

VISIQ™

Coherent Signal Analysis
Software

VISIQ version 1.0.6

PROGRAMMING GUIDE



Copyright © 2023 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 Edge®, and Microsoft Windows® are trademarks of the Microsoft Corporation. Google Chrome™ is a trademark of Google LLC.

Units of Measurement

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

User manual version: 1.04

Table of contents

1 Conventions	7
2 VISIQ overview	8
2.1 Programming interfaces	9
2.2 DSP pipeline and nodes	9
3 SCPI overview	11
3.1 Programming conventions	11
3.2 Status and event registers	12
3.2.1 Standard Event Status Register	12
3.2.2 Standard Event Status Enable Register (Mask)	12
3.2.3 Status Byte Register	12
3.2.4 Service Request Enable Register (Mask)	13
3.2.5 Status and event registers diagram	14
4 SCPI commands summary	15
4.1 Common commands	15
4.2 System menu functions	16
4.2.1 Commands	16
4.3 Select the data source	17
4.3.1 Commands	17
4.3.2 UI reference	18
4.4 Connect VISIQ with the coherent receiver	19
4.4.1 Commands	19
4.4.2 UI reference	20
4.5 Set channels	21
4.5.1 Commands	21
4.5.2 UI reference	21
4.6 Configure signal properties	22

4.6.1 Commands	22
4.6.2 UI reference	23
4.7 Configure the DSP pipeline	24
4.7.1 Pipeline configuration	24
4.7.2 Node: CoRx	26
4.7.3 Node: CoRx auto calibrate	28
4.7.4 Node: Filter	30
4.7.5 Node: Retimer	32
4.7.6 Node: PolDeMux	34
4.7.7 Node: Equalizer	36
4.7.8 Node: Phase estimator	38
4.7.9 Node: MATLAB	40
4.7.10 Node: Nonlinear correction	41
4.7.11 Node: Chromatic dispersion	43
4.7.12 Node: IQ offset removal	44
4.8 Run the analysis	45
4.8.1 Commands	45
4.8.2 UI reference	45
4.9 Run tests	46
4.9.1 Commands	46
4.9.2 UI reference	47
4.10 Select measurements	48
4.10.1 Commands	48
4.10.2 UI reference	48
4.10.3 Measurement details	49
4.11 Load and save data	55
4.11.1 Commands	55

5 SCPI commands descriptions	56
5.1 Common commands	56
5.2 System menu functions	58
5.3 Select the data source	59
5.4 Connect VISIQ with the coherent receiver	63
5.5 Set channels	67
5.6 Configure signal properties	69
5.7 Configure the DSP pipeline	76
5.7.1 Pipeline configuration	76
5.7.2 Node: CoRx	78
5.7.3 Node: CoRx auto calibrate	79
5.7.4 Node: Filter	84
5.7.5 Node: Retimer	87
5.7.6 Node: PolDeMux	89
5.7.7 Node: Equalizer	93
5.7.8 Node: Phase estimator	99
5.7.9 Node: MATLAB	101
5.7.10 Node: Nonlinear correction	102
5.7.11 Node: Chromatic dispersion	105
5.7.12 Node: IQ offset removal	107
5.8 Run the analysis	108
5.9 Run tests	109
5.10 Select measurements	115
5.11 Load and save data	122
6 SCPI programming examples	124
6.1 Example A	124
7 Programming examples and applications	127

7.1 Setting up NI-MAX application	128
7.2 Setting up NI-VISA application	129
7.3 Python® 2.7 code example	130
7.4 MATLAB® code example	131

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.

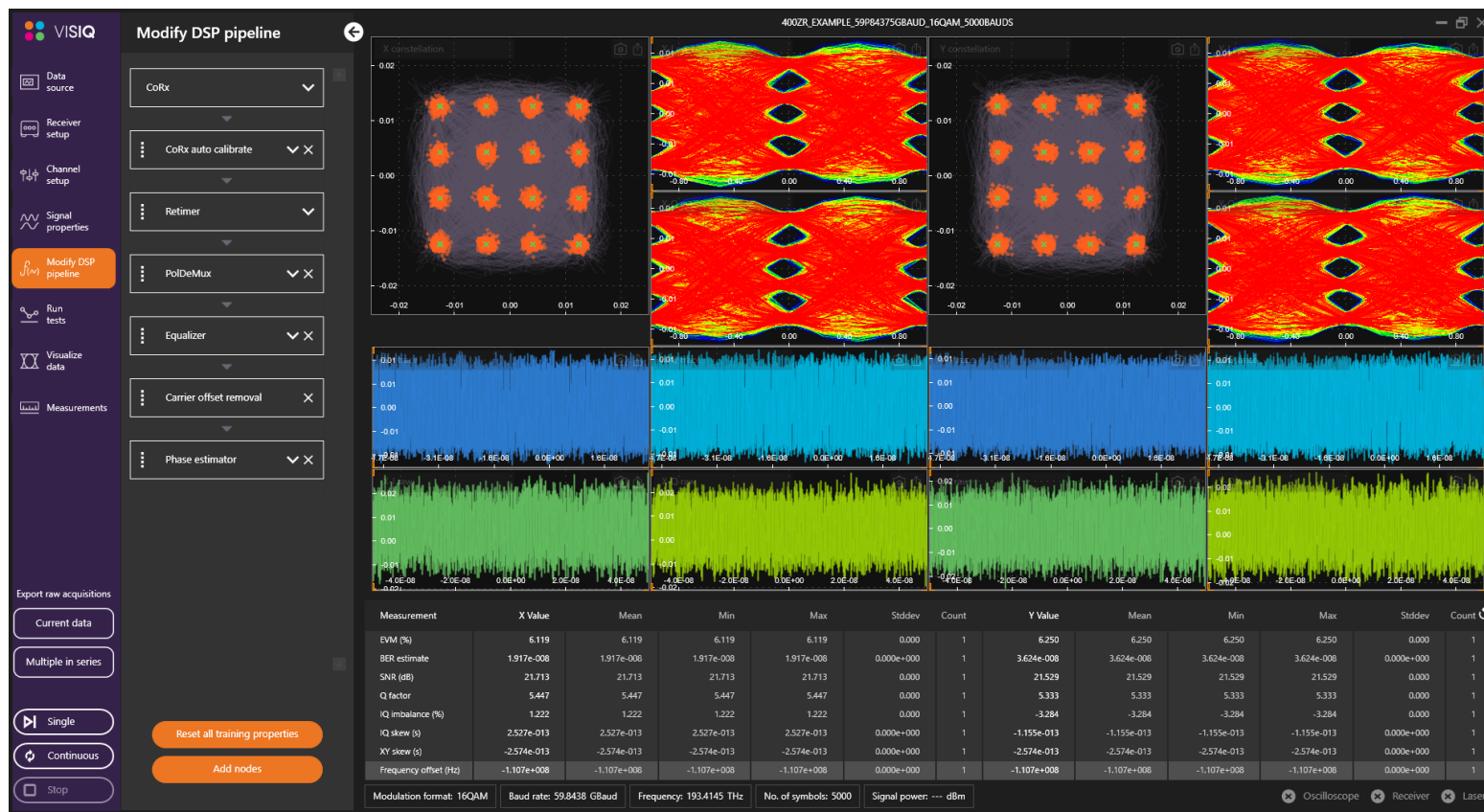
2 VISIQ overview

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 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 examples and applications](#).

3.1 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.

3.2 Status and event registers

3.2.1 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

3.2.2 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.

3.2.3 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.

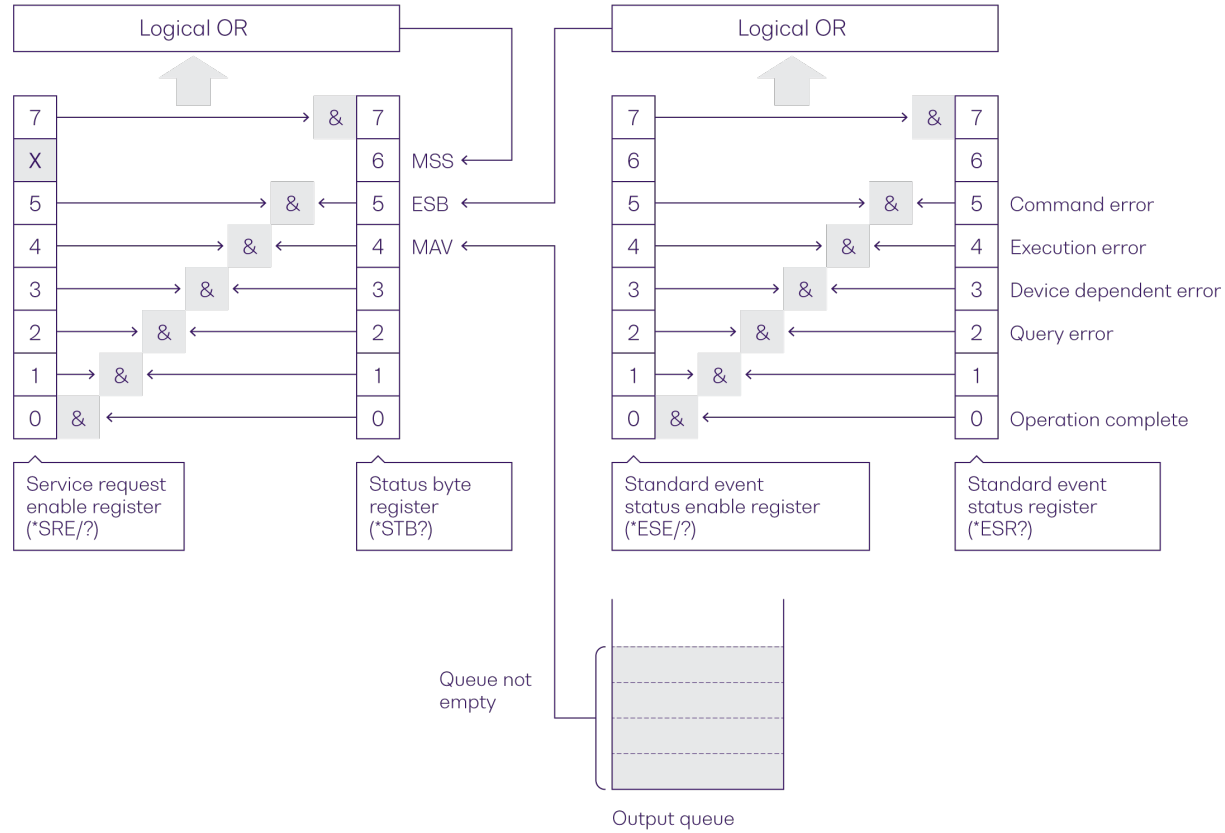
Bit	Description
7 (MSB)	Not used
6	The Master Summary Status (MSS) bit is set from the STB and SRE Mask
5	The Event Status Bit (ESB) is set from the SESR and the SESR Mask
4	Message Available (MAV) is set when there is data in the output queue
3, 2, 1, 0 (LSB)	Not used

3.2.4 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

3.2.5 Status and event registers diagram



4 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.

4.1 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 >>

4.2 System menu functions

4.2.1 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 >>

4.3 Select the data source

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

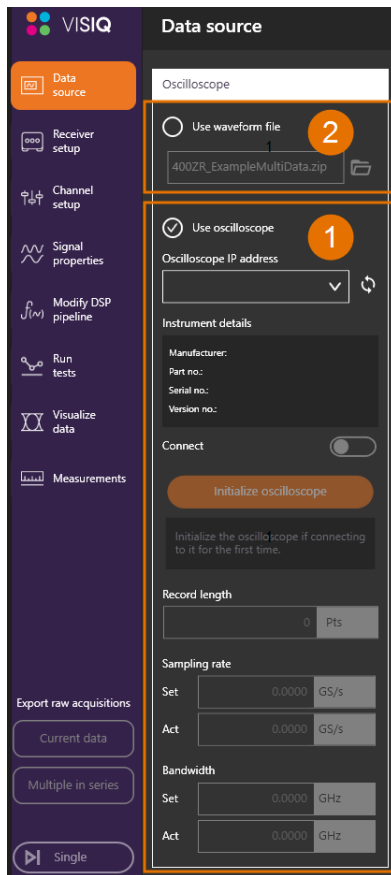
Example waveform files are included in the VISIQ software package in folder ...\\Quantifi Photonics\\VisIQ\\Examples\\Projects.

4.3.1 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_LENth?	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 >>

4.3.2 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).

4.4 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**.

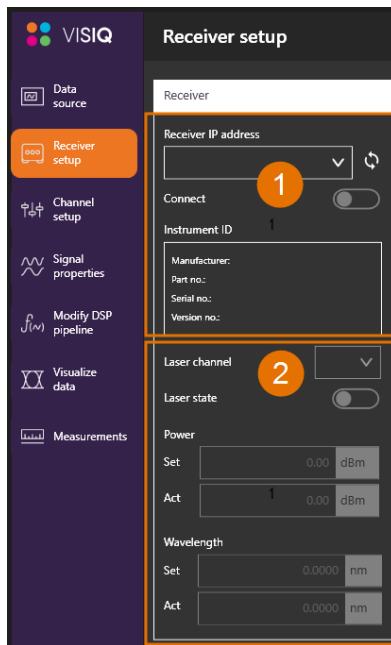
4.4.1 Commands

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 >>

4.4.2 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.

4.5 Set channels

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

4.5.1 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 >>

4.5.2 UI reference



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

4.6 Configure signal properties

VISIQ can process single and dual polarization signals.

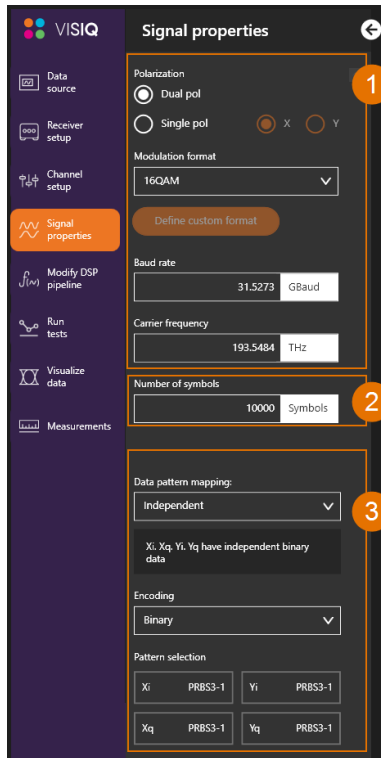
For details on setting test parameters and relevant SCPI commands, refer to [Run tests](#).

4.6.1 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 >>

4.6.2 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.

4.7 Configure the DSP pipeline

4.7.1 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.

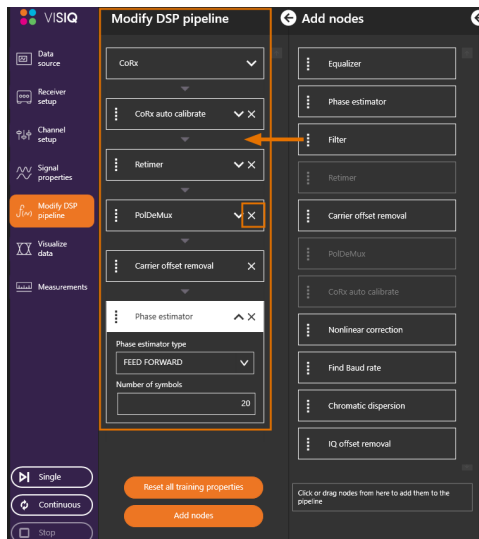
4.7.1.1 Commands

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 >>

4.7.1.2 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**.

4.7.2 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.

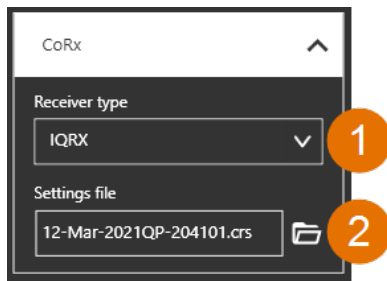
4.7.2.1 Commands

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 >>

4.7.2.2 UI reference



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

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

4.7.3 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.

4.7.3.1 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 >>

4.7.3.2 UI reference

The screenshot shows the 'CoRx auto calibrate' interface with the following sections and values:

- Gradient:** 0.300000 (highlighted with orange circle 1)
- Skew:**
 - Ch 1: 0.0000E+000 s
 - Ch 2: -1.7797E-012 s
 - Ch 3: 0.0000E+000 s
 - Ch 4: 3.7518E-013 s
- Gain:** (highlighted with orange circle 2)
 - Ch 1: 1.0000E+000
 - Ch 2: 9.8129E-001
 - Ch 3: 9.1048E-001
 - Ch 4: 9.1167E-001
- DC offset:**
 - Ch 1: 2.0876E-003 V
 - Ch 2: 5.4471E-004 V
 - Ch 3: -7.7146E-004 V
 - Ch 4: 1.7685E-003 V
- Training acquisitions:** 20 (highlighted with orange circle 3)
- Training counter:** 14
- Buttons:** Continuous training (radio button), Hold (radio button), and a large orange 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.

4.7.4 Node: Filter

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

4.7.4.1 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 >>

4.7.4.2 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 .

4.7.5 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.

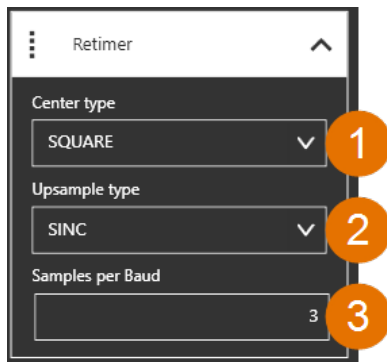
4.7.5.1 Commands

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 >>

4.7.5.2 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.

4.7.6 Node: PolDeMux

This node demultiplexes the two polarizations of the carrier.

4.7.6.1 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 >>

4.7.6.2 UI reference

The screenshot shows the PolDeMux settings panel. It contains four input fields and a checkbox, each with a numbered orange circle next to it:

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

► 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.

4.7.7 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.

4.7.7.1 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 >>

4.7.7.2 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.

4.7.8 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.

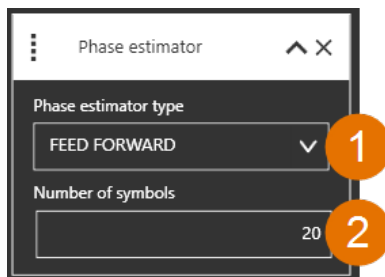
4.7.8.1 Commands

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 >>

4.7.8.2 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.

4.7.9 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.

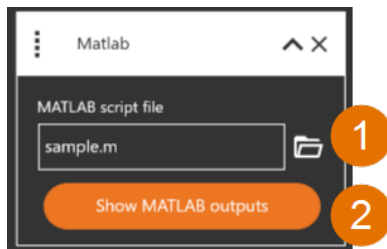
4.7.9.1 Commands

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

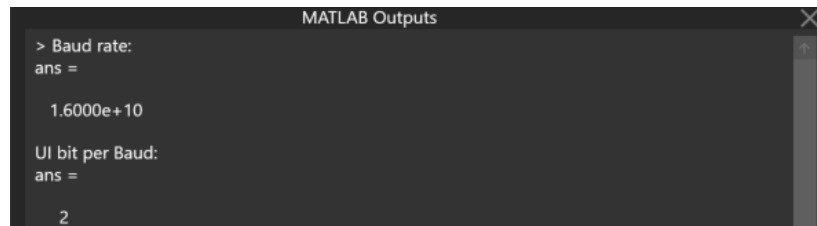
Index <n>: The position of the node in the pipeline, this is an integer <0 to 15>.

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

4.7.9.2 UI reference



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



4.7.10 Node: Nonlinear correction

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

4.7.10.1 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 >>

4.7.10.2 UI reference

Nonlinear correction

Order: 3

Training acquisitions: 20

Training counter: 20

☒ Continuous training

☐ Hold

Reset

► 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.

4.7.11 Node: Chromatic dispersion

This node compensates for chromatic dispersion due to fiber propagation.

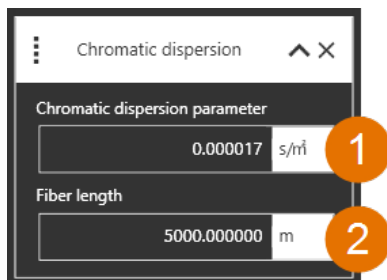
4.7.11.1 Commands

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 >>

4.7.11.2 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.

4.7.12 Node: IQ offset removal

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

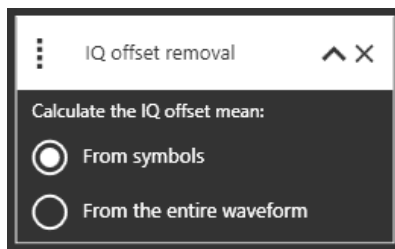
4.7.12.1 Commands

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 >>

4.7.12.2 UI reference



▶ To set the node properties:

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

4.8 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.

4.8.1 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 >>

4.8.2 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.

4.9 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.

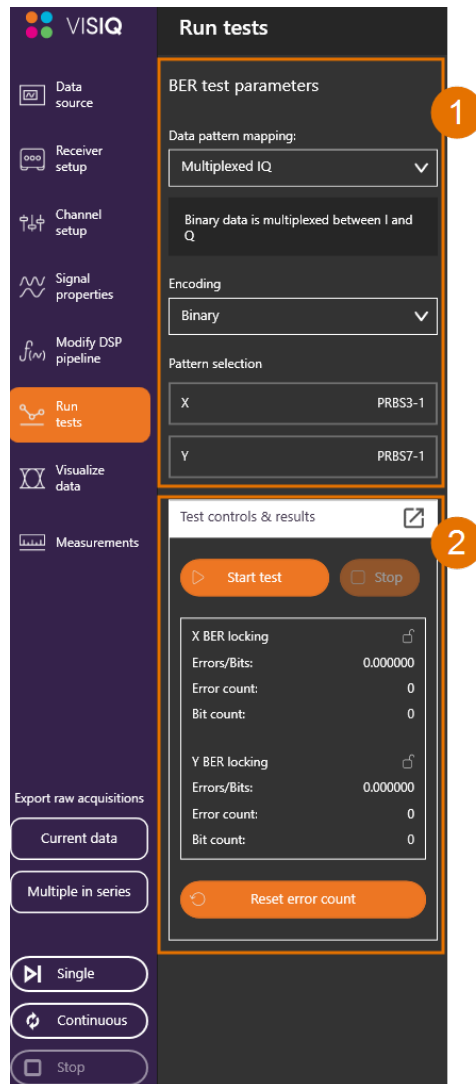
Changing the pattern in **Run tests**, will also change the pattern in **Signal properties**.

4.9.1 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 >>

4.9.2 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

4.10 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.

4.10.1 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 >>

4.10.2 UI reference

The screenshot shows the VISIQ UI interface. On the left, a sidebar contains navigation options: Instrument control, Signal properties, Pipeline, Visualization, and Measurements (highlighted with an orange circle labeled '1'). The main area is titled 'Measurements' and features a list of measurement types with checkboxes. Under the 'Noise' section, EVM (%), SNR, Q factor, IQ imbalance (dB), and Q factor (dB) are checked. Under the 'Magnitude' section, Magnitude error (%) and IQ imbalance (dB) are checked. To the right, a table displays measurement results. The table has columns: Measurement, X Value, Mean, Min, Max, Stddev, and Count. The table is highlighted with an orange circle labeled '2'. Below the table, a status bar shows modulation format: 16QAM, Symbol rate: 31.5273 GBaud, Frequency: 193.5484 THz, No. of symbols: 10000, and Signal power: --- dBm.

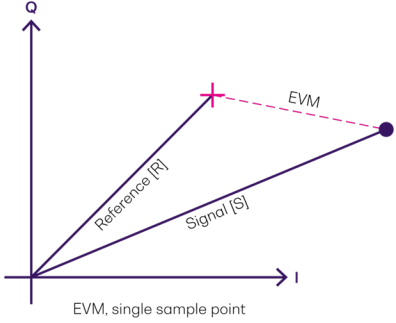
Measurement	X Value	Mean	Min	Max	Stddev	Count
EVM (%)	5.999	6.111	5.999	6.247	0.103	3
SNR	1.544e+002	1.489e+002	1.424e+002	1.544e+002	4.964e+000	3
SNR (dB)	21.886	21.727	21.534	21.886	0.146	3
Q factor	5.557	5.457	5.336	5.557	0.091	3
IQ imbalance (dB)	-0.029	-0.029	-0.029	-0.028	0.000	3
IQ skew (UI)	0.014	0.014	0.014	0.014	0.000	3
XY skew (s)	1.002e-012	1.102e-012	1.002e-012	1.215e-012	8.727e-014	3
XY skew (UI)	0.032	0.035	0.032	0.038	0.003	3

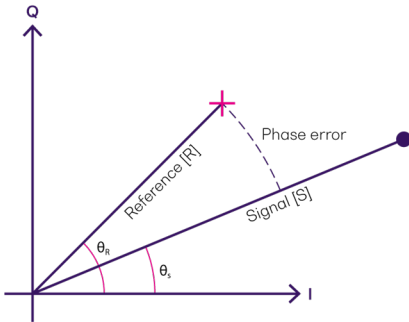
Modulation format: 16QAM Symbol rate: 31.5273 GBaud Frequency: 193.5484 THz No. of symbols: 10000 Signal power: --- dBm

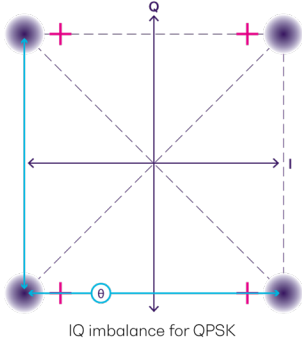
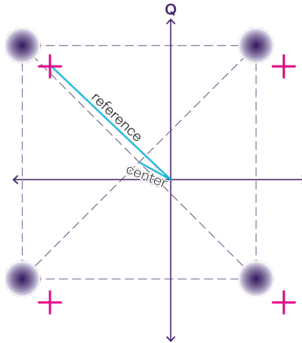
► To select measurements:

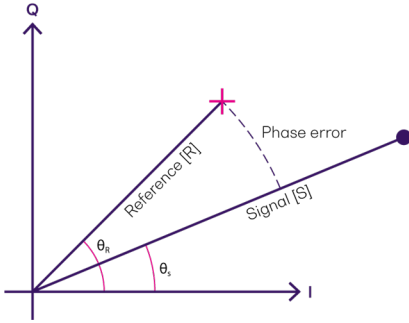
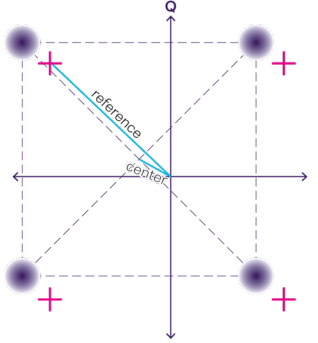
1. Select the measurements to display in the table.
2. Reset measurement values.

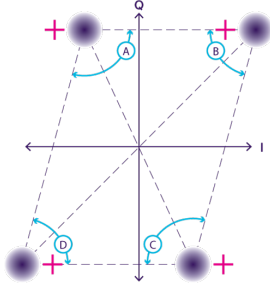
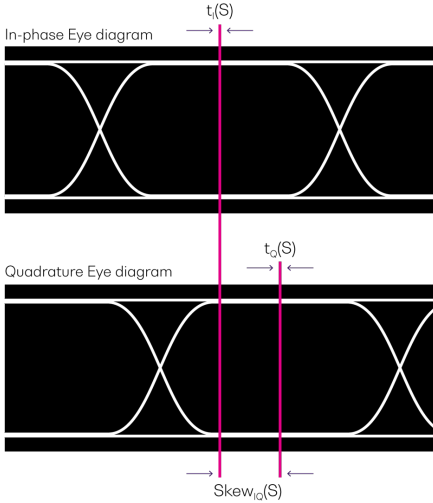
4.10.3 Measurement details

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 $EVM (\%) = 100 \times \frac{rms[signal - reference]}{\max [reference]}$  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. $SNR = \frac{Signal \ Power}{Noise \ Power}$

Type	Measurement	Details
	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. $Q\ factor_{(\%)} = \sqrt{\frac{3 \log_2 L}{L^2 - 1} \frac{2}{(EVM\%/100)^2 \log_2 2^{bits_per_symbol}}}$ 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. $\frac{Magnitude\ Error_{RMS}}{Magnitude\ Error = S - R }$ $= \frac{1}{\sqrt{N}} \sqrt{(Magnitude\ Error_1^2 + \dots + Magnitude\ Error_N^2)}$ 
	IQ imbalance (dB)	The ratio of the in-phase component vs. the quadrature component of the constellation points in dB.

Type	Measurement	Details
	IQ imbalance (%)	The ratio of the in-phase component vs. the quadrature component of the constellation points in percentage.  IQ imbalance for QPSK
	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 (%)	

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

Type	Measurement	Details
	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.

4.11 Load and save data

VISIQ offers several options to load and save data.

4.11.1 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 >>

5 SCPI commands descriptions

5.1 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	

5.2 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		

5.3 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		

5.4 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_ADDRESS">	
Description	Select a coherent receiver	
Parameters	<IP_ADDRESS>: 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		

5.5 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	

5.6 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		

5.7 Configure the DSP pipeline

5.7.1 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>	

5.7.2 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"	

5.7.3 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	

5.7.4 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 GAUSS 1 RRC 2 RCOSine 3 BRICKwall 4 SINC 5 BESSEL 6 BUTTer>	
Description	Set filter type	
Parameters	DEF:	
	0 GAUSS:	
	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	

5.7.5 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	

5.7.6 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

5.7.7 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	

5.7.8 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	

5.7.9 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"	

5.7.10 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	

5.7.11 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	

5.7.12 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	

5.8 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	

5.9 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	

5.10 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 LIN EWIDTHHZ 24 BAUDRATEHZ>	
Description	Set measurements 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: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 LINEWIDTHHHZ 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 LINEWIDTHHHZ:	
	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		

5.11 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_scpil\\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 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.

6.1 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

#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

#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

#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:SYMBOLsNumber 5000

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

#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

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

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

#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

```

7 Programming examples and 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

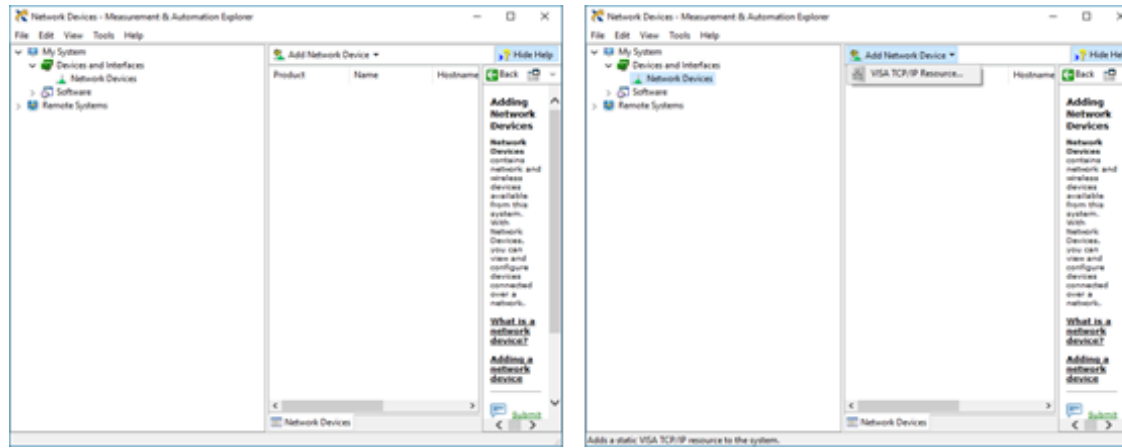
In NI-MAX a RIO interface will show up, however there are no communication methods available or implemented on this interface.

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

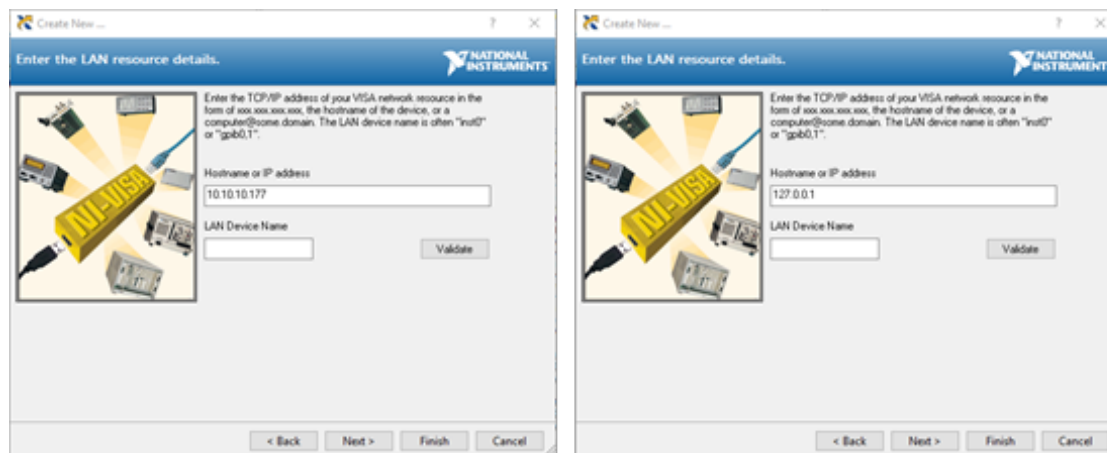
7.1 Setting up NI-MAX application

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**.



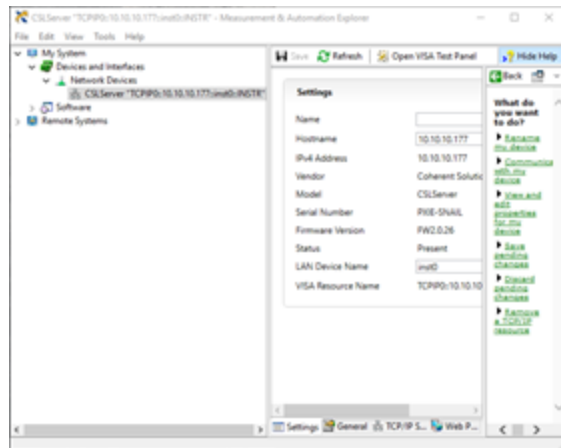
3. Select **Manual Entry of LAN Instrument**. Enter in the Hostname or IP Address.
Note when operating locally, enter in the localhost IP address of **127.0.0.1**. Click **Finish** to end the setup process.



7.2 Setting up NI-VISA application

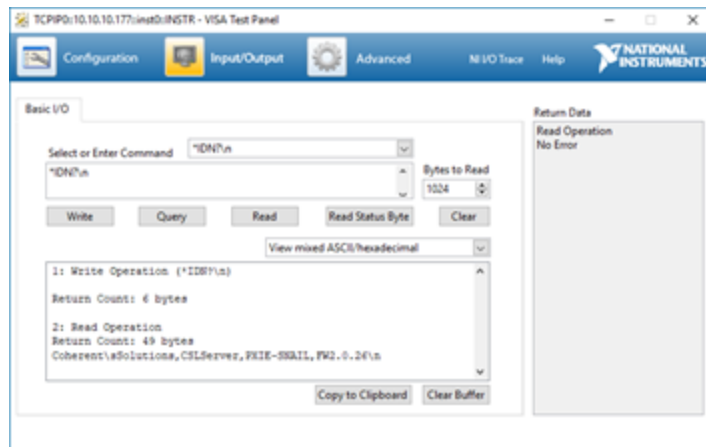
NI-VISA is used to communicate with the PXIe chassis or installed modules / instruments. The above steps must be completed before attempting to communicate using NI-VISA.

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



7.3 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))
```

7.4 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.
```

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