	Description
Order	The order of the nonlinear polynomial. Higher-order polynomials can exhibit more complex behaviours or signal correction.



► To set the nonlinear correction properties:

1. Enter **Order**.
2. Enter **Training acquisitions**. The default value is a good starting point, adjust it as required.

Training optimizes the algorithm of the node by iteratively adjusting parameters. The **Training counter** counts down by one with each acquisition, from the set value to 0. When the counter reaches 0, training will hold, the node will use the latest values.

Tick **Continuous training** for training to continue indefinitely.

Click **Hold** to pause training.

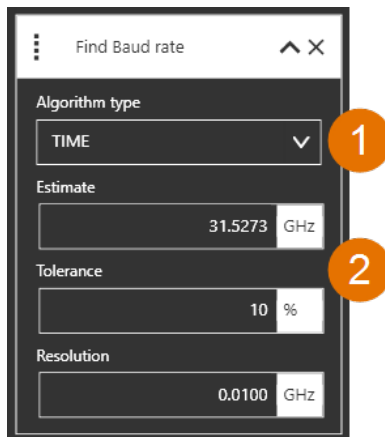
Click **Reset** to reset the training counter, and reverse values as they were before capturing training acquisitions.

5.5.11 Node: Find Baud rate

- ☒ only 1 per pipeline
- ☒ insert before **Retimer** node (if used)

This node detects the Baud rate of the signal and replaces the Baud rate setting in the Stream properties with the discovered Baud rate.

Parameter	Description
Algorithm type	The type of algorithm used. TIME: Finds the largest cross-correlation between the signal and a simulated clock in the time domain. This can be computationally expensive and slow. Use this algorithm if you have little to no signal energy at the symbol rate. FREQ: Finds the Baud rate clock component in the frequency domain and then fine tunes the clock in the time domain.
Estimate	Approximate Baud rate for the algorithm to use as a starting point.
Tolerance	The range of the Baud rate for the algorithm to consider, expressed as a percentage of the estimate Baud rate.
Resolution	The step-size or resolution of the Baud rate for the algorithm.

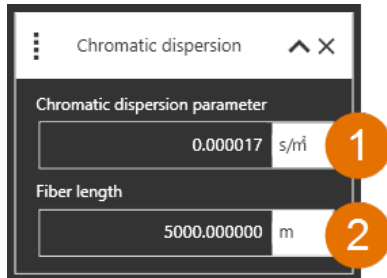


- ▶ To set the node properties:
1. Select **Algorithm type**.
 2. Enter **Estimate**, **Tolerance** and **Resolution**.

5.5.12 Node: Chromatic dispersion

This node compensates for chromatic dispersion due to fiber propagation.

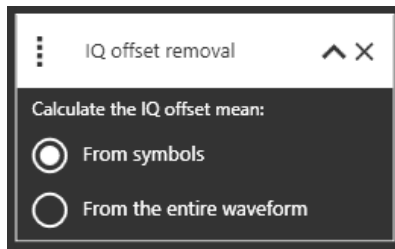
Parameter	Description
Chromatic dispersion parameter	The chromatic dispersion parameter value of the fiber in use, expressed in the unit of Seconds per meter squared.
Fiber length	Length of the dispersion inducing fiber in the link to compensate for.



- ▶ To set the node properties:
1. Enter **Chromatic dispersion parameter**.
 2. Enter **Fiber length**.

5.5.13 Node: IQ offset removal

This node corrects IQ offsets that may be present on the signal.



- ▶ To set the node properties:
1. Select how to calculate the IQ offset mean: **From symbols** or **From the entire waveform**.

5.5.14 Node: Carrier offset removal

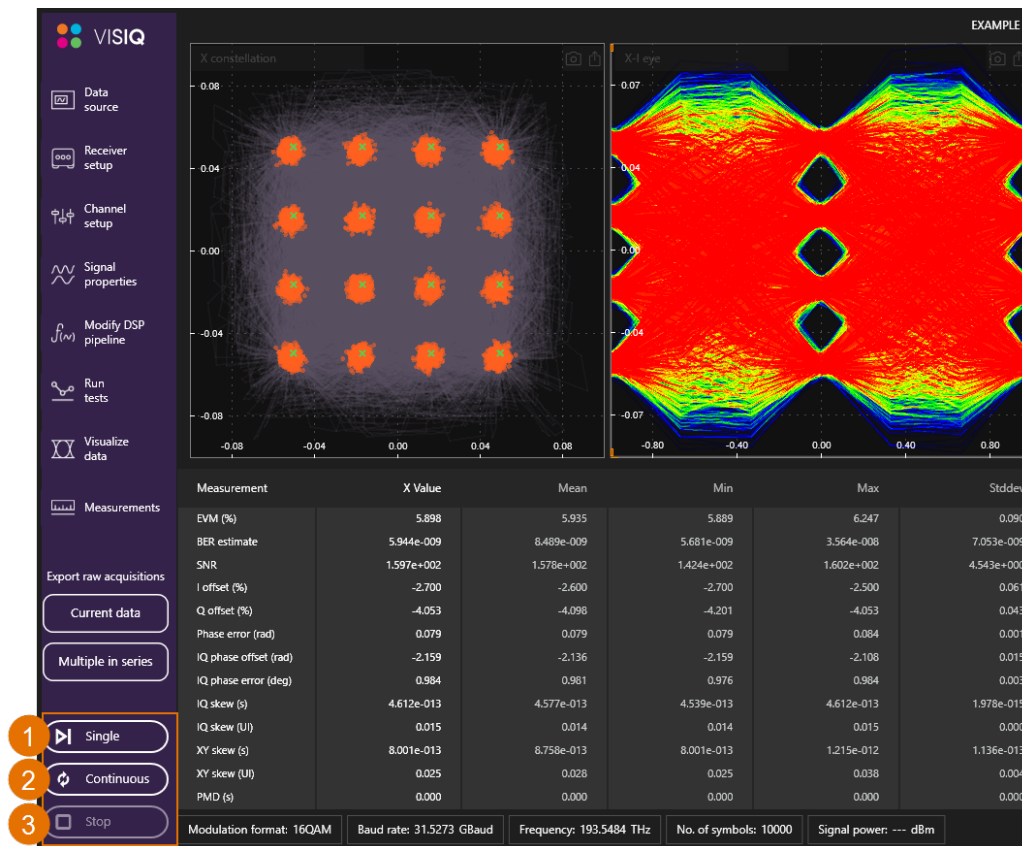
This node estimates and compensates the residual frequency offset between the carrier and the local oscillator that may be present in your signal. The residual carrier must be $\leq \pm 10\%$ of the Baud rate to work effectively.

5.6 Run the analysis

When running the analysis, VISIQ acquires data as per your selected settings and provides immediate visual feedback. You can customize the visualization area and measurement table by selecting plots and measurements to display.

► To run the analysis:

1. Click **Single** to capture a single acquisition and capture the number of symbols as selected. Plots and table values will reflect captured data; the counter in the measurement table will go up by one.
2. Click **Continuous** to repeatedly catch a single acquisition. The plots and table values will update as data is captured; the counter in the table will go up by one with each acquisition.
3. Click **Stop** to stop continuous data acquisition.

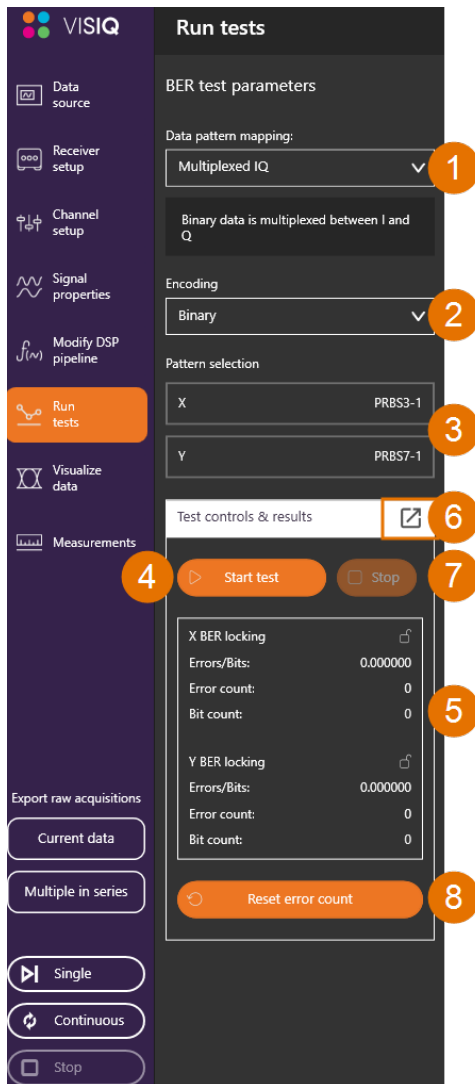


5.7 Run tests

The Bit Error Rate measurement in **Run tests** compares the received data pattern to the pre-defined reference pattern configured by the user to make an accumulative error count. VISIQ's BER test supports various multiplexing and encoding options and common PRBS patterns ranging from PRBS3 to PRBS31 are available from the selection menu.

The pattern and encoding you set up in **Run tests** will be copied over to **Signal properties** and vice versa.

Parameter	Description
Data pattern mapping	Select data pattern mapping option: Independent: The four tributaries are individually modulated with PRBS or data patterns. Multiplexed IQ: The I and Q tributaries of the same polarization are multiplexed and modulated with a single PRBS or data pattern per polarization. Multiplexed XY: All four tributaries are multiplexed and modulated with a single PRBS or data pattern.
Encoding	Select decoder settings. Binary: The symbol's I and Q bits are encoded in an incremental order on each axes. Gray: The binary values are arranged in such a way that only one bit changes between adjacent symbols.
Pattern selection	Select the test pattern.
Errors/Bits	Bit error rate: Total number of errors over total bits tested
Error count	Total number of errors detected
Bit count	Total number of bits tested

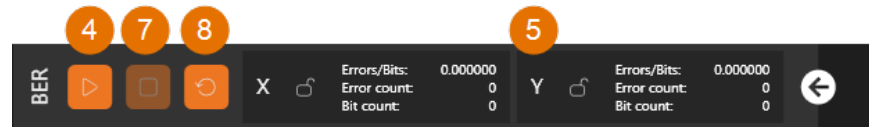


► To select test parameters:

1. In **Run tests**, select **Data pattern mapping**.
2. Select **Encoding**.
3. In **Pattern selection**: Select the test pattern by channel.

► To run the tests:

4. Click **Start test**. VISIQ will record and display errors with each acquisition.
5. Results will be displayed and refreshed with each acquisition. The **BER lock** indicator will turn true if the correlation between the received bit pattern and the reference known pattern exceeds a threshold level, indicating that there is a high confidence that the correct pattern is found and matched.
6. Additionally, you can display controls and results in the main visualization area, where they will remain visible when navigating away from **Run tests**.



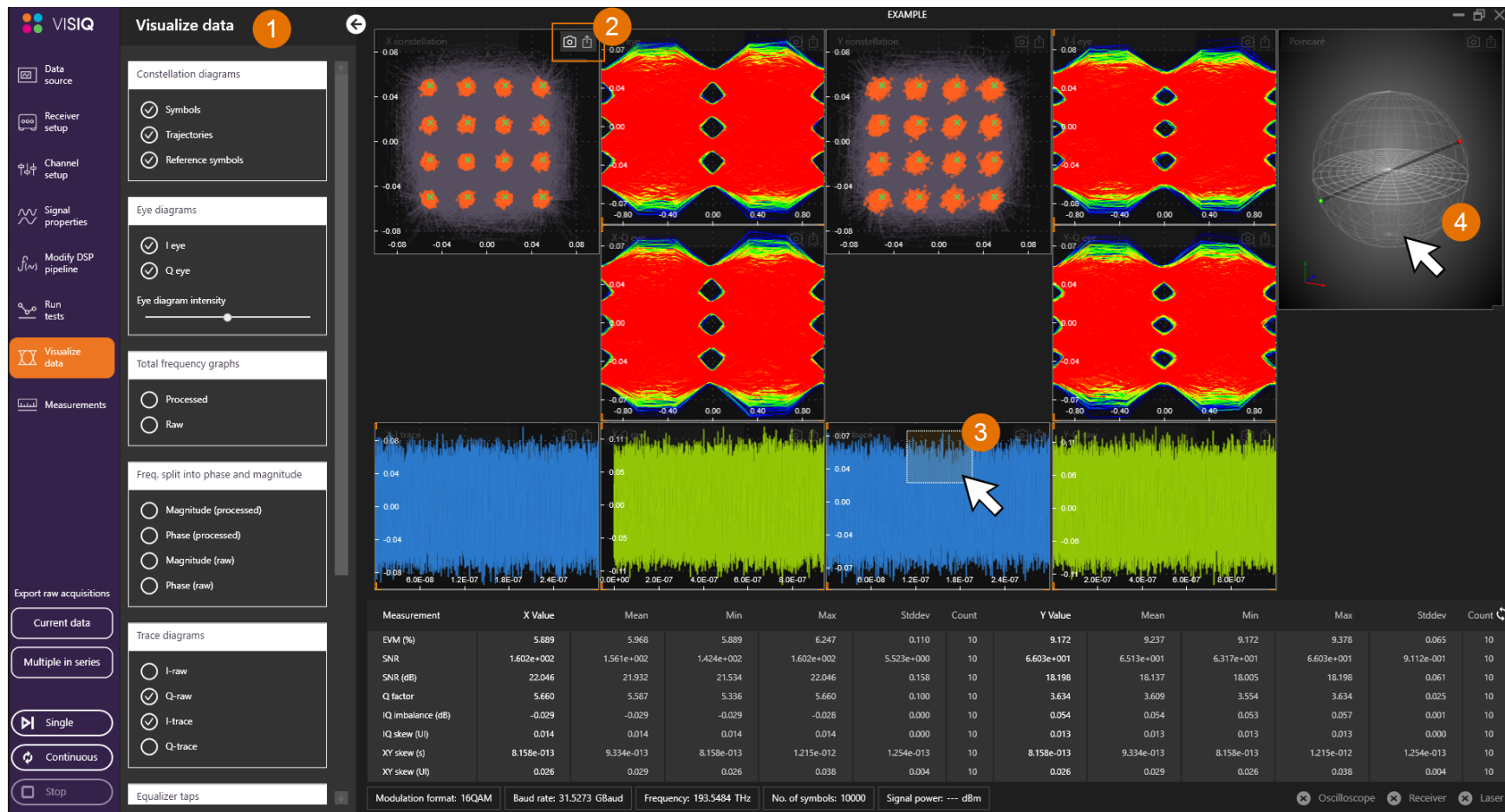
7. Click **Stop** to stop/pause the test.
8. Click **Reset error count** to reset error count to zero.

5.8 Visualize data

VISIQ can display a number of plots to visualize your signal.

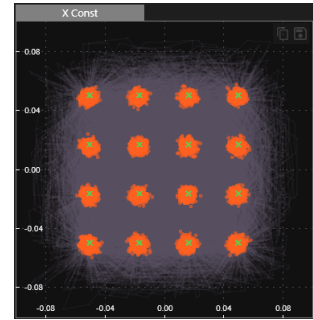
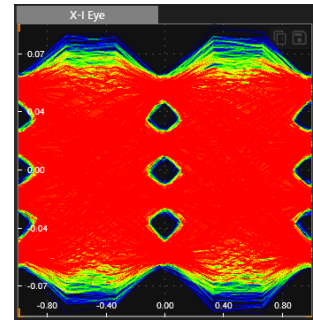
► To customize the visualization area:

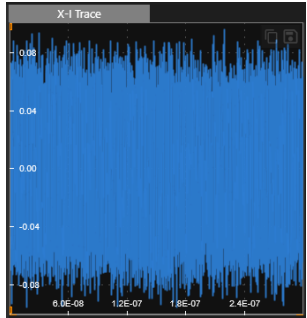
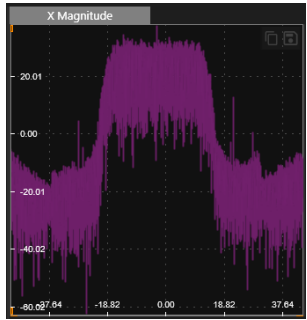
1. In **Visualize data**, select the plots to display.
2. Save a plot as an image or data file.
3. Zoom in by clicking into a black area of a plot and moving the cursor left to right. To zoom out, move the cursor right to left.
4. Rotate the Poincaré sphere by clicking and holding it with your mouse and moving it as required.

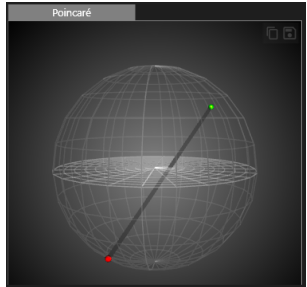


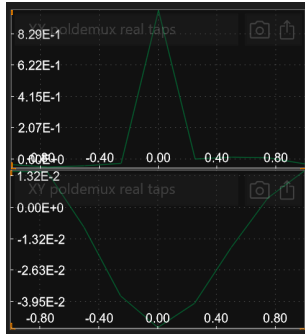
5.8.1 Plot details

VISIQ supports the following plots:

Type	Details	Plots	Example
Constellations	A constellation diagram is a graphical representation used in digital communications to visualize and analyze coherently modulated signals, particularly in the context of quadrature amplitude modulation (QAM) and phase-shift keying (PSK) modulation schemes. These diagrams are valuable tools for understanding the signal's characteristics and assessing its performance in terms of signal quality and error rate. Each of the two axes represents one of the signal's in-phase (I) and quadrature (Q) components, which are two orthogonal dimensions of the signal.	X Const Y Const	 A constellation diagram titled 'X Const' showing a 4x4 grid of 16 signal points. The points are represented as orange-red clusters on a dark background. The horizontal axis ranges from -0.08 to 0.08, and the vertical axis ranges from -0.08 to 0.08.
Eye diagrams	An eye diagram is constructed in the time domain and displays multiple overlaid waveform traces of a received signal over one or more unit intervals (UI). The horizontal axis represents time, and the vertical axis typically represents voltage or signal amplitude. A clean and wide-open eye diagram is indicative of a robust and reliable communication system.	X-I Eye X-Q Eye	 An eye diagram titled 'X-I Eye' showing multiple overlaid waveform traces. The horizontal axis represents time from -0.80 to 0.80, and the vertical axis represents signal amplitude from -0.07 to 0.07. The diagram shows a clear, wide-open eye shape, indicating good signal quality.

Type	Details	Plots	Example
Trace diagrams	A trace diagram is a graphical representation of a signal or data trace over time. It's a simple plot of signal amplitude on the y-axis and time on the x-axis. Time domain trace plots are useful for observing signal waveforms and identifying characteristics like pulse shapes, signal transitions, and timing. The Raw plots are waveforms as received from the oscilloscope before DSP, and the Trace plots show the waveform post DSP.	X-I Trace: recovered X-Q Trace: recovered X-I Raw X-Q Raw	
Frequency diagrams	A frequency or spectral diagram provides a visual representation of a signal's frequency content as a function of frequency. It is a valuable tool for understanding signal characteristics, optimizing communication systems, and diagnosing issues related to frequency interference and modulation. The raw plots show spectra before DSP, and the non-raw plots display spectra post DSP.	X Magnitude: recovered Y Magnitude: recovered X Phase: recovered Y Phase: recovered Spectrum Total: recovered X Magnitude Raw Y Magnitude Raw X Phase Raw Y Phase Raw Spectrum Total: raw	

Type	Details	Plots	Example
Polarization diagrams	The Polarization diagram or Poincaré sphere is a three-dimensional plot that serves as a geometric representation of the polarization state of light. It represents all possible polarization states of light on its surface. Linear polarization states are mapped to points on the equator of the sphere, while circular polarization states are mapped to points at the north and south poles. Elliptical polarization states occupy positions between the equator and the poles. For dual polarization signal, there will be two points on the surface of the Poincaré sphere, placed opposite to each other as the two polarizations are orthogonal.	Poincaré sphere	
Equalizer taps	Equalizer taps are the individual filter coefficients or delay elements that are adjusted to modify the amplitude and phase of specific frequency components in a signal, with the goal of compensating for distortion and interference in communication systems. The plots display the amplitude response of the equalizer as a function of tap index. It shows how the equalizer modifies the magnitude of the signal at various frequencies. Peaks and troughs in the plot indicate the equalizer's boosting and attenuation characteristics at specific frequencies.	X Equalizer Real Taps Y Equalizer Real Taps X Equalizer Imaginary Taps Y Equalizer Imaginary Taps	

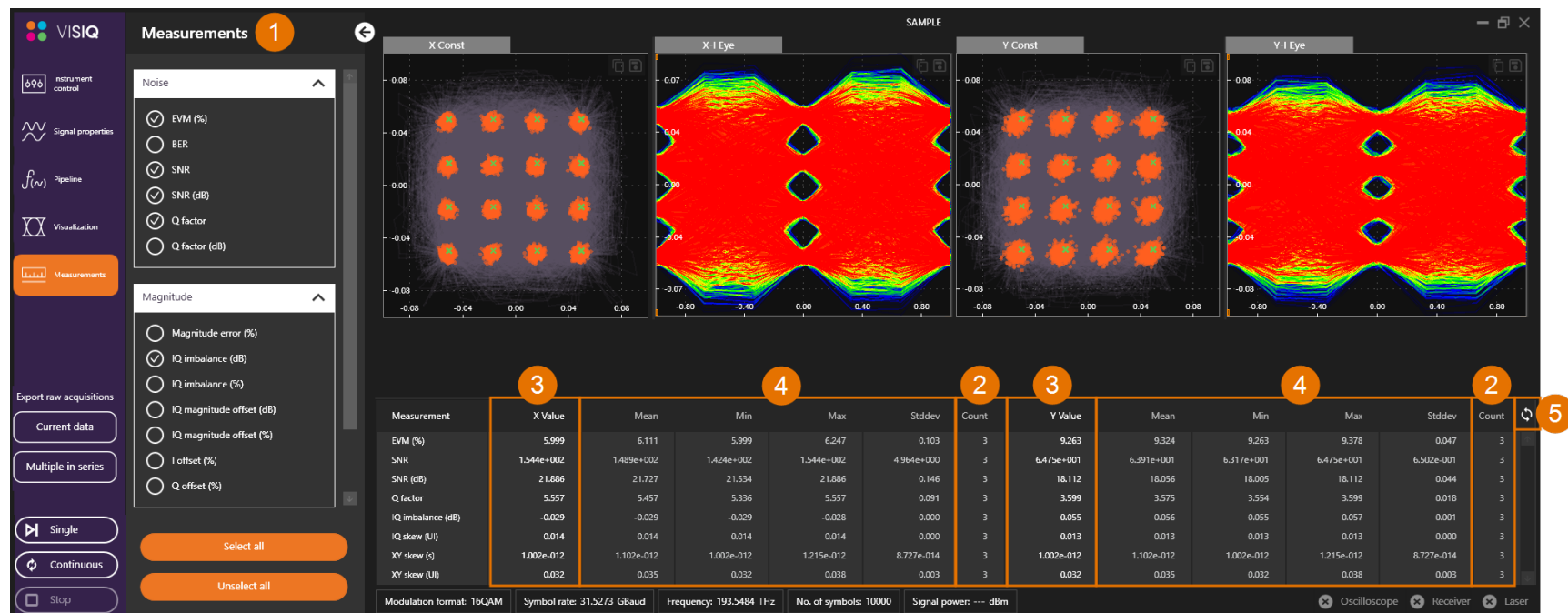
Type	Details	Plots	Example
PolDeMux taps	Polarization demultiplexer tap plots are used to visualize and analyze the characteristics of taps or coefficients in a polarization demultiplexer. These plots provide insights into how the demultiplexer separates or routes different polarization states of light. The vertical axis of the plot represents the magnitude (amplitude) of the tap coefficients which indicates the amount of light energy being directed or routed into specific polarization states or by the demultiplexer.	XX PolDeMux Real Taps YY PolDeMux Real Taps XY PolDeMux Real Taps YX PolDeMux Real Taps XX PolDeMux Imaginary Taps YY PolDeMux Imaginary Taps XY PolDeMux Real Taps YX PolDeMux Real Taps	

5.9 Select measurements

VISIQ can perform a comprehensive range of measurements and display selected values in the measurement table in the UI. The table values will update with each data acquisition.

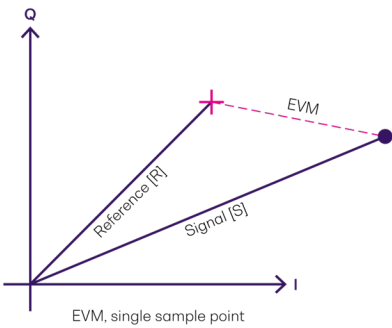
► To customize the measurement table:

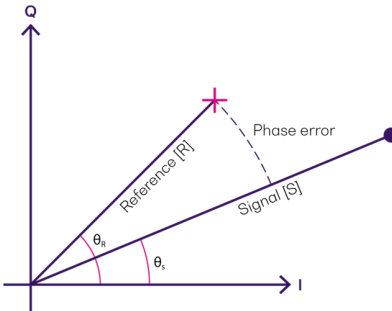
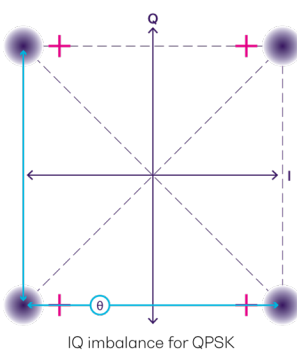
1. In **Measurements**: Select the measurements to display in the table.
2. **Count** columns list the number of data acquisitions that are reflected in the table.
3. Columns **X Value** and **Y Value** display actual measurements as per the latest acquisition.
4. Values in columns **Mean**, **Min**, **Max** and **Stddev** reflect values across acquisitions.
5. Reset the table.

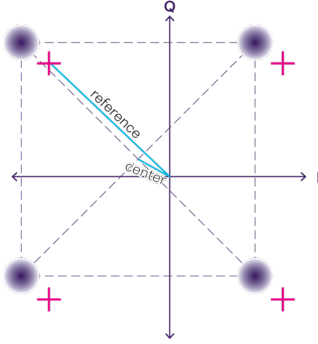
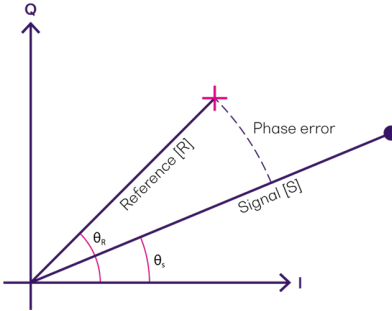


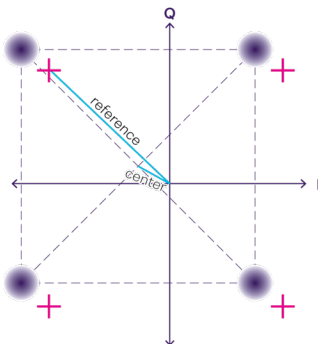
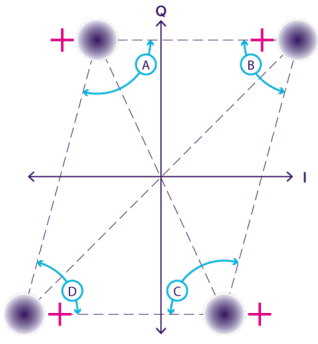
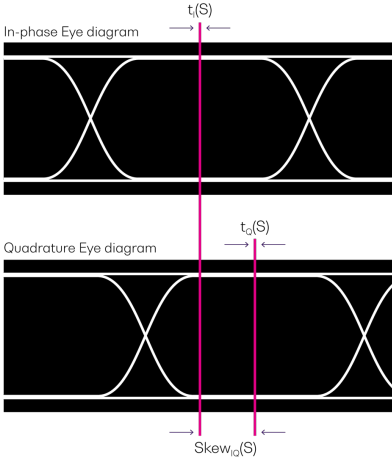
5.9.1 Measurement details

VISIQ can perform the following measurements:

Type	Measurement	Details	
Noise	EVM (%)	The magnitude of the error vector – the difference between the signal vector and the ideal reference vector. rms - root mean square $EVM (\%) = 100 \times \frac{rms[signal - reference]}{\max [reference]}$	 EVM, single sample point
	BER estimate	An estimation of the bit error rate calculated from the Q factor. Time efficient alternative comparing to an actual bit counting. $BER_{est} = \frac{1 - \frac{1}{L}}{\log_2 L} \operatorname{erfc} \left(\frac{Q \text{ factor}}{\sqrt{2}} \right)$ L = The maximum number of levels in the signal. erfc = The complimentary error function – applies the assumption of the Gaussian distribution of constellation points.	
	SNR	Signal-to-Noise Ratio expressed as a ratio. $SNR = \frac{Signal \text{ Power}}{Noise \text{ Power}}$	
	SNR (dB)	Signal to noise ratio expressed as in decibels.	
	Q factor	An alternate representation of the Signal-to-Noise Ratio - Q factor - in percentage.	

Type	Measurement	Details	
		$Q\ factor_{(dB)} = \sqrt{\frac{3 \log_2 L}{L^2 - 1} \frac{2}{(EVM\%/100)^2 \log_2 2^{bits_per_symbol}}}$ L = The maximum number of levels in the signal.	
	Q factor (dB)	An alternate representation of the Signal-to-Noise Ratio - Q factor – in dB. $Q\ factor_{(dB)} = 20 \log_{10}(Q\ factor)$	
Magnitude	Magnitude error (%)	The difference in amplitude between the reference symbol and the signal vector. $\frac{Magnitude\ Error_{RMS}}{Magnitude\ Error} = \frac{ S - R }{\sqrt{\frac{1}{N}(Magnitude\ Error_1^2 + \dots + Magnitude\ Error_N^2)}}$	
	IQ imbalance (dB)	The ratio of the in-phase component vs. the quadrature component of the constellation points in dB.	
	IQ imbalance (%)	The ratio of the in-phase component vs. the quadrature component of the constellation points in percentage.	 IQ imbalance for QPSK

Type	Measurement	Details	
	IQ magnitude offset (dB)	The center of the constellation with reference to the ideal center point in decibels.	
	IQ magnitude offset (%)	The center of the constellation with reference to the ideal center point in percentage.	
	I offset (%)	The center of the In- phase component of the constellation with reference to the ideal center point in percentage.	
	Q offset (%)		
Phase	Phase error (rad)	The phase difference between the signal vector and the ideal reference vector. $Phase\ Error_{RMS} = \frac{Phase\ Error = S - R }{\sqrt{\frac{1}{N}(Phase\ Error_1^2 + \dots + Phase\ Error_N^2)}}$	

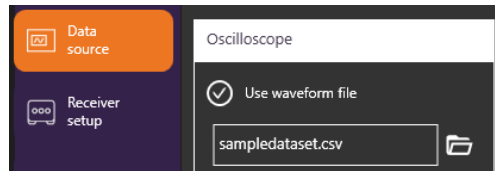
Type	Measurement	Details	
	IQ phase offset (rad)	The phase of the vector from the center of the constellation with reference to the ideal center point in radians.	
	IQ phase error (deg)	The measurement of the angle between the in-phase and quadrature components of the signal. Every deviation from 90° between the in-phase and the quadrature is an IQ Phase Imbalance. Quad Error is the average measurement within the constellation points. $IQ\ Phase\ Imbalance = \frac{\angle A - \angle B + \angle C - \angle D}{4}$	
Time	IQ skew (s)	The time difference between the in-phase and quadrature eye centers in seconds. $Skew_{IQ}(s) = t_I(s) - t_Q(s)$	

Type	Measurement	Details	
	IQ skew (UI)	Time difference between the in-phase and quadrature eye centers in UI.	
	XY skew (s)	The temporal skew between the X and Y polarization symbols.	
	XY skew (UI)	The temporal skew between the X and Y polarization symbols, measured in the percentage of the unit interval of the symbol.	
	PMD (s)	The polarization mode dispersion, shared between X Pol and Y Pol results.	
Frequency	Frequency offset (Hz)	The frequency offset of the carrier.	
	Line width (Hz)	The linewidth estimation of the carrier and local oscillator.	
	Baudrate (Hz)	Baud rate of the received signal.	

5.10 Load and save data

VISIQ offers several options to load and save data.

- ▶ To load previously saved data:
 - Load an existing project file in the start-up screen.
 - Load a waveform file in **Data source**.



Waveform files include raw signal data only. Project files include both, raw signal data and settings that were selected when saving the file.

- ▶ To export currently cached raw data:
 1. Click **Current data** to save the data in a .csv file. File names include the date and time of saving.
- ▶ To export a series of acquisitions:
 2. Click **Multiple in series**.
 3. Enter the number of acquisitions to export and confirm.

When running the analysis by clicking **Single** and/or **Continuous**, VISIQ will save a .csv file per acquisition as per the set number of acquisitions. Files are provided in a .zip file.



- ▶ To create a new project file:
- Select **New project** in the start-up screen.
 - Click the VISIQ logo and select **New**.

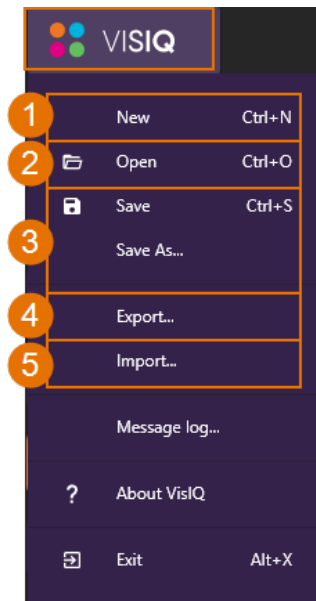
5.11 System menu functions

To access further options and relevant system information, click the **VISIQ logo**. You can:

- Load and save data.
- Open the message log.
- Open the About VISIQ window.

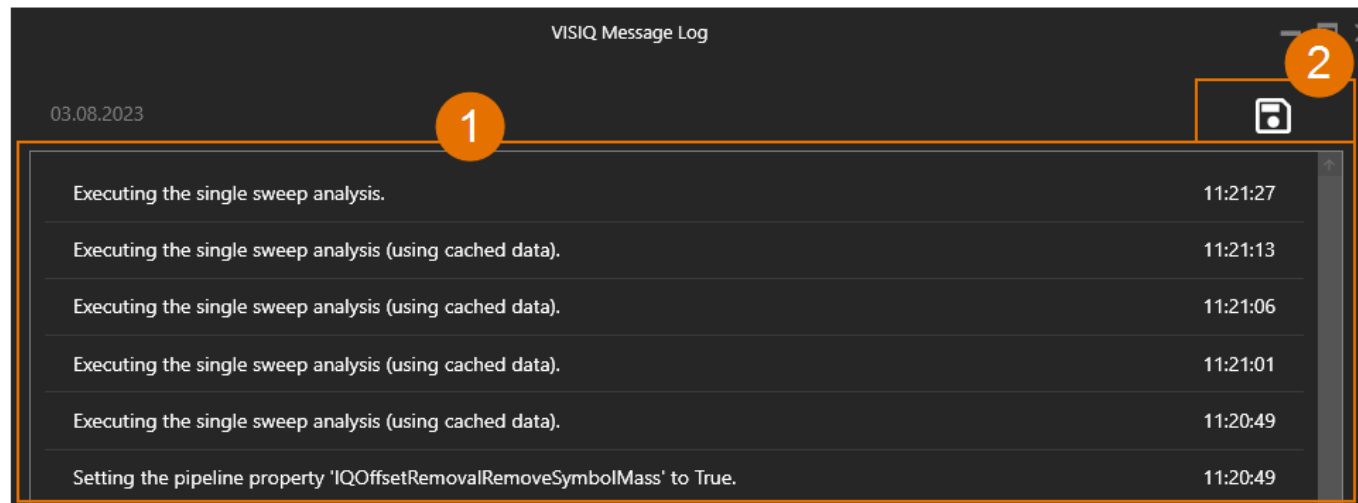
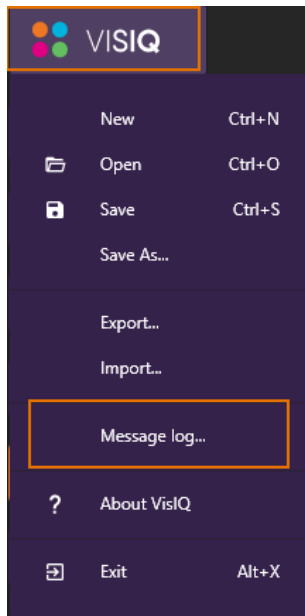
5.11.1 Load and save data

1. Create a new project file.
2. Load a project file or configuration file.
3. Save a project file or configuration file.
4. Save a pipeline file.
5. Load a pipeline file.



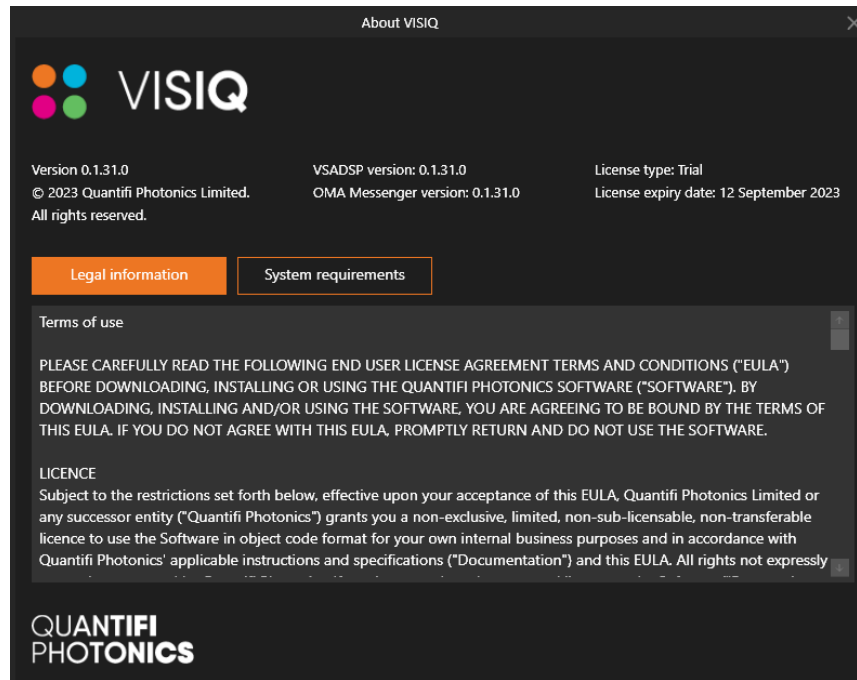
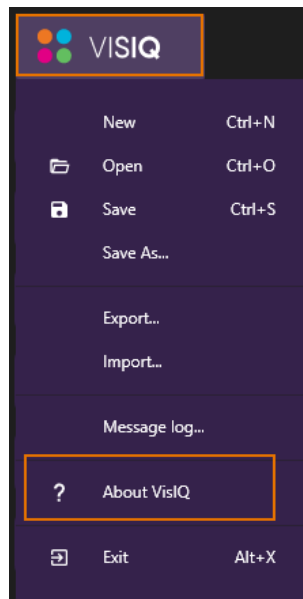
5.11.2 Message log

1. The message log contains a record of your activity in VISIQ, including remote control commands.
2. You can save the message log as a .txt file.



5.11.3 VISIQ information in About window

The About window contains information about your VISIQ version, license expiry date and other relevant information.



6 Working with SCPI commands

6.1 SCPI overview

VISIQ supports test automation via SCPI remote control.

Remote communication with the SCPI service is achieved through the Standard Commands for Programmable Instruments (SCPI). Support for VISA I/O API over TCP/IP is provided by the VXI-11 compliant VISIQSCPI service. With VISA communication drivers installed on the client, the implementation of VISA programming within environments such as MATLAB becomes available.

For an overview, refer to [Programming conventions](#) and [Status and event registers](#).

For details on available SCPI commands, refer to:

- [SCPI commands summary](#)
- [SCPI commands descriptions](#)

For programming examples, refer to [SCPI programming examples](#) and to example files in the folder ...\\Quantifi Photonics\\VisIQ\\Examples.

For an overview of programming applications, refer to [Programming applications](#).

6.1.1 Programming conventions

This section details the programming and measurement conventions to follow while executing the commands for the VISIQSCPI service.

Parameter	Default Unit	Alternative Units
Record length	Pts	
Sampling rate	GS/s	
Bandwidth	GHz	
Power	DBM	
Wavelength	M	NM, PM
Scale	mV/Div	
Baudrate	BAUD	KBAUD, MBAUD, GBAUD
Frequency	HZ	THZ, GHZ, MHZ, KHZ
Skew	s	
DC offset	V	MV, KV

Argument	Data Format
<wsp>	Specifies whitespace character (01 ₁₆ – 09 ₁₆ , 0B ₁₆ – 20 ₁₆).
<value>	Is numerical data, an integer, a decimal, exponential (10e-9 or 5.8e6) or string.
[VALUE1 VALUE2]	A parameter choice. The ' ' separates the unique parameters available, only one of the choices can be used. In the example, either the input parameter [VALUE1] or [VALUE2] can be used, but not both. Some commands may have more than two choices available. This parameter can be omitted where the command has a default defined in the command description.

6.1.2 Status and event registers

6.1.2.1 Standard Event Status Register

The Standard Event Status Register (SESR) is modified by the Quantifi Photonics product with the results of the command operations.

Bit	Description
7 (MSB), 6	Not used
5	Is set when a Command Error event has been detected
4	Is set when a command Execution Error has been detected
3	Is set when a Device Dependent Error event has been detected
2	Is set when there a Query Error event has been detected
1	Not used
0 (LSB)	Is set when an Operation Complete event has been generated

6.1.2.2 Standard Event Status Enable Register (Mask)

The Standard Event Status Enable Register (SESR Mask) is used to build the Event Status Bit (ESB) within the Status Byte Register (STB). To ignore any of the events detected and set in the SESR, set the corresponding bit within the SESR Mask to 0. The STB can then be queried and the value of the ESB can be used to determine service request requirements based on the SESR Mask applied.

NOTE

The 0 (LSB) value within the SESR Mask is 0.

6.1.2.3 Status Byte Register

The Status Byte Register (STB) is built from all other status registers and masks. This register can be used in queries to determine if an event has been detected and where that event has been detected.

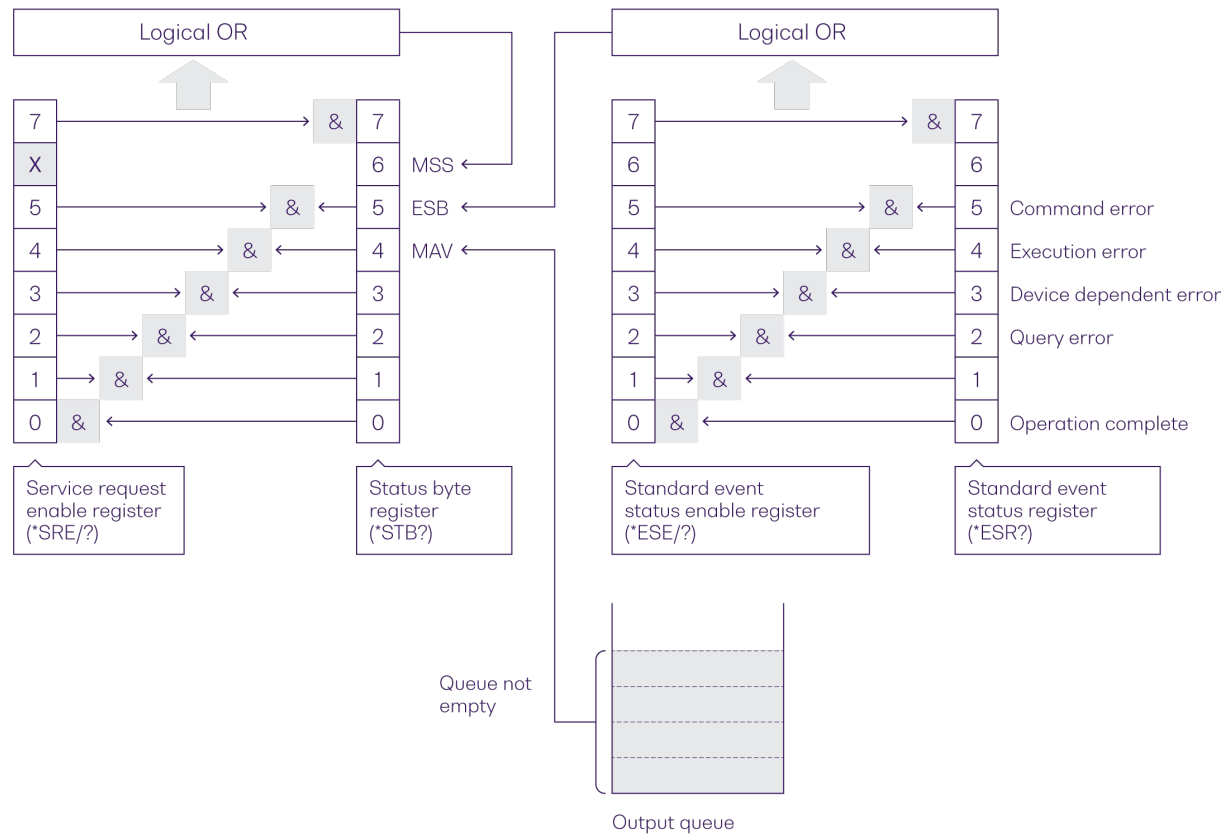
Bit	Description
7 (MSB)	Not used
6	The Master Summary Status (MSS) bit is set from the STB and SRE Mask
5	The Event Status Bit (ESB) is set from the SESR and the SESR Mask
4	Message Available (MAV) is set when there is data in the output queue
3, 2, 1, 0 (LSB)	Not used

6.1.2.4 Service Request Enable Register (Mask)

The Standard Request Enable Register (SRE Mask) is used to build the Master Summary Status Bit (MSS) within the Status Byte Register (STB). To ignore any of the events detected and set in the STB register itself, set the corresponding bit within the SRE Mask to 0. The STB can then be queried and the value of the MSS can be used to determine the type of service request required based on the SRE Mask applied.

Bit	Description
7 (MSB)	Not used
6	The Master Summary Status (MSS) bit is set from the STB and SRE Mask
5	The Event Status Bit (ESB) is set from the SESR and the SESR Mask
4	Message Available (MAV) is set when there is data in the output queue
3, 2, 1, 0 (LSB)	Not used

6.1.2.5 Status and event registers diagram



6.2 SCPI commands summary

Find summaries of SCPI commands in the following sections. For your reference, we have included relevant parts of the VISIQ user interface.

6.2.1 Common commands

Click [>>](#) in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Common commands](#).

Command	Description
*IDN?	Query the software identification >>
*OPT	Query the application >>
*ESR?	Query the Standard Event Status Register >>
*ESE?	Query the Standard Event Status Enable Register (Mask) >>
*ESE	Set the Standard Event Status Enable Register >>

6.2.2 System menu functions

6.2.2.1 Commands

Click >> in the summary table below to navigate to the detailed command description, or refer to command descriptions in [System menu functions](#).

Command	Description
:APP	
:STATE?	Query the state of the VISIQ application >>
:CONNeCT	Connect to the VISIQ application >>
:DISConnect	Disconnect from the VISIQ application >>
:EXIT	Exit VISIQ with or without saving the project >>

6.2.3 Select the data source

Connect VISIQ with an oscilloscope of your test setup to analyze live data, or load a previously acquired waveform file.

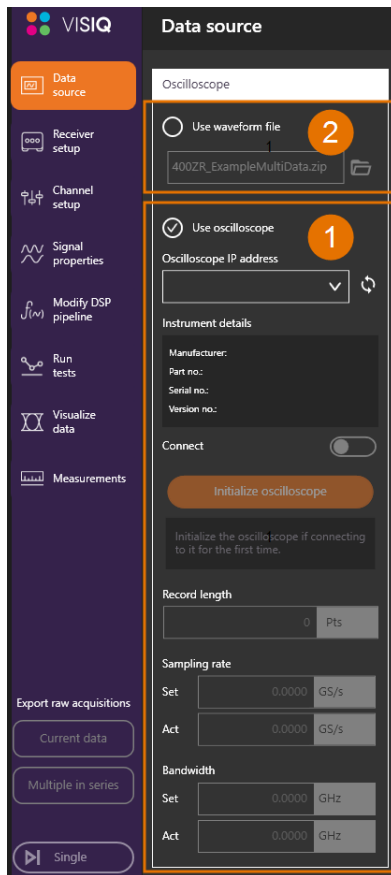
Example waveform files are included in the VISIQ software package in folder ...\\Quantifi Photonics\\VisIQ\\Examples\\Projects.

6.2.3.1 Commands

Click >> in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Select the data source](#).

Command	Description
:PROJect	
:SOURce	
:DATA	
:TYPE?	Query the selected data source >>
:TYPE	Select the data source: An oscilloscope of your test setup or a previously acquired waveform file >>
:SCOPE?	Query available scopes on the network >>
:SCOPE	
:STATe?	Query oscilloscope connection state >>
:CONNect	Connect with the selected oscilloscope >>
:DISConnect	Disconnect from an oscilloscope >>
:INITialize	Initialize the connected oscilloscope >>
:IDN?	Query the connected oscilloscope ID >>
:RECOrd_LENGTH?	Query the oscilloscope record length >>
:SAMPLing_RATE?	Query the oscilloscope sampling rate >>
:SAMPLing_RATE	Set the oscilloscope sampling rate >>
:BANDwidth?	Query the oscilloscope bandwidth >>
:BANDwidth	Set the oscilloscope bandwidth >>

6.2.3.2 UI reference



► To select the data source:

1. Connect with an oscilloscope, or
2. Load a waveform file.

Parameter	Description
Record length	The number of data points to be captured by the oscilloscope for a single acquisition.
Sampling rate	The sampling rate of the oscilloscope. If the Set value is not accepted by the oscilloscope, it will snap to an acceptable value and be displayed in Act (actual value).
Bandwidth	Sets the bandwidth of all of the oscilloscope channels in use. If the Set value is not accepted by the oscilloscope, it will snap to an acceptable value and be displayed in Act (actual value).

6.2.4 Connect VISIQ with the coherent receiver

This is required if you have selected an oscilloscope as your data source to acquire live data.

This section describes how to connect a Quantifi Photonics IQRX receiver.

If you're using a different, non-Quantifi Photonics coherent receiver, connect the receiver in the CoRx node of the DSP pipeline instead. For details, refer to **Configure the DSP pipeline > Node: CoRx**.

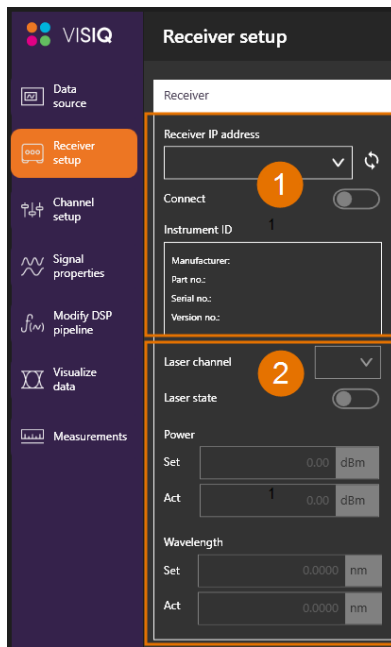
6.2.4.1 Commands

Click [>>](#) in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Connect VISIQ with the coherent receiver](#).

Index <m>: The channel index of the coherent receiver, this is an integer <1 to 4> depending on the model.

Command	Description
:PROJect	
:SOURce	
:RECeiver	Select a coherent receiver >>
:FIND?	Query the selected data source >>
:STATe?	Query receiver connection state >>
:CONNect	Connect with a coherent receiver >>
:DISConnect	Disconnect the coherent receiver >>
:IDN?	Query the connected coherent receiver ID >>
:CHANnel<n>	
:STATe?	Query the laser state >>
:STATe	Set laser state to ON or OFF >>
:POWer?	Query the laser power >>
:POWer	Set laser power >>
:WAVElength?	Query the laser wavelength >>
:WAVElength	Set laser wavelength >>

6.2.4.2 UI reference



► To connect with a Quantifi Photonics IQRX receiver:

1. Identify the IQRX.
2. Control an IQRX laser.

Parameter	Description
Power	Output power of the selected laser.
Wavelength	Output wavelength of the selected laser.

6.2.5 Set channels

Set channel mapping to link the coherent receiver tributaries (XI, XQ, YI, YQ) to the oscilloscope channels.

6.2.5.1 Commands

Click >> in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Set channels](#).

Index <p>: The channel index of the oscilloscope, this is an integer <1 to 4> depending on the model.

Command	Description
:PROJect	
:SOURce	
:DATA	
:CHANnel<n>	
:MAPPing?	Query channel mapping of the oscilloscope >>
:MAPPing	Set channel mapping of the oscilloscope >>
:SCALE?	Query the vertical scaling of the oscilloscope channels >>
:SCALE	Set vertical scaling of oscilloscope channels >>

6.2.5.2 UI reference



- To set channels:
1. Set channel mapping.
 2. Set the oscilloscope vertical scale.

6.2.6 Configure signal properties

VISIQ can process single and dual polarization signals.

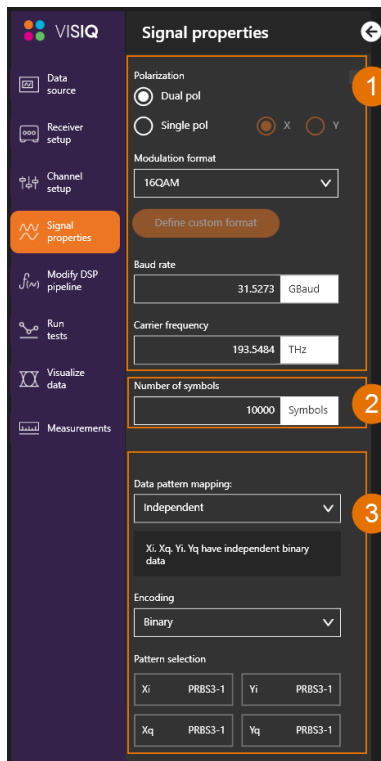
For details on setting test parameters and relevant SCPI commands, refer to [Run tests](#).

6.2.6.1 Commands

Click [>>](#) in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Configure signal properties](#).

Command	Description
:PROJect	
:SOURce	
:DATA	
:SIGNal	
:POLarization?	Query the selected polarization type of the signal >>
:POLarization	Set the polarization type of the signal >>
:AXIS?	Query the polarization axis for single polarization >>
:AXIS	Set the polarization axis for single polarization >>
:MODUlation?	Query the modulation format >>
:MODUlation	Set the modulation format >>
:INFO?	Query supported modulation formats >>
:BAUDrate?	Query the Baud rate of the signal >>
:BAUDrate	Set the Baud rate of the signal >>
:FREQuency?	Query the carrier frequency >>
:FREQuency	Set the carrier frequency >>
:SYMbolsNumber?	Query the number of symbols >>
:SYMbolsNumber	Set the number of symbols >>
:POWer?	Query signal power >>

6.2.6.2 UI reference



► To configure signal properties:

1. Define the properties of the received signal.
2. Set the total number of symbols to capture during data acquisition.
3. When running error detection testing, select test parameters. For details, refer to [Run tests](#).

Parameter	Description
Modulation format	Modulation format of the received signal.
Baud rate	Baud rate of the received signal.
Carrier frequency	Frequency of the optical carrier
Number of symbols	The total number of symbols to capture during data acquisition. The default value of 5,000 is a good starting point, adjust it as required. A greater number results in slower processing.
Data pattern mapping	Select data pattern mapping option: Independent: The four tributaries are individually modulated with PRBS or data patterns. Multiplexed IQ: The I and Q tributaries of the same polarization are multiplexed and modulated with a single PRBS or data pattern per polarization. Multiplexed XY: All four tributaries are multiplexed and modulated with a single PRBS or data pattern.
Encoding	Select decoder settings. Binary: The symbol's I and Q bits are encoded in an incremental order on each axes. Gray: The binary values are arranged in such a way that only one bit changes between adjacent symbols.
Pattern selection	Select the test pattern.

6.2.7 Configure the DSP pipeline

Configure your DSP pipeline and its nodes as required:

- [Pipeline configuration](#)
- [Node: CoRx](#)
- [Node: CoRx auto calibrate](#)
- [Node: Filter](#)
- [Node: Retimer](#)
- [Node: PolDeMux](#)
- [Node: Equalizer](#)
- [Node: Phase estimator](#)
- [Node: MATLAB](#)
- [Node: Nonlinear correction](#)
- [Node: Chromatic dispersion](#)
- [Node: IQ offset removal](#)

6.2.7.1 Pipeline configuration

You configure your DSP pipeline in VISIQ by adding, moving and/or removing configurable nodes into position as required.

- The minimum configuration of the pipeline includes: **CoRx**, **CoRx auto calibrate**, **Retimer**, **PolDeMux**, **Carrier Offset**, **Phase estimator**.
- Some nodes can appear in the pipeline only once; they will then appear grayed out in the **Add nodes** column in the UI.

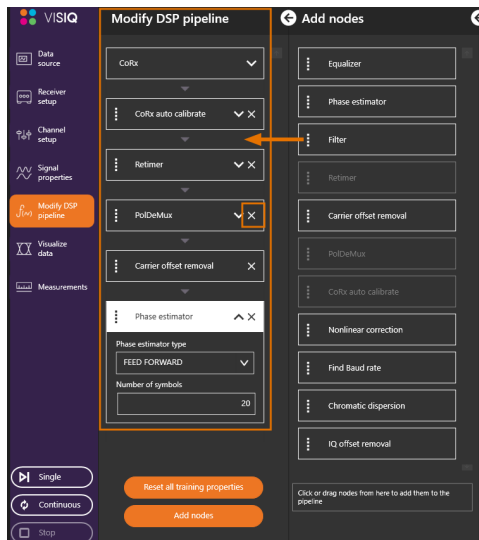
Commands

Click **>>** in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Pipeline configuration](#).

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	Description
:PROJect	
:PIPELine	
:CONFIguration	
:ADD	Add a node to the DSP pipeline in a specific location >>
:MOVE	Move a node to a different location in the DSP pipeline >>
:REMOve	Remove a node from the DSP pipeline >>
:TRAIning	
:ReSeT	Return training properties of all nodes in the DSP pipeline that support training to default values >>
:NODEs?	Query the configuration of the DSP pipeline >>

UI reference



► To configure the pipeline:

1. Add, move and/or remove nodes.
2. Configure node settings.
3. You can reset training properties for relevant nodes: **CoRx auto calibrate**, **Equalizer** and **Nonlinear correction**.

6.2.7.2 Node: CoRx

- ☒ always the first node in the pipeline
- ☒ cannot be removed

This node compensates coherent receiver imperfections such as skew, frequency response, and IQ phase error.

Relevant receiver settings are stored in a settings file (.crs format). Example settings files are included in the VISIQ software package, locate them in ...\\QuantifiPhotonics\\VisIQ\\Examples\\Configurations.

When creating a settings file:

- Limit the size of the arrays to 1,000.
- Limit the number of decimal places so that the maximum line length is less than 10,000 characters.
- When using an example settings file that has been supplied with the VISIQ software package: Edit a copy of the file, leave the original file unchanged.

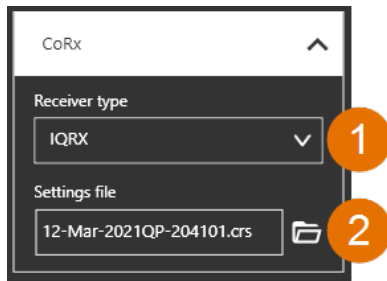
Commands

Click [>>](#) in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Node: CoRx](#).

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	Description
:PROJect	
:PIPELine	
:NODEs	
:CoRx<n>	
:RECeiver?	Query the receiver type >>
:RECeiver	Set the receiver type >>
:SETTings?	Query the location of the .crs settings file to be used with the receiver >>
:SETTings	Define the location of the .crs settings file to be used with the receiver >>

UI reference



► To define the **CoRx** node settings:

1. Select the receiver type.
2. Connect with a settings file.

6.2.7.3 Node: CoRx auto calibrate

This node manages automatic calibration of the coherent receiver. It calculates and compensates for the DC offsets and the gain for the XI/XQ pair and the YI/YQ pair. It also calculates residual skew between the XI/XQ pair and the YI/YQ pair in the system.

This node is ideal for fine-tuning coherent receiver calibration. If large adjustments are required, please correct these in the CoRx node.

Commands

Click >> in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Node: CoRx auto calibrate](#).

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	Description
:PROJect	
:PIPELine	
:NODEs	
:coRXCaLibrate<n>	
:GRADient?	Query gradient >>
:GRADient	Set the gradient >>
:SKEW?	Query skew >>
:SKEW	Set skew >>
:GAIN?	Query gain >>
:GAIN	Set gain >>
:DCOffset?	Query DC offset >>
:DCOffset	Set DC offset >>
:TRAiNing	
:ACQuisitions?	Query the number of training acquisitions >>
:ACQuisitions	Set the number of training acquisitions >>
:CONTinuous?	Query the status of continuous training acquisitions >>
:CONTinuous	Turn continuous training acquisitions ON or OFF >>
:HOLD?	Query training acquisitions hold >>
:HOLD	Hold training acquisitions >>
:COUNT?	Query training acquisition count >>

Command	Description
:COUNT	
:ReSeT	Reset training acquisitions count >>

UI reference

The screenshot shows the 'CoRx auto calibrate' interface. It is divided into several sections: 'Gradient', 'Skew', 'Gain', 'DC offset', and 'Training acquisitions'. Callout 1 points to the 'Gradient' input field, which is set to 0.300000. Callout 2 points to the 'Gain' section, which contains four channels (Ch 1 to Ch 4) with values ranging from 1.0000E+000 to 9.1167E-001. Callout 3 points to the 'Training acquisitions' section, which shows a value of 20 and a 'Training counter: 14'. At the bottom, there are radio buttons for 'Continuous training' and 'Hold', and a 'Reset' button.

- ▶ To select the settings for the **CoRx auto calibrate** node:
 1. Enter **Gradient**.
 2. Adjust **Skew**, **Gain** and **DC offset** for each channel as required.
 3. (optional) Configure the training setup.

6.2.7.4 Node: Filter

This node works as a digital filter that modifies the frequency components present in the signal.

Commands

Click [>>](#) in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Node: Filter](#).

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	Description
:PROJect	
:PIPEline	
:NODEs	
:FILTer<n>	
:TYPE?	Query filter type >>
:TYPE	Set filter type >>
:BANDwidth?	Query bandwidth >>
:BANDwidth	Set bandwidth >>
:CENTer?	Query center >>
:CENTer	Set center >>
:ROLLoff?	Query roll off >>
:ROLLoff	Set roll off >>

UI reference

Filter

Filter type

RRC

Bandwidth

31.5273 GHz

Center

0.0000 GHz

Roll off

0.2000

► To set the filter:

1. Set **Filter type**.
2. Set filter properties.

Parameter	Description
Bandwidth	Controls the filter's 3dB bandwidth.
Center	Sets the central frequency between the upper and the lower cutoffs.
Roll off	Controls the steepness of the frequency transfer function. Available for filter types RRC , RCOSINE , BRICKWALL , and SINC .
Order	The order of the filter. Available for filter types BESSEL and BUTTERWORTH .

6.2.7.5 Node: Retimer

- ☒ only 1 per pipeline
- ☒ insert after **Find Baud rate** node (if used)
- ☒ cannot be removed

This node retimes the input to a particular sample per Baud.

Interpolation is used to create traces of samples per Baud higher than the input. The algorithm also calculates the skew between x pol and y pol that may be present.

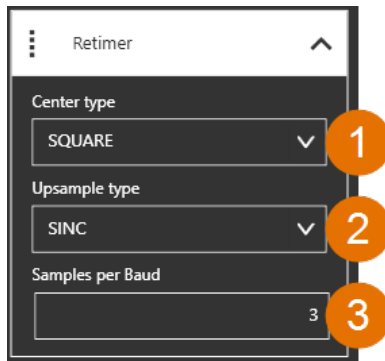
Commands

Click [>>](#) in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Node: Retimer](#).

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	Description
:PROJect	
:PIPEline	
:NODEs	
:RETiMer<n>	
:CENTer?	Query center type >>
:CENTer	Set center type >>
:UPSaMple?	Query upsample type >>
:UPSaMple	Set upsample type >>
:SAMPles?	Query samples per Baud >>
:SAMPles	Set samples per Baud >>

UI reference



► To set the retimer properties:

1. Select **Center type**.
2. Select **Upsample type**.
3. Enter **Samples per Baud**.

Parameter	Description
Center type	The type of algorithm that will be used to find the symbol centers. SQUARE : Square timing recovery STDDEV : Queries the time location of the eye diagram with the largest standard deviation. CROSSPOINT : Queries the crossing points of the eye diagram and calculates the center as the distance between these crossing points. EVM : Error Vector Magnitude. The center location is the minimum EVM across the eye diagram.
Upsample type	The up-sampling technique that will be used to interpolate the inputs. LINEAR : Linear interpolation SINC : A sinc function interpolation
Samples per Baud	The number of samples per Baud to retime your signal to.

6.2.7.6 Node: PolDeMux

This node demultiplexes the two polarizations of the carrier.

Commands

Click [>>](#) in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Node: PolDeMux](#).

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	Description
:PROJect	
:PIPEline	
:NODEs	
:PolDeMuX<n>	
:TAPsNumber?	Query the number of taps >>
:TAPsNumber	Set the number of taps >>
:SYMbolsNumber?	Query the number of symbols >>
:SYMbolsNumber	Set the number of symbols >>
:GRADient?	Query the gradient >>
:GRADient	Set the gradient >>
:TAPsperBaud?	Query taps per Baud >>
:TAPsperBaud	Set taps per Baud >>
:ReSeT?	Query the resetting of each capture status >>
:ReSeT	Activate or deactivate resetting of each capture >>

UI reference

The screenshot shows the PolDeMux settings panel with the following fields and callouts:

- 1** Number of taps: 9
- 2** Number of symbols: 10000
- 3** Gradient: 0.0007
- 4** Taps per Baud: 3
- 5** Reset each capture: ☐

► To set the PolDeMux properties:

1. Enter **Number of taps**.
2. Enter **Number of symbols**.
3. Enter **Gradient**.
4. Enter **Taps per Baud**.
5. Untick **Reset each capture** to use results from previous acquisitions.

Parameter	Description
Number of taps	The number of taps. This value must be an odd value.
Number of symbols	Defines the length of the training sequence by the number of symbols. Make sure that the value is not higher than the Number of symbols in Signal properties .
Gradient	Defines the convergence rate of the adaptation. Setting values close to the minimum of 1.0e-6 will result in longer convergence time over the training symbols, but better results in presence of noise, and vice versa for values close to the maximum value of 1.0.
Taps per Baud	The number of taps per Baud.

6.2.7.7 Node: Equalizer

This node calculates the complex (real and imaginary) Finite Impulse Response (FIR) taps. It attempts to correct the frequency response of the signal by training over a defined number over symbols. The training goal is to match the ideal reference symbols.

Commands

Click [>>](#) in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Node: Equalizer](#).

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	Description
:PROJect	
:PIPELine	
:NODEs	
:EQUALizer<n>	
:TAPsNumber?	Query the number of taps >>
:TAPsNumber	Set the number of taps >>
:SYMbolsNumber?	Query the number of symbols >>
:SYMbolsNumber	Set the number of symbols >>
:GRADient?	Query the gradient >>
:GRADient	Set the gradient >>
:TAPsperBaud?	Query taps per Baud >>
:TAPsperBaud	Set taps per Baud >>
:TRAIning	
:ACQuisitions?	Query the number of training acquisitions >>
:ACQuisitions	Set the number of training acquisitions >>
:CONTInuous?	Query the status of continuous training acquisitions >>
:CONTInuous	Turn continuous training acquisitions ON or OFF >>
:HOLD?	Query training acquisitions hold >>
:HOLD	Hold training acquisitions >>
:COUNT?	Query training acquisition count >>
:COUNT	
:ReSeT	Reset training acquisitions count >>

UI reference

Equalizer

Number of taps: 99

Number of symbols: 10000

Gradient: 0.000100

Taps per Baud: 3

Training acquisitions: 18

Training counter: 0

☐ Continuous training

☒ Hold

Reset

► To set the equalizer properties:

1. Set properties: **Number of taps**, **Number of symbols**, **Gradient**, **Taps per Baud**.
2. (optional) Configure the training setup.

Parameter	Description
Number of taps	The number of taps in the equalizer kernel. This value must be an odd value.
Number of symbols	Defines the length of the training sequence by the number of symbols.
Gradient	Defines the convergence rate of the adaptation. Setting values close to the minimum of 1.0e-6 will result in longer convergence time over the training symbols, but better results in presence of noise, and vice versa for values close to the maximum value of 1.0.
Taps per Baud	The number of taps per Baud. Set this value to the number of samples per Baud of your streams at the input into the node.

6.2.7.8 Node: Phase estimator

This node calculates and removes the phase errors over time that may be present in the signal. The phase error occurs as a result of the phase noise difference between the carrier and the local oscillator laser.

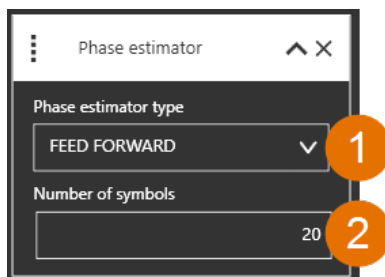
Commands

Click **>>** in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Node: Phase estimator](#).

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	Description
:PROJect	
:PIPEline	
:NODEs	
:PHASeestimator<n>	
:TYPE?	Query the type of algorithm used >>
:TYPE	Set the type of algorithm used >>
:SYMBOLsNumber?	Query the number of symbols >>
:SYMBOLsNumber	Set the number of symbols >>

UI reference



► To set the phase estimator properties:

1. Select **Phase estimator type**.
2. Enter **Number of symbols**.

Parameter	Description
Phase estimator type	The type of the algorithm used. FEED FORWARD : Square timing recovery VITERBI-VITERBI : Queries the time location of the eye diagram with the largest standard deviation. IMPROVED VITERBI-VITERBI : Queries the crossing points of the eye

Parameter	Description
	diagram and calculates the center as the distance between these crossing points.
Number of symbols	Defines the length of the training sequence by the number of symbols.

6.2.7.9 Node: MATLAB

This node enables you to add a MATLAB script to the pipeline.

The node is **only** enabled if you have a MATLAB license and an installed MATLAB version 2014a or newer. In the MATLAB node, define the location of the MATLAB file. Example scripts are included in the VISIQ software package, locate them in ...\\Quantifi Photonics\\VisIQ\\Examples\\Scripts.

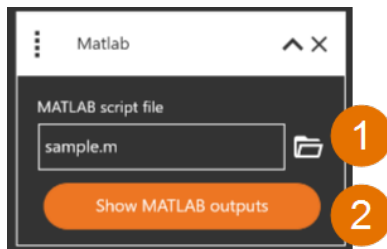
Commands

Click **>>** in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Node: MATLAB](#).

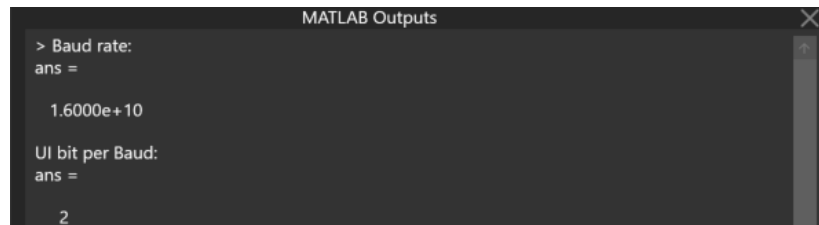
Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	Description
:PROJect	
:PIPEline	
:NODEs	
:MATLab<n>	
:SETTings	Set MATLAB script file path >>

UI reference



- To configure the MATLAB node:
1. Set the MATLAB script file path.
 2. Display command output.



6.2.7.10 Node: Nonlinear correction

This node can correct gain imbalances, IQ offsets as well as non-linear distortion that may present on the signal.

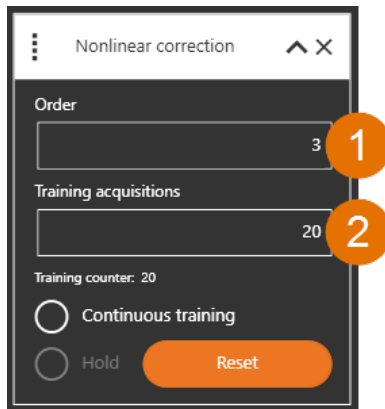
Commands

Click [>>](#) in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Node: Nonlinear correction](#).

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	Description
:PROJect	
:PIPEline	
:NODEs	
:NONlinearCorrection<n>	
:ORDeR?	Query order >>
:ORDeR	Set order >>
:TRAiNing	
:ACQuisitions?	Query the number of training acquisitions >>
:ACQuisitions	Set the number of training acquisitions >>
:CONTInuous?	Query the status of continuous training acquisitions >>
:CONTInuous	Turn continuous training acquisitions ON or OFF >>
:HOLD?	Query training acquisitions hold >>
:HOLD	Hold training acquisitions >>
:COUNT	
:ReSeT	Reset training acquisitions count >>

UI reference



► To set the nonlinear correction properties:

1. Enter **Order**.
2. Enter **Training acquisitions**. The default value is a good starting point, adjust it as required.

Training optimizes the algorithm of the node by iteratively adjusting parameters. The **Training counter** counts down by one with each acquisition, from the set value to 0. When the counter reaches 0, training will hold, the node will use the latest values.

Tick **Continuous training** for training to continue indefinitely.

Click **Hold** to pause training.

Click **Reset** to reset the training counter, and reverse values as they were before capturing training acquisitions.

Parameter	Description
Order	The order of the nonlinear polynomial. Higher-order polynomials can exhibit more complex behaviours or signal correction.

6.2.7.11 Node: Chromatic dispersion

This node compensates for chromatic dispersion due to fiber propagation.

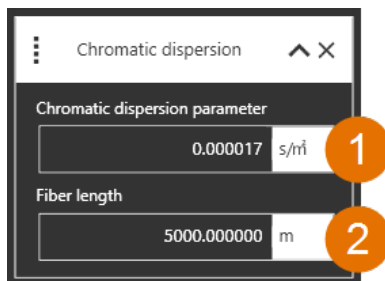
Commands

Click **>>** in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Node: Chromatic dispersion](#).

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	Description
:PROJect	
:PIPEline	
:NODEs	
:CHROmaticdispersion<n>	
:PARaMeter?	Query parameter >>
:PARaMeter	Set parameter >>
:FIBerLength?	Query fiber length >>
:FIBerLength	Set fiber length >>

UI reference



- ▶ To set the node properties:
1. Enter **Chromatic dispersion parameter**.
 2. Enter **Fiber length**.

Parameter	Description
Chromatic dispersion parameter	The chromatic dispersion parameter value of the fiber in use, expressed in the unit of Seconds per meter squared.
Fiber length	Length of the dispersion inducing fiber in the link to compensate for.

6.2.7.12 Node: IQ offset removal

This node corrects IQ offsets that may be present on the signal.

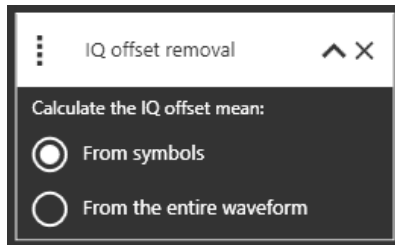
Commands

Click [>>](#) in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Node: IQ offset removal](#).

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	Description
:PROJect	
:PIPELine	
:NODEs	
:IQOFFsetremoval<n>	
:CALCulatefrom?	Query calculation algorithm >>
:CALCulatefrom	Set calculation algorithm >>

UI reference



▶ To set the node properties:

1. Select how to calculate the IQ offset mean: **From symbols** or **From the entire waveform**.

6.2.8 Run the analysis

When running the analysis, VISIQ acquires data as per your selected settings and provides immediate visual feedback. You can customize the visualization area and measurement table by selecting plots and measurements to display.

6.2.8.1 Commands

Click **>>** in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Run the analysis](#).

Command	Description
:PROJect	
:ANALysis	
:SINGle	Run a single data acquisition >>
:CONTinuous	Run continuous data acquisition >>
:STOP	Stop continuous data acquisition >>

6.2.8.2 UI reference



▶ To run the analysis:

1. Click **Single** to capture a single acquisition, or **Continuous** to repeatedly catch a single acquisition.
2. Click **Stop** to stop continuous data acquisition.

6.2.9 Run tests

The Bit Error Rate measurement in **Run tests** compares the received data pattern to the pre-defined reference pattern configured by the user to make an accumulative error count. VISIQ's BER test supports various multiplexing and encoding options and common PRBS patterns ranging from PRBS3 to PRBS31 are available from the selection menu.

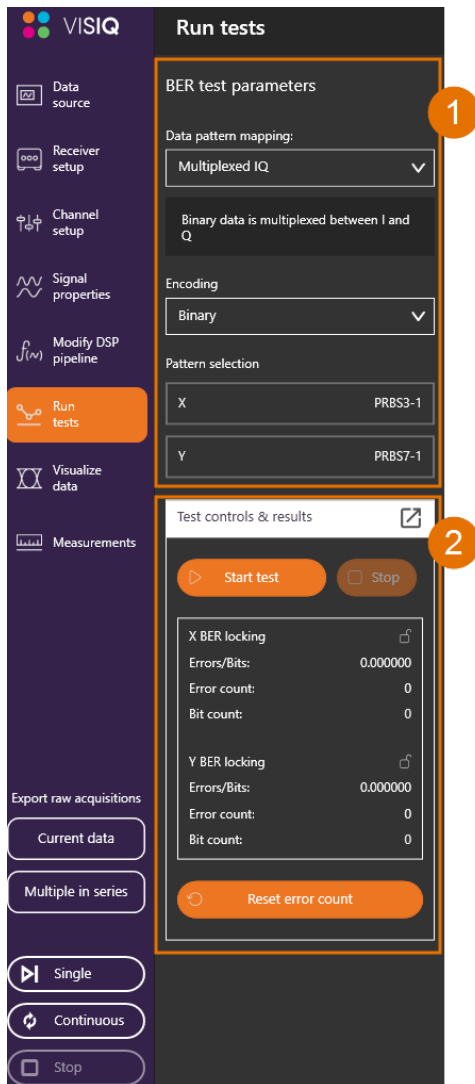
Changing the pattern in **Run tests**, will also change the pattern in **Signal properties**.

6.2.9.1 Commands

Click [>>](#) in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Run tests](#).

Command	Description
:PROJect	
:RUNTests	
:PATtern	
:MAPPing?	Query data pattern mapping selection >>
:MAPPing	Set data pattern mapping >>
:ENCoding?	Query data encoding selection >>
:ENCoding	Set data encoding >>
:INFO?	Query supported PRBS patterns >>
:SELeCt?	Query test pattern selection (4/2/1 outputs) >>
:SELeCt	Set test pattern selection (4/2/1 sections to select) >>
:BERtest	
:STATe?	Query BER test state - Start / Stop >>
:STATe	Set BER test state - Start / Stop >>
:ReSeT	Reset BER error count >>
:ERRor	
:BITS?	Query errors/bits (4/2/1 outputs) >>
:COUNT?	Query error count (4/2/1 outputs) >>
:BITS	
:COUNT?	Query bits count (4/2/1 outputs) >>
:CONTrol	Set BER control pop-up state - Show / Hide >>
:LOCKing?	Query the status of the BER lock indicator >>

6.2.9.2 UI reference



► To select test parameters:

1. Select test parameters.
2. Run the tests.

Parameter	Description
Data pattern mapping	Select data pattern mapping option: Independent: The four tributaries are individually modulated with PRBS or data patterns. Multiplexed IQ: The I and Q tributaries of the same polarization are multiplexed and modulated with a single PRBS or data pattern per polarization. Multiplexed XY: All four tributaries are multiplexed and modulated with a single PRBS or data pattern.
Encoding	Select decoder settings. Binary: The symbol's I and Q bits are encoded in an incremental order on each axes. Gray: The binary values are arranged in such a way that only one bit changes between adjacent symbols.
Pattern selection	Select the test pattern.
Errors/Bits	Bit error rate: Total number of errors over total bits tested
Error count	Total number of errors detected
Bit count	Total number of bits tested

6.2.10 Select measurements

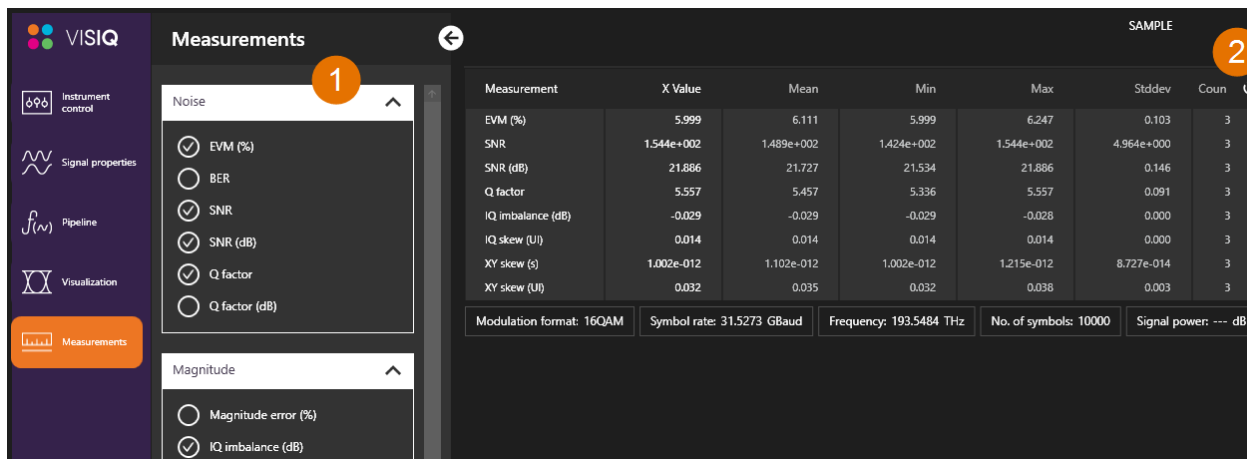
VISIQ can perform a comprehensive range of measurements and display selected values in the measurement table in the UI. The table values will update with each data acquisition.

6.2.10.1 Commands

Click **>>** in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Select measurements](#).

Command	Description
:PROJect	
:MEASurements	
:INFO?	Query what measurement types are displayed in the measurement table >>
:SHOW?	Query what measurement types are displayed in the measurement table and mapping of supported measurement types >>
:SHOW	Set measurements to be displayed in the measurement table >>
:HIDE?	Query supported measurement types >>
:HIDE	Set measurements not to be displayed in the measurement table >>
:MASK	Set bit mask to define measurements to be displayed in the measurement table (0 = OFF, 1 = ON) >>
:ReSeT	Reset statistical data for all measurements >>
:EXPOrt	Export the measurements results as a .csv file in a predefined location >>

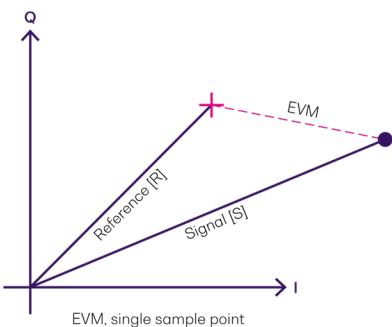
6.2.10.2 UI reference

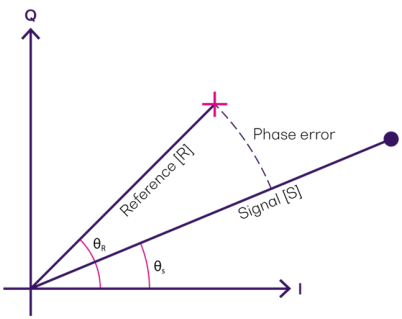
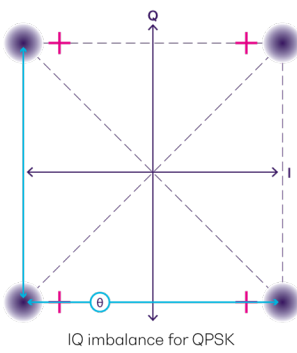


► To select measurements:

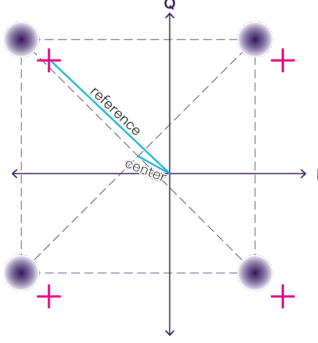
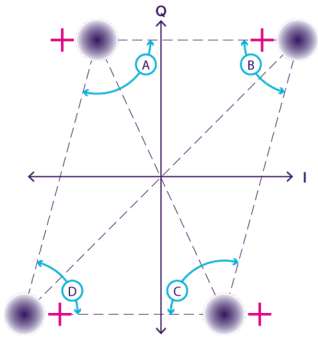
1. Select the measurements to display in the table.
2. Reset measurement values.

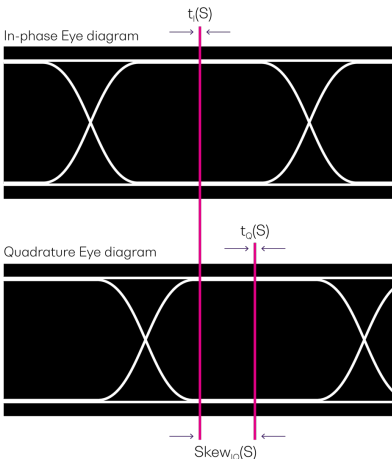
6.2.10.3 Measurement details

Type	Measurement	Details	
Noise	EVM (%)	The magnitude of the error vector – the difference between the signal vector and the ideal reference vector. rms - root mean square $EVM (\%) = 100 \times \frac{rms signal - reference }{\max reference }$	
	BER estimate	An estimation of the bit error rate calculated from the Q factor. Time efficient alternative comparing to an actual bit counting. $BER_{est} = \frac{1 - \frac{1}{L}}{\log_2 L} \operatorname{erfc} \left(\frac{Q \text{ factor}}{\sqrt{2}} \right)$ L = The maximum number of levels in the signal. erfc = The complimentary error function – applies the assumption of the Gaussian distribution of constellation points.	
	SNR	Signal-to-Noise Ratio expressed as a ratio. $SNR = \frac{Signal \ Power}{Noise \ Power}$	
	SNR (dB)	Signal to noise ratio expressed as in decibels.	
	Q factor	An alternate representation of the Signal-to-Noise Ratio - Q factor - in percentage. $Q \ factor_{(\%)} = \sqrt{\frac{3 \log_2 L}{L^2 - 1} \frac{2}{(EVM\%/100)^2 \log_2 2^{bits_per_symbol}}}$	

Type	Measurement	Details	
		L = The maximum number of levels in the signal.	
	Q factor (dB)	An alternate representation of the Signal-to-Noise Ratio - Q factor - in dB. $Q\ factor_{(dB)} = 20\log_{10}(Q\ factor)$	
Magnitude	Magnitude error (%)	The difference in amplitude between the reference symbol and the signal vector. $\frac{Magnitude\ Error_{RMS}}{Magnitude\ Error = S - R } = \frac{1}{\sqrt{N}}(Magnitude\ Error_1^2 + \dots + Magnitude\ Error_N^2)$	
	IQ imbalance (dB)	The ratio of the in-phase component vs. the quadrature component of the constellation points in dB.	
	IQ imbalance (%)	The ratio of the in-phase component vs. the quadrature component of the constellation points in percentage.	
	IQ magnitude offset (dB)	The center of the constellation with reference to the ideal center point in decibels.	

Type	Measurement	Details	
	IQ magnitude offset (%)	The center of the constellation with reference to the ideal center point in percentage.	
	I offset (%)	The center of the In- phase component of the constellation with reference to the ideal center point in percentage.	
	Q offset (%)		
Phase	Phase error (rad)	The phase difference between the signal vector and the ideal reference vector. $Phase\ Error_{RMS} = \frac{Phase\ Error = S - R }{\sqrt{\frac{1}{N}(Phase\ Error_1^2 + \dots + Phase\ Error_N^2)}}$	

Type	Measurement	Details	
	IQ phase offset (rad)	The phase of the vector from the center of the constellation with reference to the ideal center point in radians.	
	IQ phase error (deg)	The measurement of the angle between the in-phase and quadrature components of the signal. Every deviation from 90° between the in-phase and the quadrature is an IQ Phase Imbalance. Quad Error is the average measurement within the constellation points. $IQ\ Phase\ Imbalance = \frac{\angle A - \angle B + \angle C - \angle D}{4}$	

Type	Measurement	Details	
Time	IQ skew (s)	The time difference between the in-phase and quadrature eye centers in seconds. $Skew_{IQ}(s) = t_I(s) - t_Q(s)$	
	IQ skew (UI)	Time difference between the in-phase and quadrature eye centers in UI.	
	XY skew (s)	The temporal skew between the X and Y polarization symbols.	
	XY skew (UI)	The temporal skew between the X and Y polarization symbols, measured in the percentage of the unit interval of the symbol.	
	PMD (s)	The polarization mode dispersion, shared between X Pol and Y Pol results.	
Frequency	Frequency offset (Hz)	The frequency offset of the carrier.	
	Line width (Hz)	The linewidth estimation of the carrier and local oscillator.	
	Baudrate (Hz)	Baud rate of the received signal.	

6.2.11 Load and save data

VISIQ offers several options to load and save data.

6.2.11.1 Commands

Click [>>](#) in the summary table below to navigate to the detailed command description, or refer to command descriptions in [Load and save data](#).

Command	Description
:PROJect	
:NEW	Create a new project file >>
:OPEN	Open a project file from a specified file path >>
:SAVE	Save the current project in the previously saved file or save as a new file >>
:EXPOrt	
:SINGle	Export currently cached raw data >>
:STARt	Start export of raw data as a series of upcoming acquisitions >>
:STOP	Stop export of raw data >>
:PIPEline	
:CONFIguration	
:EXPOrt	Export the configuration of the DSP pipeline >>
:IMPOrt	Import the configuration of the DSP pipeline >>

6.3 SCPI commands descriptions

6.3.1 Common commands

Command	*IDN?			Summary >>
Syntax	*IDN?			
Description	Query the software identification			
Parameters	N/A			
Response	A comma-separated array of values			
Example	*IDN? -> Quantifi Photonics Ltd,VISIQSCPIServer,QPNZLAP045,HW0.00.00SW1.0.0			

Command	*OPT? APPS			Summary >>
Syntax	*OPT? APPS			
Description	Query the application			
Parameters	N/A			
Response				
Example	*OPT? APPS -> VISIQ			

Command	ESR?			Summary >>
Syntax	*ESR?			
Description	Query the Standard Event Status Register			
Parameters	N/A			
Response	Unsigned integer 8 bit value for the register <0 to 255>, as a string.			
	Bit	Description	Decimal Value	
	7 (MSB)	Not used	0	
	6	Not used	0	
	5	Command error	32	
	4	Command Execution Error	16	
	3	Device Dependent Error	8	
	2	Not used	0	
	1	Not used	0	
	0 (LSB)	Operation Complete	1	
Example	*ESR? -> 8			
	*ESR? -> 32			

NOTE

It is recommended to use the *ESR? command query after every command that is sent to the device. The *ESR? query will be able to catch:

- **Device dependent Error** – the device is reporting an error in operation.
- **Execution Error** – SCPI was unable to execute the given command.
- **Command Error** – SCPI was unable to parse the given command, likely due to an incorrect command.

Command	*ESE?	Summary >>
Syntax	*ESE?	
Description	Query the Standard Event Status Enable Register (Mask)	
Parameters	N/A	
Response	Unsigned integer 8 bit value for the register <0 to 255>, as a string.	
Example	*ESE? -> 254	

Command	*ESE	Summary >>
Syntax	*ESE<wsp><value>	
Description	Set the Standard Event Status Enable Register	
Parameters	N/A	
Response	N/A	
Example	*ESE 254	

6.3.2 System menu functions

Command	:APP:STATe?	Summary >>
Syntax	:APP:STATe? [<wsp>INFO]	
Description	Query the state of the VISIQ application	
Parameters	INFO:	
Response	0: Unknown	
	1: Unavailable	
	2: Available	
	3: Busy	
	4: Stopped	
Example	:APP:STAT? -> UNKNOWN	

Command	:APP:CONNect	Summary >>
Syntax	:APP:CONNect<wsp><VISIQ>	
Description	Connect to the VISIQ application	
Parameters	VISIQ: Connects to the VISIQ application	
Response	N/A	
Example	:APP:CONN VISIQ	

Command	:APP:DISConnect	Summary >>
Syntax	:APP:DISConnect<wsp><VISIQ>	
Description	Disconnect from the VISIQ application	
Parameters	VISIQ: Disconnects from the VISIQ application	
Response	N/A	
Example	:APP:DISC VISIQ	

Command	:APP:EXIT	Summary >>
Syntax	:APP:EXIT<wsp><NO_SAVE <file_path>>	
Description	Exit VISIQ with or without saving the project	
Parameters	NO_SAVE: Exit VISIQ without saving the current project	
	file_path: Save the current project file in this location	
Response	N/A	
Example		

6.3.3 Select the data source

Command	:PROJect:SOURce:DATA:TYPE?	Summary >>
Syntax	:PROJect:SOURce:DATA:TYPE?	
Description	Query the selected data source	
Parameters	N/A	
Response	waveform file or oscilloscope	
Example	<pre>:PROJect:SOURce:DATA:TYPE? -> WAVEFORM "C:\Program Files\QuantifiPhotonics\VisIQ\Examples\Datasets\test.csv" :PROJect:SOURce:DATA:TYPE? -> SCOPE 10.10.10.150</pre>	

Command	:PROJect:SOURce:DATA:TYPE	Summary >>
Syntax	:PROJect:SOURce:DATA:TYPE<wsp><"file_path" "ip_address">	
Description	Select the data source: An oscilloscope of your test setup or a previously acquired waveform file	
Parameters	"file_path": Sets TYPE to WAVEFORM and defines the file path to the waveform file "ip_address": Sets TYPE to SCOPE and identifies the oscilloscope by its IP address	
Response	N/A	
Example	<pre>:PROJ:SOUR:DATA:TYPE "C:\ProgramFiles\QuantifiPhotonics\VisIQ\Examples\Datasets\test.csv" :PROJ:SOUR:DATA:TYPE "10.10.10.150"</pre>	

Command	:PROJect:SOURce:DATA:TYPE:SCOPE?	Summary >>
Syntax	:PROJect:SOURce:DATA:TYPE:SCOPE?	
Description	Query available scopes on the network	
Parameters	N/A	
Response	List of available oscilloscopes, including IP addresses and oscilloscope details	
Example	:PROJect:SOURce:DATA:TYPE:SCOPE?	

Command	:PROJect:SOURce:DATA:TYPE:SCOpe:STATe?	Summary >>
Syntax	:PROJect:SOURce:DATA:TYPE:SCOpe:STATe?	
Description	Query oscilloscope connection state	
Parameters	N/A	
Response		
Example	:PROJ:SOUR:DATA:TYPE:SCOP:STAT? -> ON	

Command	:PROJect:SOURce:DATA:TYPE:SCOpe:CONNEct	Summary >>
Syntax	:PROJect:SOURce:DATA:TYPE:SCOpe:CONNEct	
Description	Connect with the selected oscilloscope	
Parameters	N/A	
Response	N/A	
Example	:PROJ:SOUR:DATA:TYPE:SCOP:CONN	

Command	:PROJect:SOURce:DATA:TYPE:SCOpe:DISConnect	Summary >>
Syntax	:PROJect:SOURce:DATA:TYPE:SCOpe:DISConnect	
Description	Disconnect from an oscilloscope	
Parameters	N/A	
Response	N/A	
Example	:PROJ:SOUR:DATA:TYPE:SCOP:DISC	

Command	:PROJect:SOURce:DATA:TYPE:SCOpe:INITialize	Summary >>
Syntax	:PROJect:SOURce:DATA:TYPE:SCOpe:INITialize	
Description	Initialize the connected oscilloscope	
Parameters	N/A	
Response	N/A	
Example	:PROJ:SOUR:DATA:TYPE:SCOP:INIT	

Command	:PROJect:SOURce:DATA:IDN?	Summary >>
Syntax	:PROJect:SOURce:DATA:IDN?	
Description	Query the connected oscilloscope ID	
Parameters	N/A	
Response	A comma separated string containing "<IP address>,<manufacturer>,<part number>,<serial number><version number>".	
Example	10.10.10.150,TEKTRONIX,ABC123,XYZ789,CF:01.0CT FV:01.01.0 Build 01	

Command	:PROJect:SOURce:DATA:RECORD_LENGTH?	Summary >>
Syntax	:PROJect:SOURce:DATA:RECORD_LENGTH?<wsp>[MIN MAX DEF ACT ALL UNIT]	
Description	Query the oscilloscope record length	
Parameters	MIN : Returns the minimum value in the default unit	
	MAX : Returns the maximum value in the default unit	
	DEF : Returns the default value in the default unit	
	ACT : Returns the currently set value in the default unit	
	ALL : Returns all the above parameters in a comma-separated string: <MIN>, <MAX>, <DEF>, <ACT>	
	UNIT : Returns the unit	
Response	A single value, or a comma-separated array of values	
Example		

Command	:PROJect:SOURce:DATA:SAMPLing_RATE?	Summary >>
Syntax	:PROJect:SOURce:DATA:SAMPLing_RATE?<wsp>[MIN MAX DEF ACT SET ALL UNIT]	
Description	Query the oscilloscope sampling rate	
Parameters	MIN : Returns the minimum value in the default unit	
	MAX : Returns the maximum value in the default unit	
	DEF : Returns the default value in the default unit	
	ACT : Returns the currently set value in the default unit	
	SET : Returns the currently set value in the default unit	
	ALL : Returns all the above parameters in a comma-separated string: <MIN>, <MAX>, <DEF>, <ACT>, <SET>	
	UNIT : Returns the unit	
Response	A single value, or a comma-separated array of values	
Example		

Command	:PROJect:SOURce:DATA:SAMPLing_RATE	Summary >>
Syntax	:PROJect:SOURce:DATA:SAMPLing_RATE<wsp><value>[MIN MAX DEF]	
Description	Set the oscilloscope sampling rate	
Parameters	<value> : Sets this value	
	MIN : Sets the minimum value	
	MAX : Sets the maximum value	
	DEF : Sets the default value	
Response	N/A	
Example		

Command	:PROJect:SOURce:DATA:BANDwidth?	Summary >>
Syntax	:PROJect:SOURce:DATA:BANDwidth?<wsp>[MIN MAX DEF ACT SET ALL UNIT]	
Description	Query the oscilloscope bandwidth	
Parameters	MIN : Returns the minimum value in the default unit	
	MAX : Returns the maximum value in the default unit	
	DEF : Returns the default value in the default unit	
	ACT : Returns the currently set value in the default unit	
	SET : Returns the currently set value in the default unit	
	ALL : Returns all the above parameters in a comma-separated string: <MIN>,<MAX>,<DEF>,<ACT>,<SET>	
	UNIT : Returns the unit	
Response	A single value, or a comma-separated array of values	
Example		

Command	:PROJect:SOURce:DATA:BANDwidth	Summary >>
Syntax	:PROJect:SOURce:DATA:BANDwidth<wsp><value MIN MAX DEF >	
Description	Set the oscilloscope bandwidth	
Parameters	<value> : Sets this value	
	MIN : Sets the minimum value	
	MAX : Sets the maximum value	
	DEF : Sets the default value	
Response	N/A	
Example		

6.3.4 Connect VISIQ with the coherent receiver

Index <m>: The channel index of the coherent receiver, this is an integer <1 to 4> depending on the model.

Command	:PROJect:SOURce:RECeiver	Summary >>
Syntax	:PROJect:SOURce:RECeiver<wsp><"IP_ADDRes">	
Description	Select a coherent receiver	
Parameters	<IP_ADDRes>: IP address of the coherent receiver	
Response	N/A	
Example	:PROJ:SOUR:REC "10.10.10.206"	

Command	:PROJect:SOURce:RECeiver:FIND?	Summary >>
Syntax	:PROJect:SOURce:RECeiver:FIND?	
Description	Query available coherent receivers on the network	
Parameters	N/A	
Response	List of available receivers, including IP addresses and receiver details	
Example	:PROJ:SOUR:REC:FIND? -> 10.10.10.206,Quantifi Photonics Ltd,IQRX-1004-FA-EPIQ,QP-204101,HW0.00.01SW3.04.04 10.10.10.107,Quantifi Photonics Ltd,IQRX-1002-FA-EPIQ,QP-233103,HW0.00.00SW3.01.18	

Command	:PROJect:SOURce:RECeiver:STATe?	Summary >>
Syntax	:PROJect:SOURce:RECeiver:STATe?	
Description	Query receiver connection state	
Parameters	N/A	
Response		
Example		

Command	:PROJect:SOURce:RECeiver:CONNect	Summary >>
Syntax	:PROJect:SOURce:RECeiver:CONNect	
Description	Connect with a coherent receiver	
Parameters	N/A	
Response	N/A	
Example	:PROJ:SOUR:REC:CONN	

Command	:PROJect:SOURce:RECeiver:DISConnect	Summary >>
Syntax	:PROJect:SOURce:RECeiver:DISConnect	
Description	Disconnect the coherent receiver	
Parameters	N/A	
Response	N/A	
Example	:PROJ:SOUR:REC:DISC	

Command	:PROJect:SOURce:RECeiver:IDN?	Summary >>
Syntax	:PROJect:SOURce:RECeiver:IDN?	
Description	Query the connected coherent receiver ID	
Parameters	N/A	
Response	A comma separated string containing "<IP address>,<manufacturer>,<part number>,<serial number><version number>".	
Example	:PROJ:SOUR:REC:IDN? -> 10.10.10.206,Quantifi Photonics Ltd,IQRX-1004-FA-EPIQ,QP-204101,HW0.00.01SW3.04.04	

Command	:PROJect:SOURce:RECeiver:CHANnel<m>:STATe?	Summary >>
Syntax	:PROJect:SOURce:RECeiver:CHANnel<m>:STATe?	
Description	Query the laser state	
Parameters	N/A	
Response		
Example		

Command	:PROJect:SOURce:RECeiver:CHANnel<m>:STATe	Summary >>
Syntax	:PROJect:SOURce:RECeiver:CHANnel<m>:STATe<wsp><ON OFF 1 0>	
Description	Set laser state to ON or OFF	
Parameters	<ON 1>: Set laser state to ON <OFF 0>: Set laser state to OFF	
Response	N/A	
Example		

Command	:PROJect:SOURce:RECeiver:CHANnel<m>:POWer? Summary >>
Syntax	:PROJect:SOURce:RECeiver:CHANnel<m>:POWer?<wsp><MIN MAX DEF SET UNIT STEP ACT ALL>
Description	Query the laser power
Parameters	MIN : Returns the minimum value in the default unit
	MAX : Returns the maximum value in the default unit
	DEF : Returns the default value in the default unit
	SET : Returns the currently set value in the default unit
	UNIT : Returns the unit
	ACT : Returns the currently set value in the default unit
	ALL : Returns all the above parameters in a comma-separated string:<value>,<MIN>,<MAX>,<DEF>,<SET>,<UNIT>,<STEP>,<ACT>
Response	A single value, or a comma-separated array of values
Example	

Command	:PROJect:SOURce:RECeiver:CHANnel<m>:POWer Summary >>
Syntax	:PROJect:SOURce:RECeiver:CHANnel<m>:POWer<wsp><value MIN MAX DEF]
Description	Set laser power
Parameters	<value> : Sets this value
	MIN : Sets the minimum value
	MAX : Sets the maximum value
	DEF : Sets the default value
Response	N/A
Example	

Command	:PROJect:SOURce:RECeiver:CHANnel<m>:WAVElength?	Summary >>
Syntax	:PROJect:SOURce:RECeiver:CHANnel<m>:WAVElength?<wsp><MIN MAX DEF SET UNIT STEP ACT ALL>	
Description	Query the laser wavelength	
Parameters	MIN : Returns the minimum value in the default unit	
	MAX : Returns the maximum value in the default unit	
	DEF : Returns the default value in the default unit	
	SET : Returns the currently set value in the default unit	
	UNIT : Returns the unit	
	ACT : Returns the currently set value in the default unit	
	ALL : Returns all the above parameters in a comma-separated string:<value>,<MIN>,<MAX>,<DEF>,<SET>,<UNIT>,<STEP>,<ACT>	
Response	A single value, or a comma-separated array of values	
Example		

Command	:PROJect:SOURce:RECeiver:CHANnel<m>:WAVElength	Summary >>
Syntax	:PROJect:SOURce:RECeiver:CHANnel<m>:WAVElength<wsp><value MIN MAX DEF>	
Description	Set laser wavelength	
Parameters	<value> : Sets this value	
	MIN : Sets the minimum value	
	MAX : Sets the maximum value	
	DEF : Sets the default value	
Response	N/A	
Example		

6.3.5 Set channels

Index <p>: The channel index of the oscilloscope, this is an integer <1 to 4> depending on the model.

Command	:PROJect:SOURce:DATA:CHANnel [<p>]:MAPPING?	Summary >>
Syntax	:PROJect:SOURce:DATA:CHANnel [<p>]:MAPPING?	
Description	Query channel mapping of the oscilloscope	
Parameters	N/A	
Response	The mapping between numerals and text format	
Example	:PROJ:SOUR:DATA:CHAN1:MAPP? -> XI:1 XQ:2 YI:3 YQ:4	

Command	:PROJect:SOURce:DATA:CHANnel<p>:MAPPING	Summary >>
Syntax	:PROJect:SOURce:DATA:CHANnel<p>:MAPPING<wsp><XI XQ YI YQ>	
Description	Set channel mapping of the oscilloscope	
Parameters	XI : Maps the oscilloscope channel <n> with the coherent receiver XI tributary XQ : Maps the oscilloscope channel <n> with the coherent receiver XQ tributary YI : Maps the oscilloscope channel <n> with the coherent receiver YI tributary YQ : Maps the oscilloscope channel <n> with the coherent receiver YQ tributary	
Response	N/A	
Example	:PROJ:SOUR:DATA:CHAN1:MAPP XI	

Command	:PROJect:SOURce:DATA:CHANnel [<p>]:SCALe?	Summary >>
Syntax	:PROJect:SOURce:DATA:CHANnel [<p>]:SCALe?<wsp><MIN MAX DEF SET UNIT ALL>	
Description	Query the vertical scaling of the oscilloscope channels	
Parameters	MIN: Returns the minimum value in the default unit	
	MAX: Returns the maximum value in the default unit	
	DEF: Returns the default value in the default unit	
	SET: Returns the currently set value in the default unit	
	UNIT: Returns the measurement unit	
	ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:SOUR:DATA:CHAN1:SCAL? -> 0.006250	

Command	:PROJect:SOURce:DATA:CHANnel [<p>]:SCALe	Summary >>
Syntax	:PROJect:SOURce:DATA:CHANnel [<p>]:SCALe<wsp><value AUTO>	
Description	Set vertical scaling of oscilloscope channels	
Parameters	<value>: Sets this value	
	AUTO: Sets an auto-calculated value	
Response	N/A	
Example	:PROJ:SOUR:DATA:CHAN:SCAL AUTO	
	:PROJ:SOUR:DATA:CHAN2:SCAL 0.006250	

6.3.6 Configure signal properties

Command	:PROJect:SOURce:DATA:SIGNal:POLarization?	Summary >>
Syntax	:PROJect:SOURce:DATA:SIGNal:POLarization?<wsp><INFO>	
Description	Query the selected polarization type of the signal	
Parameters	INFO : Returns supported polarization types	
Response	1 : SINGLE	
	2 : DUAL	
Example		

Command	:PROJect:SOURce:DATA:SIGNal:POLarization	Summary >>
Syntax	:PROJect:SOURce:DATA:SIGNal:POLarization<wsp><SINGLE DUAL>	
Description	Set the polarization type of the signal	
Parameters	SINGLE : Sets polarization type to single	
	DUAL : Sets polarization type to dual	
Response	N/A	
Example		

Command	:PROJect:SOURce:DATA:SIGNal:POLarization:AXIS?	Summary >>
Syntax	:PROJect:SOURce:DATA:SIGNal:POLarization:AXIS?<wsp><INFO>	
Description	Query the polarization axis for single polarization	
Parameters	INFO : Returns supported polarization axis	
Response	0 : X	
	1 : Y	
Example		

Command	:PROJect:SOURce:DATA:SIGNal:POLarization:AXIS	Summary >>
Syntax	:PROJect:SOURce:DATA:SIGNal:POLarization:AXIS<wsp><X Y>	
Description	Set the polarization axis for single polarization	
Parameters	x : Sets the axis to X	
	y : Sets the axis to Y	
Response	N/A	
Example		

Command	:PROJect:SOURce:DATA:SIGNal:MODUlation?	Summary >>
Syntax	:PROJect:SOURce:DATA:SIGNal:MODUlation?	
Description	Query the modulation format	
Parameters	N/A	
Response	0: BPSK	
	1: QPSK	
	2: 8PSK	
	3: 16PSK	
	4: 32PSK	
	5: 8QAM	
	6: 16QAM	
	7: 32QAM	
	8: 64QAM	
	9: 128QAM	
	10: 256QAM	
	11: 8STAR-QAM	
	12: 16STAR-QAM	
	13: 32STAR-QAM	
	14: 64STAR-QAM	
	15: 128STAR-QAM	
	16: 256STAR-QAM	
Example	:PROJ:SOUR:DATA:SIGN:MODU? -> 16QAM	

Command	:PROJect:SOURce:DATA:SIGNal:MODUlation	Summary >>
Syntax	:PROJect:SOURce:DATA:SIGNal:MODUlation<wsp><0 1 2 3....16>	
Description	Set the modulation format	
Parameters	0: MF_BPSK	
	1: MF_QPSK	
	2: MF_8PSK	
	3: MF_16PSK	
	4: MF_32PSK	
	5: MF_8QAM	
	6: MF_16QAM	
	7: MF_32QAM	
	8: MF_64QAM	
	9: MF_128QAM	
	10: MF_256QAM	
	11: MF_8STAR_QAM	
	12: MF_16STAR_QAM	
	13: MF_32STAR_QAM	
	14: MF_64STAR_QAM	
	15: MF_128STAR_QAM	
	16: MF_256STAR_QAM	
Response	N/A	
Example	:PROJ:SOUR:DATA:SIGN:MODU MF_8PSK	

Command	:PROJect:SOURce:DATA:SIGNal:MODUlation:INFO?	Summary >>
Syntax	:PROJect:SOURce:DATA:SIGNal:MODUlation:INFO?	
Description	Query supported modulation formats	
Parameters	N/A	
Response	The mapping between numerals and text format	
Example	<pre> :PROJ:SOUR:DATA:SIGN:MODU:INFO? -> 0:BPSK 1:QPSK 2:8PSK 3:16PSK 4:32PSK 5:8QAM 6:16QAM 7:32QAM 8:64QAM 9:128QAM 10:256QAM 11:8STAR-QAM 12:16STAR-QAM 13:32STAR-QAM 14:64STAR-QAM 15:128STAR-QAM 16:256STAR-QAM </pre>	

Command	:PROJect:SOURce:DATA:SIGNal:BAUDrate?	Summary >>
Syntax	:PROJect:SOURce:DATA:SIGNal:BAUDrate? [<wsp><MIN MAX DEF SET UNIT STEP ACT ALL>]	
Description	Query the Baud rate of the signal	
Parameters	None: Returns the currently set value in the default unit	
	MIN : Returns the minimum value in the default unit	
	MAX : Returns the maximum value in the default unit	
	DEF : Returns the default value in the default unit	
	SET : Returns the currently set value in the default unit	
	UNIT : Returns the measurement unit	
	ACT : Returns the currently set value in the default unit	
	ALL : Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:SOUR:DATA:SIGN:BAUD? -> 59843750000.000000	

Command	:PROJect:SOURce:DATA:SIGNal:BAUDrate	Summary >>
Syntax	:PROJect:SOURce:DATA:SIGNal:BAUDrate<wsp><valueMIN MAX DEF >	
Description	Set the Baud rate of the signal	
Parameters	<value> : Sets this value	
	MIN : Sets the minimum value	
	MAX : Sets the maximum value	
	DEF : Sets the default value	
Response	N/A	
Example	:PROJ:SOUR:DATA:SIGN:BAUD 59843750000	
	:PROJ:SOUR:DATA:SIGN:BAUD 59843750 KBAUD	
	:PROJ:SOUR:DATA:SIGN:BAUD 59843.75 MBAUD	
	:PROJ:SOUR:DATA:SIGN:BAUD 59.84375 GBAUD	

Command	:PROJect:SOURce:DATA:SIGNal:FREQuency?	Summary >>
Syntax	:PROJect:SOURce:DATA:SIGNal:FREQuency? [<wsp><SET UNIT ACT ALL>]	
Description	Query the carrier frequency	
Parameters	None: Returns the currently set value in the default unit SET : Returns the currently set value in the default unit UNIT : Returns the unit ACT : Returns the currently set value in the default unit ALL : Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:SOUR:DATA:SIGN:FREQ? -> 193414500000000.000000	

Command	:PROJect:SOURce:DATA:SIGNal:FREQuency	Summary >>
Syntax	:PROJect:SOURce:DATA:SIGNal:FREQuency<wsp><value MIN MAX DEF >	
Description	Set the carrier frequency	
Parameters	<value> : Sets this value MIN : Sets the minimum value MAX : Sets the maximum value DEF : Sets the default value	
Response	N/A	
Example	:PROJ:SOUR:DATA:SIGN:FREQ 193.4145 THz :PROJ:SOUR:DATA:SIGN:FREQ 191814500000000	

Command	:PROJect:SOURce:DATA:SIGNal:SYMbolsNumber?	Summary >>
Syntax	:PROJect:SOURce:DATA:SIGNal:SYMbolsNumber? [<wsp><SET UNIT ACT ALL>]	
Description	Query the number of symbols	
Parameters	None: Returns the currently set value in the default unit SET : Returns the currently set value in the default unit UNIT : Returns the unit ACT : RReturns the currently set value in the default unit ALL : Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:SOUR:DATA:SIGN:SYM? -> 5000	

Command	:PROJect:SOURce:DATA:SIGNal:SYMBOLsNumber	Summary >>
Syntax	:PROJect:SOURce:DATA:SIGNal:SYMBOLsNumber<wsp><valueMIN MAX DEF >	
Description	Set the number of symbols	
Parameters	<value> : Sets this value	
	MIN : Sets the minimum value	
	MAX : Sets the maximum value	
	DEF : Sets the default value	
Response	N/A	
Example	:PROJ:SOUR:DATA:SIGN:SYMN 5000	

Command	:PROJect:SOURce:DATA:SIGNal:POWer?	Summary >>
Syntax	:PROJect:SOURce:DATA:SIGNal:POWer? [<wsp><UNIT ACT>]	
Description	Query the number of symbols	
Parameters	None: Returns the currently set value in the default unit	
	UNIT : Returns the unit	
	ACT : Returns the currently set value in the default unit	
Response	A single value, or a comma-separated array of values	
Example		

6.3.7 Configure the DSP pipeline

6.3.7.1 Pipeline configuration

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	:PROJect:PIPEline:CONFIguration:ADD	Summary >>
Syntax	:PROJect:PIPEline:CONFIguration:ADD<wsp><1 2 3....15>,<type>	
Description	Add a node to the DSP pipeline in a specific location	
Parameters	1 2 3....15: adds the node to this position in the DSP pipeline <type>: specifies the type of node to add	
Response	N/A	
Example	:PROJ:PIPE:CONF:ADD 3,NONLINEARCORRECTION	

Command	:PROJect:PIPEline:CONFIguration:MOVE	Summary >>
Syntax	:PROJect:PIPEline:CONFIguration:MOVE<wsp><1 2 3....15>,<1 2 3....15>	
Description	Move a node to a different location in the DSP pipeline	
Parameters	1 2 3....15 (first parameter): current position in the DSP pipeline 1 2 3....15 (second parameter): new position in the DSP pipeline	
Response	N/A	
Example	:PROJ:PIPE:CONF:MOVE 3,5	

Command	:PROJect:PIPEline:CONFIguration:REMOve	Summary >>
Syntax	:PROJect:PIPEline:CONFIguration:REMOve<1 2 3....15>	
Description	Remove a node from the DSP pipeline	
Parameters	1 2 3....15: position of the node in the DSP pipeline	
Response	N/A	
Example	:PROJ:PIPE:CONF:REMO 7	

Command	:PROJect:PIPEline:TRAIning:ReSeT	Summary >>
Syntax	:PROJect:PIPEline:TRAIning:ReSeT	
Description	Return training properties of all nodes in the DSP pipeline that support training to default values	
Parameters	0 1 2 3...15: position of the node in the DSP pipeline	
Response	N/A	
Example	:PROJ:PIPE:TRAN:RST	

Command	:PROJect:PIPEline:NODEs [<n>] ?	Summary >>
Syntax	:PROJect:PIPEline:NODEs [<n>] ?	
Description	Query the configuration of the DSP pipeline	
Parameters	N/A	
Response	Nodes and position of nodes in the pipeline	
Example	<pre> :PROJ:PIPE:NODE0? -> 0:CoRx :PROJ:PIPE:NODE1? -> 1:CoRx auto calibrate :PROJ:PIPE:NODE? -> 0:CoRx 1:CoRx auto calibrate 2:Retimer 3:Nonlinear correction 4:PolDeMux 5:Equalizer 6:Carrier offset removal 7:Phase estimator </pre>	

6.3.7.2 Node: CoRx

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	:PROJect:PIPEline:NODEs:CORX<n>:RECeiver?	Summary >>
Syntax	:PROJect:PIPEline:NODEs:CORX<n>:RECeiver?	
Description	Query the receiver type	
Parameters	N/A	
Response	IQRX: Quantifi Photonics' Coherent Optical Receiver IQRX OTHeR: A non-Quantifi Photonics coherent receiver.	
Example	:PROJ:PIPE:NODE:CORX0:REC? -> IQRX	

Command	:PROJect:PIPEline:NODEs:CORX<n>:RECeiver	Summary >>
Syntax	:PROJect:PIPEline:NODEs:CORX<n>:RECeiver<wsp><IQRX OTHeR>	
Description	Set the receiver type	
Parameters	IQRX: Quantifi Photonics' Coherent Optical Receiver IQRX OTHeR: A non-Quantifi Photonics coherent receiver.	
Response	N/A	
Example	:PROJ:PIPE:NODE:CORX0:REC IQRX	

Command	:PROJect:PIPEline:NODEs:CORX<n>:SETTings?	Summary >>
Syntax	:PROJect:PIPEline:NODEs:CORX<n>:SETTings?	
Description	Query the location of the .crs settings file to be used with the receiver	
Parameters	N/A	
Response	<file_path>: file path to the .crs settings file	
Example	:PROJ:PIPE:NODE:CORX0:SETT? -> C:\Program Files\Quantifi Photonics\VisIQ\Examples\Configurations\ DefaultSettings.crs	

Command	:PROJect:PIPEline:NODEs:CORX<n>:SETTings	Summary >>
Syntax	:PROJect:PIPEline:NODEs:CORX<n>:SETTings<wsp><"file_path">	
Description	Define the location of the .crs settings file to be used with the receiver	
Parameters	<file_path>: file path to the .crs settings file	
Response	N/A	
Example	:PROJ:PIPE:NODE:CORX0:SETT "C:\Program Files\Quantifi Photonics\VisIQ\Examples\Configurations\DefaultSettings.crs"	

6.3.7.3 Node: CoRx auto calibrate

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	:PROJect:PIPEline:NODEs:coRXCaLibrate<n>:GRADient?	Summary >>
Syntax	:PROJect:PIPEline:NODEs:coRXCaLibrate<n>:GRADient? [<wsp><MIN MAX DEF SET ALL>]	
Description	Query gradient	
Parameters	None: Returns the currently set value in the default unit	
	MIN: Returns the minimum value in the default unit	
	MAX: Returns the maximum value in the default unit	
	DEF: Returns the default value in the default unit	
	SET: Returns the currently set value in the default unit	
	ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:RXCL1:GRAD? -> 0.300000	

Command	:PROJect:PIPEline:NODEs:coRXCaLibrate<n>:GRADient	Summary >>
Syntax	:PROJect:PIPEline:NODEs:coRXCaLibrate<n>:GRADient<wsp><value MIN MAX DEF>	
Description	Set the gradient	
Parameters	<value>: Sets this value	
	MIN: Sets the minimum value	
	MAX: Sets the maximum value	
	DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:RXCL1:GRAD 0.3	

Command	:PROJect:PIPEline:NODEs:coRXCaLibrate<n>:SKEW?	Summary >>
Syntax	:PROJect:PIPEline:NODEs:coRXCaLibrate<n>:SKEW?<wsp><ALL XI XQ YI YQ UNIT>	
Description	Query skew	
Parameters	ALL XI XQ YI YQ : Query all channel(s) or one of the channels	
	UNIT : Returns the unit	
Response	N/A	
Example	:PROJ:PIPE:NODE:RXCL1:SKEW? XI -> 0.0	

Command	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:SKEW	Summary >>
Syntax	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:SKEW<wsp><XI XQ YI YQ>,<value MIN MAX DEF>	
Description	Set skew	
Parameters	XI XQ YI YQ: Sets the channel to apply skew to	
	<value>: Sets this value	
	MIN: Sets the minimum value	
	MAX: Sets the maximum value	
	DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:RXCL1:SKEW XI,0.0	

Command	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:GAIN?	Summary >>
Syntax	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:GAIN?<wsp><ALL XI XQ YI YQ>	
Description	Query gain	
Parameters	ALL XI XQ YI YQ: Set the channel(s) to query: all channels or one of the channels	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:RXCL1:GAIN? XI -> 0.9	

Command	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:GAIN	Summary >>
Syntax	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:GAIN<wsp><XI XQ YI YQ>,<value MIN MAX DEF>	
Description	Set gain	
Parameters	XI XQ YI YQ: Sets the channel to apply skew to	
	<value>: Sets this value	
	MIN: Sets the minimum value	
	MAX: Sets the maximum value	
	DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:RXCL1:GAIN XI,0.9	

Command	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>DCOFFset?	Summary >>
Syntax	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:DCOFFset?<wsp><ALL XI XQ YI YQ UNIT>	
Description	Query DC offset	
Parameters	ALL XI XQ YI YQ : Set the channel(s) to query: all channels or one of the channels UNIT : Returns the unit	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:RXCL1:DCOF? XI -> 0.00000	

Command	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>DCOFFset	Summary >>
Syntax	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:DCOFFset<wsp><XI XQ YI YQ>,<value MIN MAX DEF>	
Description	Set DC offset	
Parameters	XI XQ YI YQ: Set the channel to query <value> : Sets this value MIN: Sets the minimum value MAX: Sets the maximum value DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:RXCL1:DCOF XI,0.00000	

Command	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:TRAiNing:ACQuisitions?	Summary >>
Syntax	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:TRAiNing:ACQuisitions? [<wsp><MIN MAX DEF SET ALL>]	
Description	Query the number of training acquisitions	
Parameters	None: Returns the currently set value in the default unit MIN: Returns the minimum value in the default unit MAX: Returns the maximum value in the default unit DEF: Returns the default value in the default unit SET: Returns the currently set value in the default unit ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:RXCL1:TRAN:ACQ? -> 25	

Command	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:TRaiNing:ACQuisitions	Summary >>
Syntax	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:TRaiNing:ACQuisitions<wsp><value MIN MAX DEF>	
Description	Set the number of training acquisitions	
Parameters	<value>: Sets this value	
	MIN: Sets the minimum value	
	MAX: Sets the maximum value	
	DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:RXCL1:TRAN:ACQ 25	

Command	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:TRaiNing:CONTinuous?	Summary >>
Syntax	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:TRaiNing:CONTinuous?	
Description	Query the status of continuous training acquisitions	
Parameters	N/A	
Response	ON:	
	OFF:	
Example	:PROJ:PIPE:NODE:RXCL1:TRAN:CONT? -> ON	

Command	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:TRaiNing:CONTinuous	Summary >>
Syntax	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:TRaiNing:CONTinuous<wsp><ON OFF>	
Description	Turn continuous training acquisitions ON or OFF	
Parameters	ON:	
	OFF:	
Response	N/A	
Example	:PROJ:PIPE:NODE:RXCL1:TRAN:CONT ON	

Command	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:TRaiNing:HOLD?	Summary >>
Syntax	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:TRaiNing:HOLD?	
Description	Query training acquisitions hold	
Parameters	N/A	
Response	ON:	
	OFF:	
Example	:PROJ:PIPE:NODE:RXCL1:TRAN:HOLD? -> ON	

Command	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:TRAiNing:HOLD	Summary >>
Syntax	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:TRAiNing:HOLD<wsp><ON OFF>	
Description	Hold training acquisitions	
Parameters	ON: OFF:	
Response	N/A	
Example	:PROJ:PIPE:NODE:RXCL1:TRAN:HOLD ON	

Command	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:TRAiNing:COUNt?	Summary >>
Syntax	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:TRAiNing:COUNt?	
Description	Query training acquisition count	
Parameters	N/A	
Response	A single value	
Example		

Command	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:TRAiNing:COUNt:ReSeT	Summary >>
Syntax	:PROJect:PIPELine:NODEs:coRXCaLibrate<n>:TRAiNing:COUNt:ReSeT	
Description	Reset training acquisitions count	
Parameters	N/A	
Response	N/A	
Example	:PROJ:PIPE:NODE:RXCL1:TRAN:COUN:RST	

6.3.7.4 Node: Filter

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	:PROJect:PIPELine:NODEs:FILTer<n>:TYPE?	Summary >>
Syntax	:PROJect:PIPELine:NODEs:FILTer<n>:TYPE?	
Description	Query filter type	
Parameters	N/A	
Response		
Example	:PROJ:PIPE:NODE:FILT2:TYPE? -> RRC	

Command	:PROJect:PIPELine:NODEs:FILTer<n>:TYPE	Summary >>
Syntax	:PROJect:PIPELine:NODEs:FILTer<n>:TYPE<wsp><DEF 0 GAUSSs 1 RRC 2 RCOSine 3 BRICKwall 4 SINC 5 BESSel 6 BUTTer>	
Description	Set filter type	
Parameters	DEF: 0 GAUSSs: 1 RRC: 2 RCOSine: 3 BRICKwall: 4 SINC: 5 BESSel: 6 BUTTer:	
Response	N/A	
Example	:PROJ:PIPE:NODE:FILT2:TYPE RRC	

Command	:PROJect:PIPELine:NODEs:FILTer<n>:BANDwidth?	Summary >>
Syntax	:PROJect:PIPELine:NODEs:FILTer<n>:BANDwidth? [<wsp><MIN MAX DEF SET UNIT ALL>]	
Description	Query bandwidth	
Parameters	None: Returns the currently set value in the default unit	
	MIN: Returns the minimum value in the default unit	
	MAX: Returns the maximum value in the default unit	
	DEF: Returns the default value in the default unit	
	SET: Returns the currently set value in the default unit	
	UNIT: Returns the measurement unit	
	ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:FILT2:BAND? -> 59843800	

Command	:PROJect:PIPELine:NODEs:FILTer<n>:BANDwidth	Summary >>
Syntax	:PROJect:PIPELine:NODEs:FILTer<n>:BANDwidth<wsp><value MIN MAX DEF>	
Description	Set bandwidth	
Parameters	<value>: Sets this value	
	MIN: Sets the minimum value	
	MAX: Sets the maximum value	
	DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:FILT2:BAND 59843800	

Command	:PROJect:PIPELine:NODEs:FILTer<n>:CENTer?	Summary >>
Syntax	:PROJect:PIPELine:NODEs:FILTer<n>:CENTer? [<wsp><MIN MAX DEF SET UNIT ALL>]	
Description	Query center	
Parameters	None: Returns the currently set value in the default unit	
	MIN: Returns the minimum value in the default unit	
	MAX: Returns the maximum value in the default unit	
	DEF: Returns the default value in the default unit	
	SET: Returns the currently set value in the default unit	
	UNIT: Returns the measurement unit	
	ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:FILT2:CENT? -> 0.0	

Command	:PROJect:PIPELine:NODEs:FILTer<n>:CENTer	Summary >>
Syntax	:PROJect:PIPELine:NODEs:FILTer<n>:CENTer<wsp><valueMIN MAX DEF >	
Description	Set center	
Parameters	<value> : Sets this value MIN: Sets the minimum value MAX: Sets the maximum value DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:FILT2:CEN 0.0	

Command	:PROJect:PIPELine:NODEs:FILTer<n>:ROLLoff?	Summary >>
Syntax	:PROJect:PIPELine:NODEs:FILTer<n>:ROLLoff? [<wsp><MIN MAX DEF SET ALL>]	
Description	Query roll off	
Parameters	None: Returns the currently set value in the default unit MIN: Returns the minimum value in the default unit MAX: Returns the maximum value in the default unit DEF: Returns the default value in the default unit SET: Returns the currently set value in the default unit ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:FILT2:ROLL? -> 0.950000	

Command	:PROJect:PIPELine:NODEs:FILTer<n>:ROLLoff	Summary >>
Syntax	:PROJect:PIPELine:NODEs:FILTer<n>:ROLLoff<wsp><value MIN MAX DEF>	
Description	Set roll off	
Parameters	<value> : Sets this value MIN: Sets the minimum value MAX: Sets the maximum value DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:FILT2:ROLL 0.950000	

6.3.7.5 Node: Retimer

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	:PROJect:PIPEline:NODEs:RETiMer<n>:CENTer?	Summary >>
Syntax	:PROJect:PIPEline:NODEs:RETiMer<n>:CENTer?	
Description	Query center type	
Parameters	N/A	
Response	0:SQUaRe 1:STDDev 2:CROSSpoint 3:EVM	
Example	:PROJ:PIPE:NODE:RETM2:CENT? -> EVM	

Command	:PROJect:PIPEline:NODEs:RETiMer<n>:CENTER	Summary >>
Syntax	:PROJect:PIPEline:NODEs:RETiMer<n>:CENTER<wsp><DEF 0 SQUaRe 1 STDDev 2 CROSSpoint 3 EVM>	
Description	Set center type	
Parameters	DEF 0 SQUaRe 1 STDDev 2 CROSSpoint 3 EVM	
Response	N/A	
Example	:PROJ:PIPE:NODE:RETM2:CENT EVM	

Command	:PROJect:PIPEline:NODEs:RETiMer<n>:UPSaMple?	Summary >>
Syntax	:PROJect:PIPEline:NODEs:RETiMer<n>:UPSaMple?	
Description	Query upsample type	
Parameters	N/A	
Response	1:LINear: 0:SINC:	
Example	:PROJ:PIPE:NODE:RETM2:UPSM? -> SINC	

Command	:PROJect:PIPEline:NODEs:RETiMer<n>:UPSaMple	Summary >>
Syntax	:PROJect:PIPEline:NODEs:RETiMer<n>:UPSaMple<wsp><1 LINear 0 SINC>	
Description	Set upsample type	
Parameters	1 LINear: 0 SINC:	
Response	N/A	
Example	:PROJ:PIPE:NODE:RETM2:UPSM SINC	

Command	:PROJect:PIPEline:NODEs:RETiMer<n>:SAMPles?	Summary >>
Syntax	:PROJect:PIPEline:NODEs:RETiMer<n>:SAMPles?<wsp><MIN MAX DEF SET ALL>	
Description	Query samples per Baud	
Parameters	None: Returns the currently set value in the default unit MIN: Returns the minimum value in the default unit MAX: Returns the maximum value in the default unit DEF: Returns the default value in the default unit SET: Returns the currently set value in the default unit ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:RETM2:SAMP? -> 4	

Command	:PROJect:PIPEline:NODEs:RETiMer<n>:SAMPles	Summary >>
Syntax	:PROJect:PIPEline:NODEs:RETiMer<n>:SAMPles<wsp><valueMIN MAX DEF >	
Description	Set samples per Baud	
Parameters	<value> : Sets this value None: Returns the currently set value in the default unit MIN: Returns the minimum value in the default unit MAX: Returns the maximum value in the default unit	
Response	N/A	
Example	:PROJ:PIPE:NODE:RETM2:SAMP 4	

6.3.7.6 Node: PolDeMux

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	:PROJect:PIPEline:NODEs:PolDeMuX<n>:TAPsNumber?	Summary >>
Syntax	:PROJect:PIPEline:NODEs:PolDeMuX<n>:TAPsNumber? [<wsp><MIN MAX DEF SET ALL>]	
Description	Query the number of taps	
Parameters	None: Returns the currently set value in the default unit	
	MIN: Returns the minimum value in the default unit	
	MAX: Returns the maximum value in the default unit	
	DEF: Returns the default value in the default unit	
	SET: Returns the currently set value in the default unit	
	ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:PDMX3:TAPN? -> 9	

Command	:PROJect:PIPEline:NODEs:PolDeMuX<n>:TAPsNumber	Summary >>
Syntax	:PROJect:PIPEline:NODEs:PolDeMuX<n>:TAPsNumber<wsp><value MIN MAX DEF>	
Description	Set the number of taps	
Parameters	<value>: Sets this value	
	MIN: Sets the minimum value	
	MAX: Sets the maximum value	
	DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:PDMX3:TAPN 9	

Command	:PROJect:PIPEline:NODEs:PolDeMuX<n>:SYMbolsNumber?	Summary >>
Syntax	:PROJect:PIPEline:NODEs:PolDeMuX<n>:SYMbolsNumber?<wsp><MIN MAX DEF SET ALL>	
Description	Query the number of symbols	
Parameters	MIN: Returns the minimum value in the default unit	
	MAX: Returns the maximum value in the default unit	
	DEF: Returns the default value in the default unit	
	SET: Returns the currently set value in the default unit	
	ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:PDMX3:SYMN? -> 5000	

Command	:PROJect:PIPEline:NODEs:PolDeMuX<n>:SYMbolsNumber	Summary >>
Syntax	:PROJect:PIPEline:NODEs:PolDeMuX<n>:SYMbolsNumber<wsp><value MIN MAX DEF>	
Description	Set the number of symbols	
Parameters	<value>: Sets this value	
	MIN: Sets the minimum value	
	MAX: Sets the maximum value	
	DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:PDMX3:SYMN 5000	

Command	:PROJect:PIPEline:NODEs:PolDeMuX<n>:GRADient?	Summary >>
Syntax	:PROJect:PIPEline:NODEs:PolDeMuX<n>:GRADient? [<wsp><MIN MAX DEF SET ALL>]	
Description	Query the gradient	
Parameters	None: Returns the currently set value in the default unit	
	MIN: Returns the minimum value in the default unit	
	MAX: Returns the maximum value in the default unit	
	DEF: Returns the default value in the default unit	
	SET: Returns the currently set value in the default unit	
	ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:PDMX3:GRAD? -> 0.000700	

Command	:PROJect:PIPELine:NODEs:PolDeMuX<n>:GRADient	Summary >>
Syntax	:PROJect:PIPELine:NODEs:PolDeMuX<n>:GRADient<wsp><value MIN MAX DEF>	
Description	Set the gradient	
Parameters	<value> : Sets this value MIN: Sets the minimum value MAX: Sets the maximum value DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:PDMX3:GRAD 0.0007	

Command	:PROJect:PIPELine:NODEs:PolDeMuX<n>:TAPsperBaud?	Summary >>
Syntax	:PROJect:PIPELine:NODEs:PolDeMuX<n>:TAPsperBaud? [<wsp><MIN MAX DEF SET ALL>]	
Description	Query taps per Baud	
Parameters	None: Returns the currently set value in the default unit MIN: Returns the minimum value in the default unit MAX: Returns the maximum value in the default unit DEF: Returns the default value in the default unit SET: Returns the currently set value in the default unit ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:PDMX3:TAPB? -> 4	

Command	:PROJect:PIPELine:NODEs:PolDeMuX<n>:TAPsperBaud	Summary >>
Syntax	:PROJect:PIPELine:NODEs:PolDeMuX<n>:TAPsperBaud<wsp><value MIN MAX DEF>	
Description	Set taps per Baud	
Parameters	<value> : Sets this value MIN: Sets the minimum value MAX: Sets the maximum value DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:PDMX3:TAPB 4	

Command	:PROJect:PIPEline:NODEs:PolDeMuX<n>:ReSeT? Summary >>
Syntax	:PROJect:PIPEline:NODEs:PolDeMuX<n>:ReSeT? [<wsp>INFO]
Description	Query the resetting of each capture status
Parameters	INFO : Returns supported status
Response	1 ON : Reset each capture 0 OFF : Use results from previous acquisitions
Example	:PROJ:PIPE:NODE:PDMX3:RST? -> ON :PROJ:PIPE:NODE:PDMX3:RST? INFO -> 0:OFF 1:ON

Command	:PROJect:PIPEline:NODEs:PolDeMuX<n>:ReSeT Summary >>
Syntax	:PROJect:PIPEline:NODEs:PolDeMuX<n>:ReSeT<wsp><ON OFF>
Description	Activate or deactivate resetting of each capture
Parameters	1 ON : Reset each capture 0 OFF : Use results from previous acquisitions
Response	N/A
Example	:PROJ:PIPE:NODE:PDMX3:RST ON :PROJ:PIPE:NODE:PDMX3:RST 1

6.3.7.7 Node: Equalizer

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	:PROJect:PIPEline:NODEs:EQUALizer<n>:TAPsNumber?	Summary >>
Syntax	:PROJect:PIPEline:NODEs:EQUALizer<n>:TAPsNumber? [<wsp><MIN MAX DEF SET ALL>]	
Description	Query the number of taps	
Parameters	None: Returns the currently set value in the default unit	
	MIN: Returns the minimum value in the default unit	
	MAX: Returns the maximum value in the default unit	
	DEF: Returns the default value in the default unit	
	SET: Returns the currently set value in the default unit	
	ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:EQUA4:TAPN? -> 99	

Command	:PROJect:PIPEline:NODEs:EQUALizer<n>:TAPsNumber	Summary >>
Syntax	:PROJect:PIPEline:NODEs:EQUALizer<n>:TAPsNumber<wsp><value MIN MAX DEF>	
Description	Set the number of taps	
Parameters	<value>: Sets this value	
	MIN: Sets the minimum value	
	MAX: Sets the maximum value	
	DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:EQUA4:TAPN 99	

Command	:PROJect:PIPEline:NODEs:EQUALizer<n>:SYMBOLsNumber?	Summary >>
Syntax	:PROJect:PIPEline:NODEs:EQUALizer<n>:SYMBOLsNumber? [<wsp><MIN MAX DEF SET ALL>]	
Description	Query the number of symbols	
Parameters	None: Returns the currently set value in the default unit	
	MIN: Returns the minimum value in the default unit	
	MAX: Returns the maximum value in the default unit	
	DEF: Returns the default value in the default unit	
	SET: Returns the currently set value in the default unit	
	ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:EQUA4:SYMN? -> 5000	

Command	:PROJect:PIPEline:NODEs:EQUALizer<n>:SYMBOLsNumber	Summary >>
Syntax	:PROJect:PIPEline:NODEs:EQUALizer<n>:SYMBOLsNumber<wsp><value MIN MAX DEF>	
Description	Set the number of symbols	
Parameters	<value>: Sets this value	
	MIN: Sets the minimum value	
	MAX: Sets the maximum value	
	DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:EQUA4:SYMN 5000	

Command	:PROJect:PIPEline:NODEs:EQUALizer<n>:GRADient?	Summary >>
Syntax	:PROJect:PIPEline:NODEs:EQUALizer<n>:GRADient? [<wsp><MIN MAX DEF SET ALL>]	
Description	Query the gradient	
Parameters	None: Returns the currently set value in the default unit	
	MIN: Returns the minimum value in the default unit	
	MAX: Returns the maximum value in the default unit	
	DEF: Returns the default value in the default unit	
	SET: Returns the currently set value in the default unit	
	ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:EQUA4:GRAD? -> 0.000100	

Command	:PROJect:PIPEline:NODEs:EQUALizer<n>:GRADient	Summary >>
Syntax	:PROJect:PIPEline:NODEs:EQUALizer<n>:GRADient<wsp><value MIN MAX DEF>	
Description	Set the gradient	
Parameters	<value> : Sets this value MIN: Sets the minimum value MAX: Sets the maximum value DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:EQUA4:GRAD 0.000100	

Command	:PROJect:PIPEline:NODEs:EQUALizer<n>:TAPsperBaud?	Summary >>
Syntax	:PROJect:PIPEline:NODEs:EQUALizer<n>:TAPsperBaud? [<wsp><MIN MAX DEF SET ALL>]	
Description	Query taps per Baud	
Parameters	None: Returns the currently set value in the default unit MIN: Returns the minimum value in the default unit MAX: Returns the maximum value in the default unit DEF: Returns the default value in the default unit SET: Returns the currently set value in the default unit ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:EQUA4:TAPB? -> 4	

Command	:PROJect:PIPEline:NODEs:EQUALizer<n>:TAPsperBaud	Summary >>
Syntax	:PROJect:PIPEline:NODEs:EQUALizer<n>:TAPsperBaud<wsp><value MIN MAX DEF>	
Description	Set taps per Baud	
Parameters	<value> : Sets this value MIN: Sets the minimum value MAX: Sets the maximum value DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:EQUA4:TAPB 4	

Command	:PROJect:PIPELine:NODEs:EQUALizer<n>:TRAIning:ACQuisitions?	Summary >>
Syntax	:PROJect:PIPELine:NODEs:EQUALizer<n>:TRAIning:ACQuisitions? [<wsp><MIN MAX DEF SET ALL>]	
Description	Query the number of training acquisitions	
Parameters	None: Returns the currently set value in the default unit	
	MIN: Returns the minimum value in the default unit	
	MAX: Returns the maximum value in the default unit	
	DEF: Returns the default value in the default unit	
	SET: Returns the currently set value in the default unit	
	ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:EQUA4:TRAN:ACQ? -> 25	

Command	:PROJect:PIPELine:NODEs:EQUALizer<n>:TRAIning:ACQuisitions	Summary >>
Syntax	:PROJect:PIPELine:NODEs:EQUALizer<n>:TRAIning:ACQuisitions<wsp><value MIN MAX DEF>	
Description	Set the number of training acquisitions	
Parameters	<value>: Sets this value	
	MIN: Sets the minimum value	
	MAX: Sets the maximum value	
	DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:EQUA4:TRAN:ACQ 25	

Command	:PROJect:PIPELine:NODEs:EQUALizer<n>:TRAIning:CONTInuous?	Summary >>
Syntax	:PROJect:PIPELine:NODEs:EQUALizer<n>:TRAIning:CONTInuous?	
Description	Query the status of continuous training acquisitions	
Parameters	N/A	
Response	ON:	
	OFF:	
Example	:PROJ:PIPE:NODE:EQUA5:TRAN:CONT? -> ON	

Command	:PROJect:PIPELine:NODEs:EQUALizer<n>:TRAIning:CONTinuous	Summary >>
Syntax	:PROJect:PIPELine:NODEs:EQUALizer<n>:GAIN<n>?<wsp><ON OFF>	
Description	Turn continuous training acquisitions ON or OFF	
Parameters	ON: OFF:	
Response	N/A	
Example	:PROJ:PIPE:NODE:EQUA5:TRAN:CONT ON	

Command	:PROJect:PIPELine:NODEs:EQUALizer<n>:TRAIning:HOLD?	Summary >>
Syntax	:PROJect:PIPELine:NODEs:EQUALizer<n>:TRAIning:HOLD?	
Description	Query training acquisitions hold	
Parameters	N/A	
Response	ON: OFF:	
Example	:PROJ:PIPE:NODE:EQUA5:TRAN:HOLD? -> ON	

Command	:PROJect:PIPELine:NODEs:EQUALizer<n>:TRAIning:HOLD	Summary >>
Syntax	:PROJect:PIPELine:NODEs:EQUALizer<n>:TRAIning:HOLD<wsp><ON OFF>	
Description	Hold training acquisitions	
Parameters	ON: OFF:	
Response	N/A	
Example	:PROJ:PIPE:NODE:EQUA5:TRAN:HOLD ON	

Command	:PROJect:PIPELine:NODEs:EQUALizer<n>:TRAIning:COUNt?	Summary >>
Syntax	:PROJect:PIPELine:NODEs:EQUALizer<n>:TRAIning:COUNt?	
Description	Query training acquisition count	
Parameters	N/A	
Response	A single value	
Example		

Command	:PROJect:PIPEline:NODEs:EQUALizer<n>:TRAIning:COUNT:ReSeT	Summary >>
Syntax	:PROJect:PIPEline:NODEs:EQUALizer<n>:TRAIning:COUNT:ReSeT	
Description	Reset training acquisitions count	
Parameters	N/A	
Response	N/A	
Example	:PROJ:PIPE:NODE:EQUA4:TRAN:COUN:RST	

6.3.7.8 Node: Phase estimator

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	:PROJect:PIPEline:NODEs:PHASeestimator<n>:TYPE?	Summary >>
Syntax	:PROJect:PIPEline:NODEs:PHASeestimator<n>:TYPE? [<wsp>INFO]	
Description	Query the type of algorithm used	
Parameters	INFO: Returns the type of algorithm	
Response	0:FEEDFORWARD:	
	1:VITERBIVITERBI:	
	2:IMPROVEDVITERBIVITERBI:	
Example	:PROJ:PIPE:NODE:PHAS6:TYPE? -> FEEDFORWARD	

Command	:PROJect:PIPEline:NODEs:PHASeestimator<n>:TYPE	Summary >>
Syntax	:PROJect:PIPEline:NODEs:PHASeestimator<n>:TYPE<wsp><0 FEEDFORWARD 1 VITERBIVITERBI 2 IMPROVEDVITERBIVITERBI>	
Description	Set the type of algorithm used	
Parameters	0 FEEDFORWARD:	
	1 VITERBIVITERBI:	
	2 IMPROVEDVITERBIVITERBI:	
Response	N/A	
Example	:PROJ:PIPE:NODE:PHAS6:TYPE FEEDFORWARD	
	:PROJ:PIPE:NODE:PHAS6:TYPE 0	

Command	:PROJect:PIPEline:NODEs:PHASeestimator<n>:SYMBOLsNumber?	Summary >>
Syntax	:PROJect:PIPEline:NODEs:PHASeestimator<n>:SYMBOLsNumber? [<wsp><MIN MAX DEF SET ALL>]	
Description	Query the number of symbols	
Parameters	MIN: Returns the minimum value in the default unit	
	MAX: Returns the maximum value in the default unit	
	DEF: Returns the default value in the default unit	
	SET: Returns the currently set value in the default unit	
	ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:PHAS6:SYMN? -> 10	

Command	:PROJect:PIPELine:NODEs:PHASeestimator<n>:SYMBOLsNumber	Summary >>
Syntax	:PROJect:PIPELine:NODEs:PHASeestimator<n>:SYMBOLsNumber<wsp><value MIN MAX DEF>	
Description	Set the number of symbols	
Parameters	<value>: Sets this value	
	MIN: Sets the minimum value	
	MAX: Sets the maximum value	
	DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:PHAS6:SYMN 10	

6.3.7.9 Node: MATLAB

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	:PROJect:PIPEline:NODEs:MATLab<n>:SETTings	Summary >>
Syntax	:PROJect:PIPEline:NODEs:MATLab<n>:SETTings<wsp><"FILE_PATH">	
Description	Set MATLAB script file path	
Parameters	<FILE_PATH>: MATLAB script file path	
Response	N/A	
Example	:PROJ:PIPE:NODE:MATL8:SETT "C:\Program Files\Quantifi Photonics\VisIQ\Examples\Scripts\ExampleMATLABscript.m"	

6.3.7.10 Node: Nonlinear correction

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	:PROJect:PIPEline:NODEs:NONlinearCorrection<n>:ORDeR?	Summary >>
Syntax	:PROJect:PIPEline:NODEs:NONlinearCorrection<n>:ORDeR? [<MIN MAX DEF SET ALL>]	
Description	Query order	
Parameters	None: Returns the currently set value in the default unit	
	MIN: Returns the minimum value in the default unit	
	MAX: Returns the maximum value in the default unit	
	DEF: Returns the default value in the default unit	
	SET: Returns the currently set value in the default unit	
	ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value	
Example	:PROJ:PIPE:NODE:NONC7:ORDR? -> 5	

Command	:PROJect:PIPEline:NODEs:NONlinearCorrection<n>:ORDeR	Summary >>
Syntax	:PROJect:PIPEline:NODEs:NONlinearCorrection<n>:ORDeR<value MIN MAX DEF>	
Description	Query order	
Parameters	<value>: Sets this value	
	MIN: Sets the minimum value	
	MAX: Sets the maximum value	
	DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:NONC7:ORDR 5	

Command	:PROJect:PIPEline:NODEs:NONlinearCorrection<n>:TRAIning:ACQuisitions?	Summary >>
Syntax	:PROJect:PIPEline:NODEs:NONlinearCorrection<n>:TRAIning:ACQuisitions? [<wsp><MIN MAX DEF SET ALL>]	
Description	Query the number of training acquisitions	
Parameters	None: Returns the currently set value in the default unit MIN: Returns the minimum value in the default unit MAX: Returns the maximum value in the default unit DEF: Returns the default value in the default unit SET: Returns the currently set value in the default unit ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:NONC7:TRAN:ACQ? 25	

Command	:PROJect:PIPEline:NODEs:NONlinearCorrection<n>:TRAIning:ACQuisitions	Summary >>
Syntax	:PROJect:PIPEline:NODEs:NONlinearCorrection<n>:TRAIning:ACQuisitions<wsp><value MIN MAX DEF>	
Description	Set the number of training acquisitions	
Parameters	<value>: Sets this value MIN: Sets the minimum value MAX: Sets the maximum value DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:NONC7:TRAN:ACQ 25	

Command	:PROJect:PIPEline:NODEs:NONlinearCorrection<n>:TRAIning:CONTInuous?	Summary >>
Syntax	:PROJect:PIPEline:NODEs:NONlinearCorrection<n>:TRAIning:CONTInuous?	
Description	Query the status of continuous training acquisitions	
Parameters	N/A	
Response	ON: OFF:	
Example	:PROJ:PIPE:NODE:NONC7:TRAN:CONT? -> OFF	

Command	:PROJect:PIPELine:NODEs:NONlinearCorrection<n>:TRAIning:CONTinuous	Summary >>
Syntax	:PROJect:PIPELine:NODEs:NONlinearCorrection<n>:TRAIning:CONTinuous<wsp><ON OFF>	
Description	Turn continuous training acquisitions ON or OFF	
Parameters	ON: OFF:	
Response	N/A	
Example	:PROJ:PIPE:NODE:NONC7:TRAN:CONT OFF	

Command	:PROJect:PIPELine:NODEs:NONlinearCorrection<n>:TRAIning:HOLD?	Summary >>
Syntax	:PROJect:PIPELine:NONlinearCorrection<n>:TRAIning:HOLD?	
Description	Hold training acquisitions	
Parameters	N/A	
Response	ON: OFF:	
Example	:PROJ:PIPE:NODE:NONC7:TRAN:HOLD? -> ON	

Command	:PROJect:PIPELine:NODEs:NONlinearCorrection<n>:TRAIning:HOLD	Summary >>
Syntax	:PROJect:PIPELine:NONlinearCorrection<n>:TRAIning:HOLD<wsp><ON OFF>	
Description	Hold training acquisitions	
Parameters	ON: OFF:	
Response	N/A	
Example	:PROJ:PIPE:NODE:NONC7:TRAN:HOLD ON	

Command	:PROJect:PIPELine:NODEs:NONlinearCorrection<n>:TRAIning:COUNt:ReSeT	Summary >>
Syntax	:PROJect:PIPELine:NODEs:NONlinearCorrection<n>:TRAIning:COUNt:ReSeT	
Description	Reset training acquisitions count	
Parameters	N/A	
Response	N/A	
Example	:PROJ:PIPE:NODE:NONC7:TRAN:COUN:RST	

6.3.7.11 Node: Chromatic dispersion

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	:PROJect:PIPEline:NODEs:CHROmaticdispersion<n>:PARaMeter?	Summary >>
Syntax	:PROJect:PIPEline:NODEs:CHROmaticdispersion<n>:PARaMeter? [<wsp><MIN MAX DEF SET UNIT ALL>]	
Description	Query parameter	
Parameters	None: Returns the currently set value in the default unit	
	MIN: Returns the minimum value in the default unit	
	MAX: Returns the maximum value in the default unit	
	DEF: Returns the default value in the default unit	
	SET: Returns the currently set value in the default unit	
	UNIT: Returns the measurement unit	
	ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:CHRO8:PARM?	

Command	:PROJect:PIPEline:NODEs:CHROmaticdispersion<n>:PARaMeter	Summary >>
Syntax	:PROJect:PIPEline:NODEs:CHROmaticdispersion<n>:PARaMeter<wsp><valueMIN MAX DEF >	
Description	Set parameter	
Parameters	<value>: Sets this value	
	MIN: Sets the minimum value	
	MAX: Sets the maximum value	
	DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:CHRO8:PARM MIN	

Command	:PROJect:PIPELine:NODEs:CHROmaticdispersion<n>:FIBerLength?	Summary >>
Syntax	:PROJect:PIPELine:NODEs:CHROmaticdispersion<n>:FIBerLength? [<wsp><MIN MAX DEF SET UNIT ALL>]	
Description	Query fiber length	
Parameters	None: Returns the currently set value in the default unit	
	MIN: Returns the minimum value in the default unit	
	MAX: Returns the maximum value in the default unit	
	DEF: Returns the default value in the default unit	
	SET: Returns the currently set value in the default unit	
	UNIT: Returns the measurement unit	
	ALL: Returns all the above parameters in a comma-separated string:	
Response	A single value, or a comma-separated array of values	
Example	:PROJ:PIPE:NODE:CHRO8:FIBL?	

Command	:PROJect:PIPELine:NODEs:CHROmaticdispersion<n>:FIBerLength	Summary >>
Syntax	:PROJect:PIPELine:NODEs:CHROmaticdispersion<n>:FIBerLength<wsp><valueMIN MAX DEF >	
Description	Set fiber length	
Parameters	<value>: Sets this value	
	MIN: Sets the minimum value	
	MAX: Sets the maximum value	
	DEF: Sets the default value	
Response	N/A	
Example	:PROJ:PIPE:NODE:CHRO8:FIBL MIN	

6.3.7.12 Node: IQ offset removal

Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

Command	:PROJect:PIPELine:NODEs:IQOffsetremoval<n>:CALCulatefrom?	Summary >>
Syntax	:PROJect:PIPELine:NODEs:IQOffsetremoval<n>:CALCulatefrom? [<wsp>INFO>	
Description	Query calculation algorithm	
Parameters	None: Returns the set calculation algorithm INFO : Returns supported calculation algorithms	
Response	A single value, or the mapping between numerals and text format	
Example	:PROJ:PIPE:NODE:IQOF9:CALC? INFO -> 0:ALLWAVEFORM 1:SYMBOLS	

Command	:PROJect:PIPELine:NODEs:IQOffsetremoval<n>:CALCulatefrom	Summary >>
Syntax	:PROJect:PIPELine:NODEs:IQOffsetremoval<n>:CALCulatefrom<wsp><0 allWAVEform 1 SYMBOLs>	
Description	Set calculation algorithm	
Parameters	0 allWAVEform : From entire waveform 1 SYMBOLs : From symbols	
Response	N/A	
Example	:PROJ:PIPE:NODE:IQOF9:CALC SYMBOLs	

6.3.8 Run the analysis

Command	:PROJect:ANALysis:SINGle	Summary >>
Syntax	:PROJect:ANALysis:SINGle	
Description	Run a single data acquisition	
Parameters	N/A	
Response	N/A	
Example	:PROJ:ANAL:SING	

Command	:PROJect:ANALysis:CONTinuous	Summary >>
Syntax	:PROJect:ANALysis:CONTinuous	
Description	Run continuous data acquisition	
Parameters	N/A	
Response	N/A	
Example	:PROJ:ANAL:CONT	

Command	:PROJect:ANALysis:STOP	Summary >>
Syntax	:PROJect:ANALysis:STOP	
Description	Stop continuous data acquisition	
Parameters	N/A	
Response	N/A	
Example	:PROJ:ANAL:STOP	

6.3.9 Run tests

Command	:PROJect:RUNTests:PATtern:MAPPing?	Summary >>
Syntax	:PROJect:RUNTests:PATtern:MAPPing? [<wsp>INFO]	
Description	Query data pattern mapping selection	
Parameters	INFO: Returns supported data pattern mapping	
Response	0:UNDeFined:	
	1:INDePendent:	
	2:MUltiplexedIQ:	
	3:MUltiplexedXY:	
Example	:PROJ:RUNT:PATT:MAPP? -> INDP	

Command	:PROJect:RUNTests:PATtern:MAPPING	Summary >>
Syntax	:PROJect:RUNTests:PATtern:MAPPING?<wsp><0 UNDeFined 1 INDePendent 2 MUltiplexedIQ 3 MUl tiplexedXY>	
Description	Set data pattern mapping	
Parameters	0 UNDeFined:	
	1 INDePendent:	
	2 MUltiplexedIQ:	
	3 MUltiplexedXY:	
Response	N/A	
Example	:PROJ:RUNT:PATT:MAPP INDP	
	:PROJ:RUNT:PATT:MAPP 1	

Command	:PROJect:RUNTests:PATtern:ENCOding?	Summary >>
Syntax	:PROJect:RUNTests:PATtern:ENCOding?<wsp>INFO	
Description	Query data encoding selection	
Parameters	INFO: Returns supported data encoding	
Response	0:BINary:	
	1:GRAY1:	
	2:GRAY2:	
Example	:PROJ:RUNT:PATT:ENCO? -> BIN	

Command	:PROJect:RUNTests:PATtern:ENCoding	Summary >>
Syntax	:PROJect:RUNTests:PATtern:ENCoding<wsp><0 BINary 1 GRAY1 2 GRAY2>	
Description	Set data encoding	
Parameters	0 BINary:	
	1 GRAY1:	
	2 GRAY2:	
Response	N/A	
Example	:PROJ:RUNT:PATT:ENCO BIN	

Command	:PROJect:RUNTests:PATtern:INFO?	Summary >>
Syntax	:PROJect:RUNTests:PATtern:INFO?	
Description	Query supported PRBS patterns	
Parameters	N/A	
Response	The mapping between numerals and text format	
Example	<pre>:PROJect:RUNTests:PATtern:INFO? -></pre> <pre> 0: PRBS3-1 1: PRBS6-1 2: PRBS7-1 3: PRBS8-1 4: PRBS9-1 5: PRBS10-1 6: PRBS11-1 7: PRBS12-1 8: PRBS13-1 9: PRBS14-1 10: PRBS15-1 11: PRBS16-1 12: PRBS20-1 13: PRBS23-1 14: PRBS31-1 15: PRBS3 16: PRBS6 17: PRBS7 18: PRBS8 19: PRBS9 20: PRBS10 21: PRBS11 22: PRBS12 23: PRBS13 24: PRBS14 25: PRBS15 26: PRBS16 27: PRBS20 28: PRBS23 29: PRBS31 </pre>	

Command	:PROJect:RUNTests:PATtern:SELeCt?	Summary >>
Syntax	:PROJect:RUNTests:PATtern:SELeCt?	
Description	Query test pattern selection (4/2/1 outputs)	
Parameters	N/A	
Response	Selected test pattern by channel	
Example	:PROJect:RUNTests:PATtern:SELeCt? -> Xi:PRBS3-1 Xq:PRBS6-1 Yi:PRBS7-1 Yq:PRBS8-1	

Command	:PROJect:RUNTests:PATtern:SELeCt	Summary >>
Syntax	:PROJect:RUNTests:PATtern:SELeCt<wsp><XI XQ YI YQ>,<0 1 2 3.....15>	
Description	Set test pattern selection (4/2/1 sections to select)	
Parameters	<XI XQ YI YQ>: Channel <0 1 2 3.....29>: Test pattern	
Response	N/A	
Example	:PROJ:RUNT:PATT:SELC XI,0 :PROJ:RUNT:PATT:SELC XQ,1 :PROJ:RUNT:PATT:SELC YI,2 :PROJ:RUNT:PATT:SELC YQ,3	

Command	:PROJect:RUNTests:BERtest:STATe?	Summary >>
Syntax	:PROJect:RUNTests:BERtest:STATe? [<wsp>INFO]	
Description	Query BER test state - Start / Stop	
Parameters	INFO : Returns the type of algorithm	
Response	1:ON 0:OFF	
Example	:PROJ:RUNT:BER:STAT? -> ON	

Command	:PROJect:RUNTests:BERtest:STATe	Summary >>
Syntax	:PROJect:RUNTests:BERtest:STATe<wsp><1 ON 0 OFF>	
Description	Set BER test state - Start / Stop	
Parameters	1 ON : Start 0 OFF : Stop	
Response	N/A	
Example	:PROJ:RUNT:BER:STAT ON	

Command	:PROJect:RUNTests:BERtest:ReSeT	Summary >>
Syntax	:PROJect:RUNTests:BERtest:ReSeT	
Description	Reset BER error count	
Parameters	N/A	
Response	N/A	
Example	:PROJ:RUNT:BER:RST	

Command	:PROJect:RUNTests:BERtest:ERRor:BITS?	Summary >>
Syntax	:PROJect:RUNTests:BERtest:ERRor:BITS?	
Description	Query errors/bits (4/2/1 outputs)	
Parameters	N/A	
Response	Bit error rate by channel	
Example	:PROJ:RUNT:BER:ERR:BITS? -> Xi:0.472745 Xq:0.459920 Yi:0.450150 Yq:0.452204	

Command	:PROJect:RUNTests:BERtest:ERRor:COUNt?	Summary >>
Syntax	:PROJect:RUNTests:BERtest:ERRor:COUNt?	
Description	Query error count (4/2/1 outputs)	
Parameters	N/A	
Response	Error count by channel	
Example	:PROJ:RUNT:BER:ERR:COUN? -> Xi:35812 Xq:34940 Yi:34073 Yq:34450	

Command	:PROJect:RUNTests:BERtest:BITS:COUNT?	Summary >>
Syntax	:PROJect:RUNTests:BERtest:BITS:COUNT?	
Description	Query bits count (4/2/1 outputs)	
Parameters	N/A	
Response	Bit count by channel	
Example	:PROJ:RUNT:BER:BITS:COUN? -> Xi:89820 Xq:89820 Yi:89820 Yq:89820	

Command	:PROJect:RUNTests:BERtest:CONTRol	Summary >>
Syntax	:PROJect:RUNTests:BERtest:CONTRol<wsp><SHOW HIDE>	
Description	Set BER control pop-up state - Show / Hide	
Parameters	SHOW: HIDE:	
Response	N/A	
Example	:PROJ:RUNT:BER:CONT SHOW	

Command	:PROJect:RUNTests:BERtest:LOCKing?	Summary >>
Syntax	:PROJect:RUNTests:BERtest:LOCKing?	
Description	Query the status of the BER lock indicator This command returns a valid response when: - the continuous analysis is running (:PROJect:ANALysis:CONTinuous), and - the BER test state is ON (:PROJect:RUNTests:BERtest:STATe ON). The BER lock indicator will turn true if the correlation between the received bit pattern and the reference known pattern exceeds a threshold level, indicating that there is a high confidence that the correct pattern is found and matched.	
Parameters	N/A	
Response	Lock indicator status by channel	
Example	:PROJect:RUNTests:BERtest:LOCKing? -> Xi:NOT LOCKED Xq:NOT LOCKED Yi:NOT LOCKED Yq:NOT LOCKED	

6.3.10 Select measurements

Command	:PROJect:MEASurements:INFO?	Summary >>
Syntax	:PROJect:MEASurements:INFO?	
Description	Query what measurement types are displayed in the measurement table	
Parameters	N/A	
Response	The mapping between numerals and text format	
Example	<pre>:PROJect:MEASurements:INFO? -> Evm:VISIBLE Ber:VISIBLE Snr:VISIBLE SnrDb:VISIBLE QFactor:VISIBLE QFactorDb:VISIBLE MagnitudeErrorRms:HIDDEN IQImbalanceDb:VISIBLE IQImbalancePerc:VISIBLE IQMagnitudeOffsetDb:VISIBLE IQMagnitudeOffsetPerc:VISIBLE IOffsetPerc:VISIBLE QOffsetPerc:VISIBLE PhaseErrorRadsRms:VISIBLE IQPhaseOffsetRads:VISIBLE IQPhaseErrorDeg:VISIBLE IQSkewS:VISIBLE IQSkewUI:VISIBLE XYSkewS:VISIBLE XYSkewUI:VISIBLE PmdS:HIDDEN FrequencyOffsetHz:VISIBLE LinewidthHz:VISIBLE BaudrateHz:VISIBLE</pre>	

Command	:PROJect:MEASurements:SHOW? [<wsp>INFO]	Summary >>
Syntax	:PROJect:MEASurements:SHOW?	
Description	Query what measurement types are displayed in the measurement table and mapping of supported measurement types	
Parameters	INFO: Returns supported measurement types	
Response	The mapping between numerals and text format	
Example	<pre> :PROJect:MEASurements:SHOW? -> Evm:VISIBLE Ber:HIDDEN Snr:VISIBLE SnrDb:VISIBLE QFactor:VISIBLE QFactorDb:HIDDEN MagnitudeErrorRms:HIDDEN IQImbalanceDb:VISIBLE IQImbalancePerc:HIDDEN IQMagnitudeOffsetDb:HIDDEN IQMagnitudeOffsetPerc:HIDDEN IOffsetPerc:HIDDEN QOffsetPerc:HIDDEN PhaseErrorRadsRms:HIDDEN IQPhaseOffsetRads:HIDDEN IQPhaseErrorDeg:HIDDEN IQSkewS:HIDDEN IQSkewUI:VISIBLE XYSkewS:VISIBLE XYSkewUI:VISIBLE PmdS:HIDDEN FrequencyOffsetHz:HIDDEN LinewidthHz:HIDDEN BaudrateHz:HIDDEN :PROJect:MEASurements:SHOW? INFO -> 0:ALL </pre>	

Command	:PROJect:MEASurements:SHOW? [<wsp>INFO]	Summary >>
	1:EVM 2:BER 3:SNR 4:SNRDB 5:QFACTOR 6:QFACTORDB 7:MAGNITUDEERRORRMS 8:IQIMBALANCEDB 9:IQIMBALANCEPERC 10:IQMAGNITUDEOFFSETDB 11:IQMAGNITUDEOFFSETPERC 12:IOFFSETPERC 13:QOFFSETPERC 14:PHASEERRORRADSRMS 15:IQPHASEOFFSETRADS 16:IQPHASEERRORDEG 17:IQSKEWS 18:IQSKEWUI 19:XYSKEWS 20:XYSKEWUI 21:PMDS 22:FREQUENCYOFFSETHZ 23:LINEWIDTHHZ 24:BAUDRATEHZ	

Command	:PROJect:MEASurements:SHOW	Summary >>
Syntax	:PROJect:MEASurements:SHOW<wsp><0 ALL 1 EVM 2 BER 3 SNR ... 22 FREQUENCYOFFSETHZ 23 LINEWIDTHHHZ 24 BAUDRATEHZ>	
Description	Set measurements to be displayed in the measurement table	
Parameters	0 ALL:	
	1 EVM:	
	2 BER:	
	3 SNR:	
	...	
	22 FREQUENCYOFFSETHZ:	
	23 LINEWIDTHHHZ:	
	24 BAUDRATEHZ:	
Response	N/A	
Example	:PROJ:MEAS:SHOW ALL	
	:PROJ:MEAS:SHOW 0	

Command	:PROJect:MEASurements:HIDE?<wsp>INFO	Summary >>
Syntax	:PROJect:MEASurements:HIDE?	
Description	Query supported measurement types	
Parameters	INFO : Returns supported measurement types	
Response	The mapping between numerals and text format	
Example	<pre> :PROJect:MEASurements:HIDE? INFO -> 0:ALL 1:EVM 2:BER 3:SNR 4:SNRDB 5:QFACTOR 6:QFACTORDB 7:MAGNITUDEERRORRMS 8:IQIMBALANCEDB 9:IQIMBALANCEPERC 10:IQMAGNITUDEOFFSETDB 11:IQMAGNITUDEOFFSETPERC 12:IOFFSETPERC 13:QOFFSETPERC 14:PHASEERRORRADSRMS 15:IQPHASEOFFSETRADS 16:IQPHASEERRORDEG 17:IQSKEWS 18:IQSKEWUI 19:XYSKEWS 20:XYSKEWUI 21:PMDS 22:FREQUENCYOFFSETHZ 23:LINEWIDTHHHZ 24:BAUDRATEHZ </pre>	

Command	:PROJect:MEASurements:HIDE	Summary >>
Syntax	:PROJect:MEASurements:HIDE<wsp><0 ALL 1 EVM 2 BER 3 SNR ... 22 FREQUENCYOFFSETHZ 23 LIN EWIDTHHZ 24 BAUDRATEHZ>	
Description	Set measurements not to be displayed in the measurement table	
Parameters	0 ALL:	
	1 EVM:	
	2 BER:	
	3 SNR:	
	...	
	22 FREQUENCYOFFSETHZ:	
	23 LINEWIDTHHZ:	
	24 BAUDRATEHZ:	
Response	N/A	
Example	:PROJ:MEAS:HIDE 2	
	:PROJ:MEAS:HIDE BER	

Command	:PROJect:MEASurements:MASK	Summary >>
Syntax	:PROJect:MEASurements:MASK<wsp><maskvalue>	
Description	Set bit mask to define measurements to be displayed in the measurement table (0 = OFF, 1 = ON)	
Parameters	<maskvalue> : bit mask	
Response	N/A	
Example	:PROJ:MEAS:MASK #b111111	
	:PROJ:MEAS:MASK #b100110011	

Command	:PROJect:MEASurements:ReSeT	Summary >>
Syntax	:PROJect:MEASurements:ReSeT	
Description	Reset statistical data for all measurements	
Parameters	N/A	
Response	N/A	
Example		

Command	:PROJect:MEASurements:EXPOrt	Summary >>
Syntax	:PROJect:MEASurements:EXPOrt<wsp><file_path>	
Description	Export the measurements results as a .csv file in a predefined location	
Parameters	<file_path>:	
Response	N/A	
Example		

6.3.11 Load and save data

Command	:PROJect:NEW	Summary >>
Syntax	:PROJect:NEW	
Description	Create a new project file Note: Save your current project before creating a new project file.	
Parameters	N/A	
Response	N/A	
Example		

Command	:PROJect:OPEN	Summary >>
Syntax	:PROJect:OPEN<wsp><"file_path">	
Description	Open a project file from a specified file path Note: Save your current project before opening a new project file.	
Parameters	file_path : file path to the project file	
Response	N/A	
Example	:PROJ:OPEN "C:\Program Files\Quantifi Photonics\myfiles\xyz.project"	

Command	:PROJect:SAVE	Summary >>
Syntax	:PROJect:SAVE<wsp><CURRENT "file_path">	
Description	Save the current project in the previously saved file or save as a new file	
Parameters	CURRENT : Saves the current project in the previously saved file "file_path" : Saves the current project as a new file in this location	
Response	N/A	
Example	:PROJ:SAVE "C:\Program Files\Quantifi Photonics\VisIQ\myfiles\abc.project"	

Command	:PROJect:EXPOrt:SINGle	Summary >>
Syntax	:PROJect:EXPOrt:SINGle<wsp><"file_path">	
Description	Export currently cached raw data	
Parameters	file_path : file path	
Response	N/A	
Example	:PROJ:EXPO:SING "C:\VISIQ\Export\"	

Command	:PROJect:EXPort:STARt	Summary >>
Syntax	:PROJect:EXPort:STARt<wsp><acquisitions>,<"file_path">	
Description	Start export of raw data as a series of upcoming acquisitions	
Parameters	acquisitions : number of acquisitions file_path : file path	
Response	N/A	
Example	:PROJ:EXPO:STAR 5,"C:\VISIQ\Export\"	

Command	:PROJect:EXPort:STOP	Summary >>
Syntax	:PROJect:EXPort:STOP	
Description	Stop export of raw data	
Parameters	N/A	
Response	N/A	
Example	:PROJ:EXPO:STOP	

Command	:PROJect:PIPEline:CONFIguration:EXPort	Summary >>
Syntax	:PROJect:PIPEline:CONFIguration:EXPort<wsp><"file_path">	
Description	Export the configuration of the DSP pipeline	
Parameters	"file_path" :	
Response	N/A	
Example	VISIQ\\output_scp\\pipeline.csv"	

Command	:PROJect:PIPEline:CONFIguration:IMPort	Summary >>
Syntax	:PROJect:PIPEline:CONFIguration:IMPort<wsp><"file_path">	
Description	Import the configuration of the DSP pipeline	
Parameters	"file_path" :	
Response	N/A	
Example		

6.4 SCPI programming examples

The following are simple examples of how to control your VISIQ product using SCPI commands:

We recommend that you use the *ESR? query after every command that is sent to the device. This enables you to debug unreceived or incorrect commands sent to the product.

6.4.1 Example A

The following is a brief example that demonstrates how to work with VISIQ.

```
#Connect to the VISIQ application
:APP:STATE?
:APP:CONNECT VISIQ

#Check the state of the VISIQ application to ensure it is running
#Connect to the VISIQ app

#Select the data source and connect to hardware
:PROJect:SOURce:DATA:TYPE:SCOPE?
:PROJect:SOURce:DATA:TYPE "10.10.10.34"
:PROJect:SOURce:DATA:TYPE:SCOPE:CONNECT
:PROJect:SOURce:DATA:TYPE:SCOPE:State?
:PROJect:SOURce:RECeiver:FIND?
:PROJect:SOURce:RECeiver: "10.10.10.206"
:PROJect:SOURce:RECeiver:CONNECT
:PROJect:SOURce:DATA:TYPE:SCOPE:INITialize

#Query available oscilloscopes on the network
#Select the scope with the IP address 10.10.10.34 as the source
#Connect to the selected scope
#Check that VISIQ is connected to THE scope
#Query available coherent receivers on the network
#Select the receiver with IP address 10.10.10.206
#Connect to the selected receiver
#Initialize the Oscilloscope

#Set channels
:PROJect:SOURce:DATA:CHANnel:SCALE AUTO
:PROJect:SOURce:DATA:CHANnel1:SCALE?
:PROJect:SOURce:DATA:CHANnel2:SCALE?
:PROJect:SOURce:DATA:CHANnel3:SCALE?
:PROJect:SOURce:DATA:CHANnel4:SCALE?
:PROJect:SOURce:DATA:CHANnel1:MAPPING XI
:PROJect:SOURce:DATA:CHANnel2:MAPPING XQ
:PROJect:SOURce:DATA:CHANnel3:MAPPING YI
:PROJect:SOURce:DATA:CHANnel4:MAPPING YQ

#Oscilloscope autoscale
#Query channel scale (index each channel as needed 1-4)

#Set channel mapping (for all the channels 1-4)

#Configure the signal
:PROJect:SOURce:DATA:SIGNAL:MODulation?
:PROJect:SOURce:DATA:SIGNAL:MODulation MF_16QAM
:PROJect:SOURce:DATA:SIGNAL:BAUDrate 59.84375 GBAUD
:PROJect:SOURce:DATA:SIGNAL:FREQuency 193.4145 THz
:PROJect:SOURce:DATA:SIGNAL:SYMBolesNumber 5000

#Query modulation format
#Set modulation format
#Set Baud rate
#Set Frequency
#Set number of symbols
```

```

#Configure the DSP pipeline
:PROJect:PIPELine:NODEs?                                #Query the configuraiton of the DSP Pipeline. This will return the
nodes and their position in the pipeline.
:PROJect:PIPELine:CONFIguration:ADD 3,NONLINEARCORRECTION #Add node Nonlinear Correction into pipeline position 3
:PROJect:PIPELine:CONFIguration:REMOve 5                #Remove the node that sits in position 5; other nodes in the
pipeline will shift accordingly.

#Set node properties: CoRx node
:PROJect:PIPELine:NODEs:CORX0:RECEiver IQRX             #Select receiver type
:PROJect:PIPELine:NODEs:CORX0:SETTings "C:\MyFiles\rec.crs" #Set file path to receiver settings file

#Set node properties: CoRx auto calibrate node
:PROJect:PIPELine:NODEs:coRXCaLibrate1:GRADient 0.3      #Set the gradient
:PROJect:PIPELine:NODEs:coRXCaLibrate1:TRAIning:COUNT:ReSeT #Reset the training counter
:PROJect:PIPELine:NODEs:coRXCaLibrate1:TRAIning:HOLD?    #Query if training acquisitions are on hold

#Set node properties: Retimer node
:PROJect:PIPELine:NODEs:RETImer2:CENter EVM             #Set the center type
:PROJect:PIPELine:NODEs:RETImer2:UPSaMple SINC          #Set upsample type
:PROJect:PIPELine:NODEs:RETImer2:SAMPles 4              #Set samples per Baud

#Set node properties: PolDeMux node
:PROJect:PIPELine:NODEs:PolDeMuX3:TAPsNumber 9          #Set the number of taps
:PROJect:PIPELine:NODEs:PolDeMuX3:SYMboLsNumber 5000    #Set the number of symbols
:PROJect:PIPELine:NODEs:PolDeMuX3:GRADient 0.0007       #Set the gradient
:PROJect:PIPELine:NODEs:PolDeMuX3:TAPsperBaud 4         #Set taps per Baud
:PROJect:PIPELine:NODEs:PolDeMuX3:ReSeT ON              #Activate resetting of each capture

#Set node properties: Equalizer node
:PROJect:PIPELine:NODEs:EQUAlizer4:TRAIning:COUNT:ReSeT #Reset training aquisitions count
:PROJect:PIPELine:NODEs:EQUAlizer4:TAPsNumber 99        #Set the number of taps
:PROJect:PIPELine:NODEs:EQUAlizer4:SYMboLsNumber 5000   #Set the number of symbols
:PROJect:PIPELine:NODEs:EQUAlizer4:GRADient 0.000100    #Set the gradient
:PROJect:PIPELine:NODEs:EQUAlizer4:TAPsperBaud 4        #Set taps per Baud
:PROJect:PIPELine:NODEs:EQUAlizer4:TRAIning:ACQuisitions 25 #Set the number of training acquisitions
:PROJect:PIPELine:NODEs:EQUAlizer4:TRAIning:HOLD?       #Query if training acquisitions are on hold

#Set node properties: Phase estimator node
:PROJect:PIPELine:NODEs:PHASeestimator6:TYPE FEEDFORWARD #Set the type of algorithm used
:PROJect:PIPELine:NODEs:PHASeestimator6:SYMboLsNumber 10 #Set the number of symbols

#Set node properties: Nonlinear correction mode
:PROJect:PIPELine:NODEs:NONlinearCorrection7:TRAIning:COUNT:ReSeT #Reset training acquisitions count
:PROJect:PIPELine:NODEs:NONlinearCorrection7:ORDeR 5         #Set order
:PROJect:PIPELine:NODEs:NONlinearCorrection7:TRAIning:ACQuisitions 25 #Set the number of training acquisitions
:PROJect:PIPELine:NODEs:NONlinearCorrection7:TRAIning:CONTInuous ON #Turn continuous training acquisitions ON

```

```
#Set measurements to be displayed in the measurement table in the UI
:PROJect:MEASurements:SHOW ALL           #Display all measurements

#Run the analysis
:PROJect:ANALysis:SINGLE                   #Run a single data acquisition
:PROJect:ANALysis:CONTinuous              #Run continuous data acquisition
:PROJect:ANALysis:STOP                    #Stop continuou data acquisition

#Save data
:PROJect:SAVE CURRENT                     #Save the current project in the previously saved file

#Close connection
:PROJect:SOURce:DATA:TYPE:SCOPE:DISConnect #Disconnect from the connected oscilloscope
:PROJect:SOURce:RECeiver:DISConnect        #Disconnect from the connected receiver
```

6.5 Programming applications

Remote communication with the CohesionSCPI service is achieved through the Standard Commands for Programmable Instruments (SCPI). Support for VISA I/O API over TCP/IP is provided by the VXI-11 compliant CohesionSCPI service. With VISA communication drivers installed on the client, the implementation of VISA programming within environments such as MATLAB becomes available.

NOTE

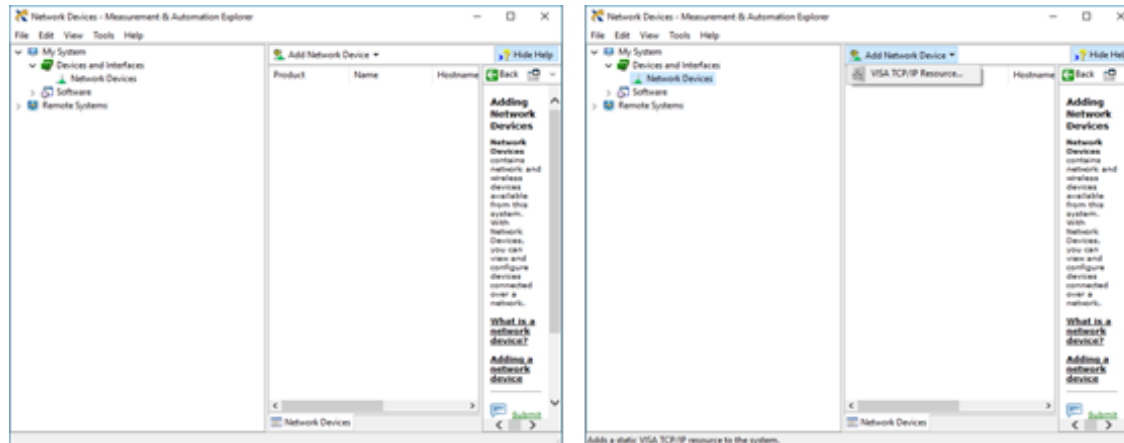
In NI-MAX a RIO interface will show up, however there are no communication methods available or implemented on this interface.

Quantifi Photonics products are **ONLY** accessible through the **VISA TCPIP INSTR** interface provided by the CohesionSCPI service installed on the system.

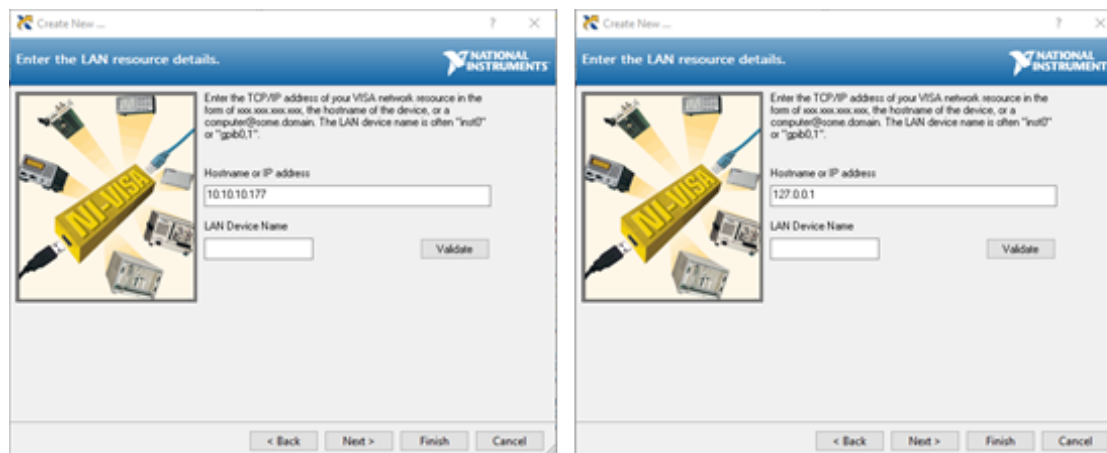
6.5.1 Setting up NI-MAX application

To communicate with any Quantifi Photonics product, the chassis / benchtop product must first be setup as a TCP/IP instrument.

1. After installing NI-MAX, launch the application. In the left side panel of the window, click the **Devices and Interfaces** option. A drop down of available instruments detected will show up.
2. Click on **Network Devices**, then click **Add Network Devices** and select **VISA TCP/IP Resource**.



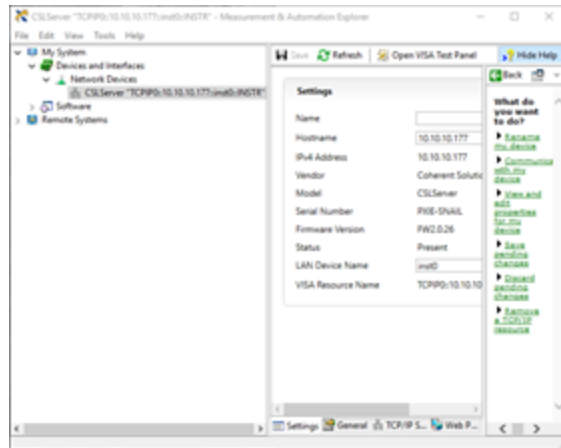
3. Select **Manual Entry of LAN Instrument**. Enter in the Hostname or IP Address.
Note when operating locally, enter in the localhost IP address of **127.0.0.1**. Click **Finish** to end the setup process.



6.5.2 Setting up NI-VISA application

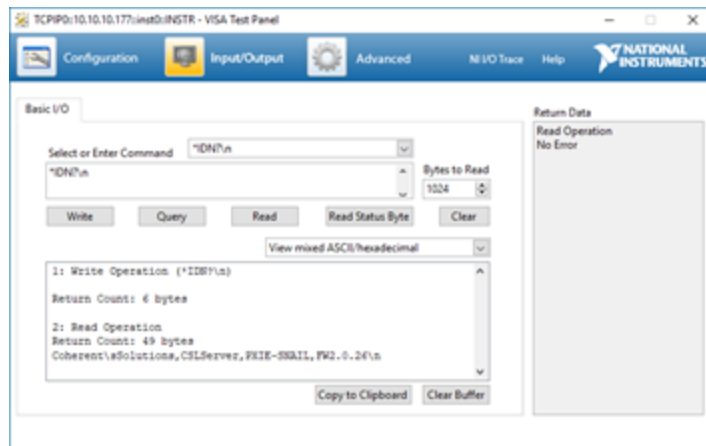
NI-VISA is used to communicate with the PXIe chassis or installed modules / instruments. The above steps must be completed before attempting to communicate using NI-VISA.

1. Launch NI-MAX. In the left-hand side menu, select an instrument from the **Network Devices** list.



2. On the right-hand side panel, select **Open VISA Test Panel**. A new window will popup. Click the **Input / Output** button from the window menu.

Valid chassis and module commands can be entered in, and their returns queried



6.5.3 Python® 2.7 code example

The following example shows how to communicate with the Quantifi Photonics product using Python code.

```
# You can get VXI11 from pip:
# pip install python-vxi11==0.9
import vxi11
from vxi11.vxi11 import Vxi11Exception
# replace this with the IP of your device
ip = "127.0.0.1"
try:
    print("connecting to " + ip + " ... ")
    instrument = vxi11.Instrument(ip)
    print("connected")
    print("checking IDN...")
    command = "*IDN?"
    data = instrument.ask(command)
    print("IDN: " + data)
    print("checking OPT...")
    command = "*OPT?"
    data = instrument.ask(command)
    print("OPT: " + data)
    # replace this with a valid command for your device (read # the programming guide section for examples)
    command = ""
    print("writing a specific command")
    instrument.write(command)
    print("checking ESR")
    command = "*ESR?"
    data = instrument.ask(command)
    print("*ESR?: " + data)
except Vxi11Exception as e:
    # pass
    print("ERROR" + str(e) + ", command: " + str(command))
```

6.5.4 MATLAB® code example

To communicate with the Quantifi Photonics product in MATLAB® the installation of a VISA IO driver is required. These drivers enable the creation of the Interface Object for instrument communication.

If developing locally on the PXI Platform, then these will already be installed. However, if development is on a remotely connected system the VISA Libraries, e.g. National Instruments NI-VISA will have to be installed.

NOTE

MATLAB 2010x or later with the Instrument Control Toolbox is required to execute the code detailed in this section.

The following example shows how to communicate with a PXI product using MATLAB code.

```
% Find a VISA-TCPIP object. This is if the VISA object has already been
% created with tmtool or has been removed from the workspace without
% first being closed (cleanly disconnected).
PXIE_Chassis = instrfind('Type', 'visa-tcpip', ...
    'RsrcName', 'TCPIP0::10.10.10.89::inst0::INSTR', 'Tag', '');
% Create the 'agilent' VISA-TCPIP object if it does not exist
% otherwise use the object that was found.
if isempty(PXIE_Chassis)
    PXIE_Chassis = visa('agilent', 'TCPIP0::10.10.10.89::inst0::INSTR');
else
    fclose(PXIE_Chassis);
    PXIE_Chassis = PXIE_Chassis (1);
end
% Open the connection to the VISA object.
fopen(PXIE_Chassis);
% Query the PXIE_Chassis.
response = query(PXIE_Chassis, '*IDN?');
disp('The *IDN query response:');
disp(response);
response = query(PXIE_Chassis, '*OPT?');
disp('The *OPT query response:');
disp(response);
% Replace this with a valid command for your device (read the programming
% guide section for examples)
command = ''
% Close the connection to the object.
```

7 System requirements

You will require access to the following hardware:

Required hardware components:

- Compatible real-time oscilloscope(s) with four input channels
- Coherent optical receiver
- Coherent modulation signal source
- Computer that will run VISIQ

Compatible oscilloscopes include:

- Tektronix SX series real-time oscilloscopes
- Tektronix DX series real-time oscilloscopes
- TeledyneLeCroy LabMaster 10Zi real-time oscilloscopes
- TeledyneLeCroy LabMaster 9Zi real-time oscilloscopes
- Keysight Infiniium UXR-Series real-time oscilloscopes
- Keysight Infiniium V-Series real-time oscilloscopes

Computer running VISIQ - minimum requirements:

- Operating system: Microsoft Windows® 10 (64-bit)
- Processor: Intel® Core™ i5 or faster CPU
- Memory: 8 GB or greater

8 Technical Support

8.1 Contacting the Technical Support Group

To obtain after-sales service or technical support for this product, contact Quantifi Photonics:

support@quantifiphotonics.com

To accelerate the process, please confirm that your setup meets the minimum system requirements (refer to System requirements) and provide all relevant information, including:

- VISIQ version number and license type
To access this information: Click the VISIQ logo and select **About VISIQ**.
- Hardware ID
This is the hardwareid.hid file that you have generated when activating VISIQ.
- A description of the issue, including the project file and relevant screen captures if possible.

Test. Measure. Solve.

Quantifi Photonics is transforming the world of photonics test and measurement. Our portfolio of optical and electrical test instruments is rapidly expanding to meet the needs of engineers and scientists around the globe. From enabling ground-breaking experiments to driving highly efficient production testing, you'll find us working with customers to solve complex problems with optimal solutions.

To find out more, get in touch with us today.

General Enquiries	sales@quantifiphotonics.com
Technical Support	support@quantifiphotonics.com
Phone	+64 9 478 4849
North America	+1-800-803-8872