Keysight N109X-Series DCA-M Sampling Oscilloscopes

KEYSIGHT TECHNOLOGIES

User's Guide

Notices

Copyright Notice

Manual Part Number

N1090-90006

Edition

Edition 1, September 2016

Published by:

Keysight Technologies, Inc.
1400 Fountaingrove Parkway
Santa Rosa, CA, 95403

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Keysight disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Keysight shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Keysight and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

U.S. Government Rights

The Software is "commercial computer software," as defined by Federal Acquisition Regulation ("FAR") 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement ("DFARS") 227.7202, the U.S. government acquires commercial computer software under the same terms by which the software is customarily provided to the public. Accordingly, Keysight provides the Software to U.S. government customers under its standard commercial license, which is embodied in its End User License Agreement (EULA), a copy of which can be found at http://www.keysight.com/find/sweula. The license set forth in the EULA represents the exclusive authority by which the U.S. government may use, modify, distribute, or disclose the Software. The EULA and the license set forth therein, does not require or permit, among other things, that Keysight: (1) Furnish technical information related to commercial computer software or commercial computer software documentation that is not customarily provided to the public; or (2) Relinquish to, or otherwise provide, the government rights in excess of these rights customarily provided to the public to use, modify, reproduce, release, perform, display, or disclose commercial computer software or commercial computer software documentation. No additional government requirements beyond those set forth in the EULA shall apply, except to the extent that those terms, rights, or licenses are explicitly required from all providers of commercial computer software pursuant to the FAR and the DFARS and are set forth specifically in writing elsewhere in the EULA. Keysight shall be under no obligation to update, revise or otherwise modify the Software. With respect to any technical data as defined by FAR 2.101, pursuant to FAR 12.211 and 27.404.2 and DFARS 227.7102, the U.S. government acquires no greater than Limited Rights as defined in FAR 27.401 or DFAR 227.7103-5 (c), as applicable in any technical data.

Warranty

THE MATERIAL CONTAINED IN THIS DOCUMENT IS PROVIDED "AS IS," AND IS SUBJECT TO BEING CHANGED, WITHOUT NOTICE, IN FUTURE EDITIONS. FURTHER, TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, KEYSIGHT DISCLAIMS ALL WARRANTIES, EI-THER EXPRESS OR IMPLIED, WITH REGARD TO THIS MANUAL AND ANY INFORMATION CONTAINED HEREIN, INCLUDING BUT NOT LIM-ITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FIT-NESS FOR A PARTICULAR PUR-POSE. KEYSIGHT SHALL NOT BE LIABLE FOR ERRORS OR FOR INCI-DENTAL OR CONSEQUENTIAL DAM-AGES IN CONNECTION WITH THE FURNISHING, USE, OR PERFOR-MANCE OF THIS DOCUMENT OR OF ANY INFORMATION CONTAINED HEREIN. SHOULD KEYSIGHT AND THE USER HAVE A SEPARATE WRITTEN AGREEMENT WITH WAR-RANTY TERMS COVERING THE MA-TERIAL IN THIS DOCUMENT THAT CONFLICT WITH THESE TERMS, THE WARRANTY TERMS IN THE SEPARATE AGREEMENT SHALL CONTROL.

Safety Information

CAUTION

A CAUTION notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a CAUTION notice until the indicated conditions are fully understood and met.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

Contents

1 Introduction

The Keysight Technologies N109X-series DCA-M sampling oscilloscopes combine the high-performance elements of both the 86100 oscilloscope mainframe acquisition system and the hardware of a receiver plug-in module. The N1010A FlexDCA application provides both the Graphical User Interface (GUI) as well as programming interface either on a PC or on an 86100D.

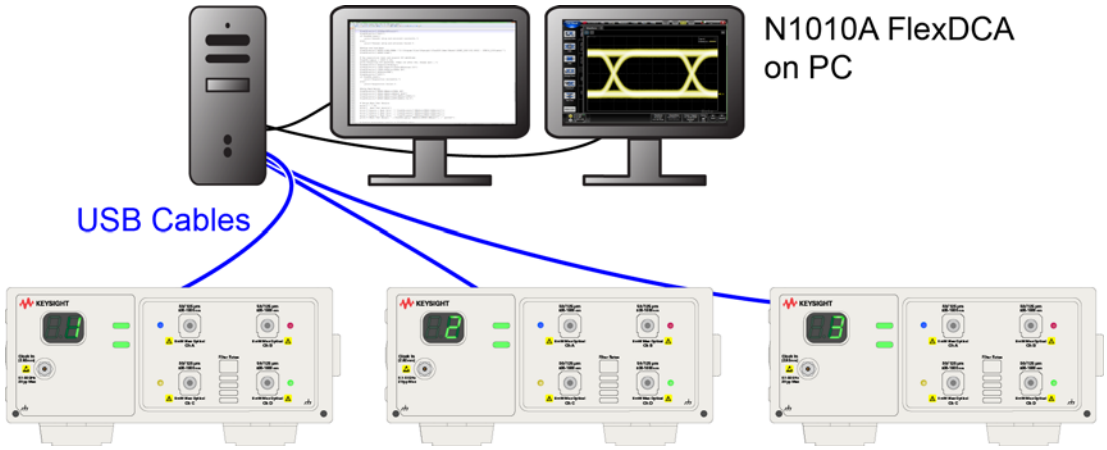


Figure 1 Setup with Multiple DCA-M Oscilloscopes

With a DCA-M, you can perform eye-mask tests, measure eye diagram parameters including extinction ratio, and basic oscilloscope mode measurements of pulses (single-valued waveforms rather than eye diagrams). This is limited to patterns less than 2 ns in duration. You can also perform optical transmitter compliance tests at a variety of standard data rates depending on options installed at the time of order. Installed filter rates must be selected at the time of order and cannot be changed.

Multiple DCA-M instruments can be connected to FlexDCA. But, delta time measurements between channels on separate DCA-M oscilloscopes *cannot* be compared because each N109X-series oscilloscope (and 86100D if connected) uses an independent trigger. Because the N109X-series oscilloscope has a significantly faster sampling rate when compared to an 86100D, measurement results can be returned up to 50% faster.

As shown in Table 1, a variety of optical and electrical channels are available depending on the model number. Table 2 on page 7 compares the inputs and outputs for the different models. The information in this table is presented to help you see the differences at a glance; it is not meant to be a replacement for the product specifications, which are located in FlexDCA's help system.

Table 1 DCA-M Channels Compared

	Channel Count		Minimum FlexDCA Firmware Version
Model	**Optical**	**Electrical**	
N1090A	1		A.05.00
N1090A-EEC	1	1	A.05.00
N1092A	1		A.05.40
N1092B	2		A.05.40
N1092C	1	2	A.05.50
N1092D	4		A.05.40
N1092E	2	2	A.05.50
N1094A		2	A.05.50
N1094B		4	A.05.50

Table 2 Inputs/Outputs Compared.
Refer to FlexDCA help system for complete, official specifications.

Feature	N1090-Series	N1092-Series	N1094-Series
Wavelength range (nominal)	750 nm to 1650 nm	830 nm to 1600 nm	–
Maximum input power (non-destruct)			
Optical (peak power)	5 mW (+7 dBm)	8 mW (+9 dBm)	–
Optical (average power)	0.5 mW (–3 dBm)	–	–
Electrical (peak power)	±2V (+16 dBm)	±2V (+16 dBm)	±2V (+16 dBm)
Electrical 3 dB BW			
Option 030	–	20 and 33 GHz	20 and 33 GHz
Option 035	–	20, 33, 40, and 50 GHz	20, 33, 40, and 50 GHz
N1090-series only	12.4 and 20 GHz	–	–
Data Input	1 to 10 Gb/s	20 to 28 Gb/s	20 to 33 Gb/s (Opt. 030) 20 to 50 Gb/s (Opt. 050)
Clock input			
Recommended	< 500 mVp-p to prevent crosstalk between trigger and channel input.		
Maximum non-destruct voltage		2.6 Vp-p	2.6 Vp-p
Frequency (full or sub-rate clocks)	455 MHz to 12 GHz, full or sub-rate clocks	500 MHz to 28.5 GHz, full or sub-rate clocks*	500 MHz to 28.5 GHz, full or sub-rate clocks[1]
Frequency (sub-rate clocks only)†	100 MHz to 455 MHz	100 MHz to 500 MHz‡	100 MHz to 500 MHz[3]
Trigger input (maximum)	2Vp-p**	–	–

*. Pattern lock mode on or off.

†. Provided that the data rate exceeds 455 Mb/s and the clock divide ratio is a power of two.

‡. Pattern lock off. Provided that the data rate exceeds 500 Mb/s and the clock divide ratio is a power of two.

**.< 500 mVp-p recommended.

NOTE To avoid damaging front-panel fiber-optic connectors, use proper connection techniques. Refer to "Cleaning the Fiber-Optic Connectors" on page 39.

NOTE The N1010A FlexDCA interface is free and can be downloaded at "www.keysight.com/find/flexdca_download".

N1090A

The N1090A sampling oscilloscope has one optical channel input and, if option EEC is installed, an electrical channel input. Refer "Options" on page 13 and "Accessories" on page 16 for tables of available options and accessories.

The N1090A can perform the following tasks:

- Eye-mask mote tests and measuring eye diagram parameters including extinction ratio.
- Basic oscilloscope mode measurements of pulses (single-valued waveforms rather than eye diagrams). This is limited to patterns less than 2 ns in duration.
- You can also perform optical transmitter compliance tests at a variety of standard data rates from 1.25 to 11.3 Gb/s.

The N1090A specifications are located in FlexDCA's help system.

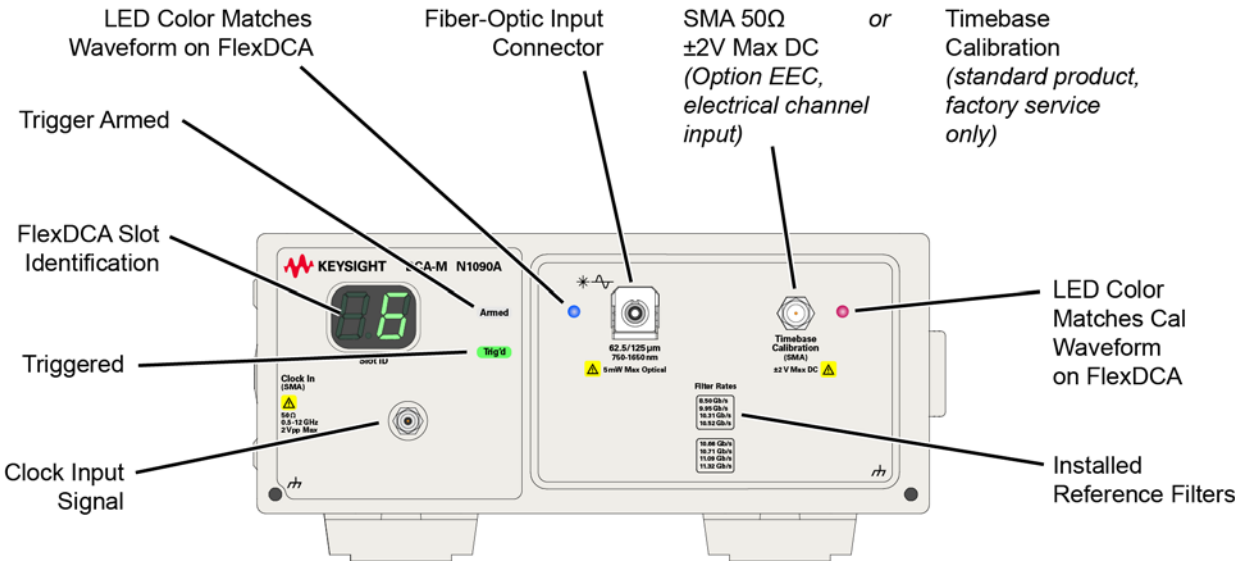


Figure 2 N1090A Front Panel

N1092A/B/C/D/E

The N1092A/B/C/D/E sampling oscilloscopes provide 1, 2, or 4 optical channels respectively. If option PLK is installed, you can lock data acquisition on a pattern trigger. Refer "Options" on page 13 and "Accessories" on page 16 for tables of available options and accessories.

The N1092A/B/D can perform the following tasks:

- Eye-mask mote tests and measuring eye diagram parameters including extinction ratio.
- Basic oscilloscope mode measurements of pulses (single-valued waveforms rather than eye diagrams). This is limited to patterns less than 2 ns in duration.
- You can also perform optical transmitter compliance tests at a variety of standard data rates from 1.25 to 11.3 Gb/s.
- Use pattern lock if option N1092A/B/D-PLK is installed.

The N1092A/B/D specifications are located in FlexDCA's help system.

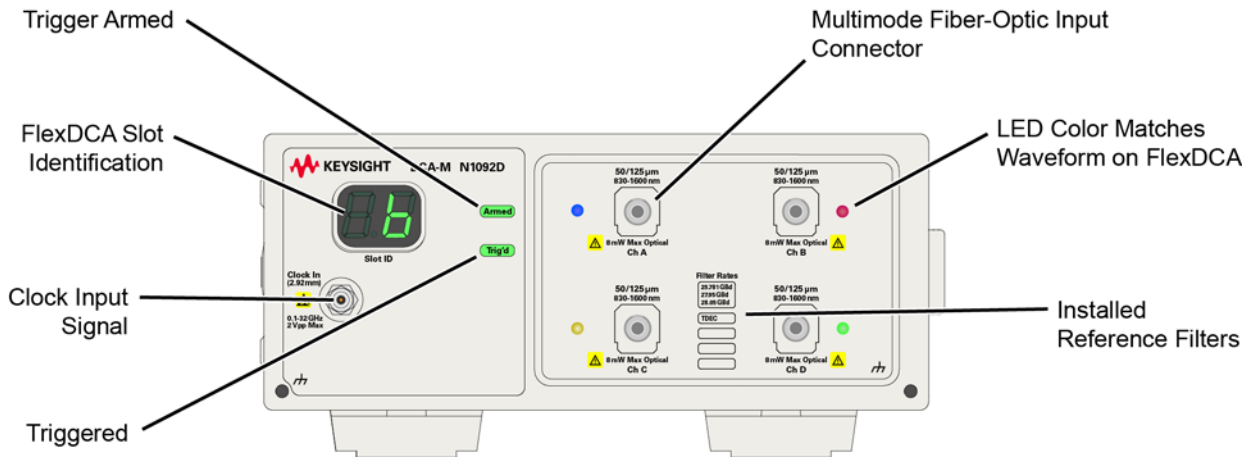


Figure 3 N1092A/B/D Front Panel

N1094A/B

The N1094A/B sampling oscilloscopes provide 2 or 4 electrical channels respectively. If option PLK is installed, you can lock data acquisition on a pattern trigger. Refer "Options" on page 13 and "Accessories" on page 16 for tables of available options and accessories.

The N1094A/B can perform the following tasks:

- Eye-mask mote tests and measuring eye diagram parameters including extinction ratio.
- Basic oscilloscope mode measurements of pulses (single-valued waveforms rather than eye diagrams). This is limited to patterns less than 2 ns in duration.
- You can also perform optical transmitter compliance tests at a variety of standard data rates from 1.25 to 11.3 Gb/s.
- Use pattern lock if option N1094A/B-PLK is installed.

The N1094A/B specifications are located in FlexDCA's help system.

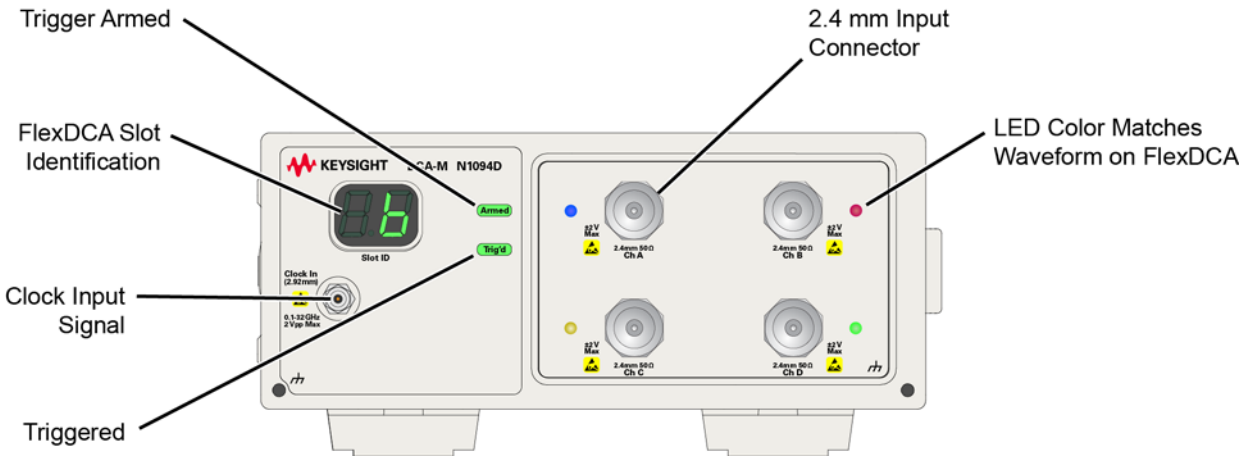


Figure 4 N1094A/B Front Panel

Front-Panel Indicator Lights

Table 3 DCA-M Front-Panel Indicator Lights

Indicator	Description
	Displays the FlexDCA extended module slot where the DCA-M is installed. The number indicates the slot, which in this picture would indicate slot 6 resulting in channel 6A, 6B, and so forth.
	**Trig'd** light light color indicates the state of the DCA-M: **Green**. The DCA-M is being triggered. As with the 86100D, this does not indicate that the trigger signal is synchronous with the channel input signal. In single acquisition, the light momentarily turns green each time **Single** is pressed. **Red**. The clock input signal is missing. **Off**. The channel is turned off *or* FlexDCA is in single/stop acquisition mode *or* the clock is missing.
	Trigger armed light. This light is red if FlexDCA is in single/stop acquisition mode.
	Channel indicator light. This light is next to the fiber-optic input connector. When on, the light indicates that the associated FlexDCA channel is turned on and that the waveform is displayed. The light is the same color as the displayed waveform. The Trig'd and Armed indicators do not light unless the channel is turned on.

Rear Panel

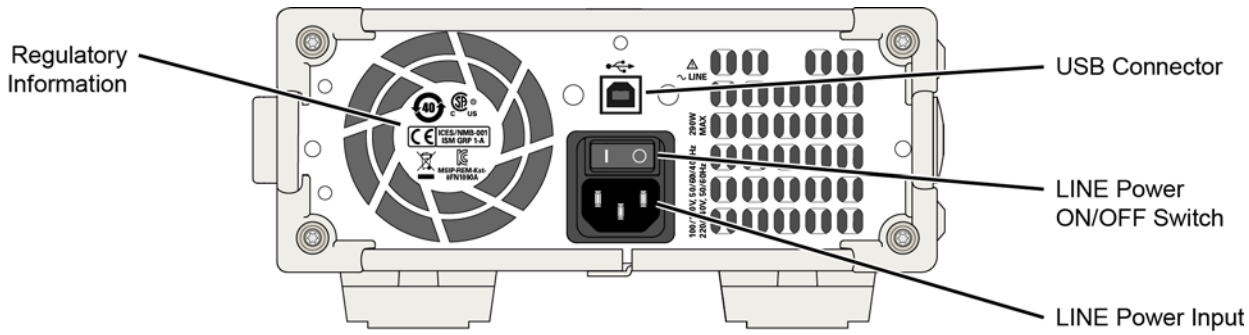


Figure 5 DCA-M Rear Panel

Options

Table 4 Options (Sheet 1 of 2)

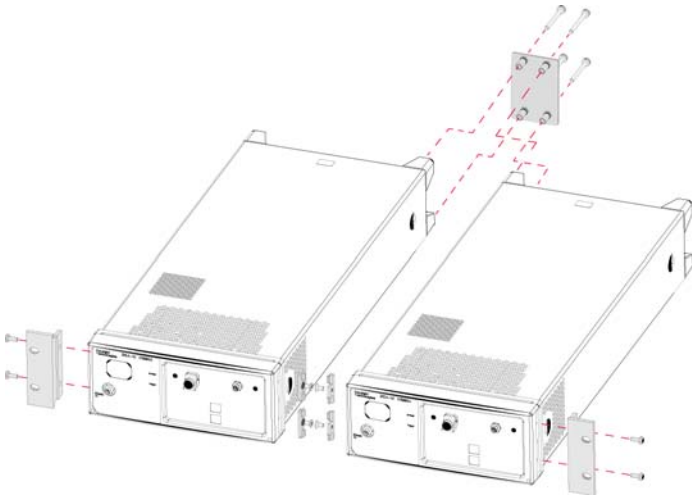
Option	Description	N1090A	N1092A/B/C/D/E	N1094A/B
Reference Filter Options (available only at time of order and cannot be changed)				
140	GPON, 1.244 Gb/s Gigabit Ethernet, 1.250Â Gb/s	♦		
160	OC-48/STM-16, 2.488 Gb/s 2 Gb Ethernet, 2.500 Gb/s	♦		
168	25.78 Gb/s TDEC		♦	
180	10Gb Ethernet LX-4, 3.125 Gb/s	♦		
200	8x Fibre Channel, 8.500 Gb/s OC-192/STM-64, 9.953 Gb/s 10Gb Ethernet, 10.3125 Gb/s 10x Fibre Channel, 10.51875 Gb/s OC-192/STM-64 FEC, 10.664 Gb/s OC-192/STM-64 FEC, 10.709 Gb/s 10Gb Ethernet FEC, 11.0957 Gb/s 10x Fibre Channel FEC, 11.317Gb/s	♦		
204	8x Fibre Channel, 8.500 Gb/s OC-192/STM-64, 9.953 Gb/s 10Gb Ethernet FEC, 10.3125 Gb/s 10x Fibre Channel, 10.51875 Gb/s	♦		
206	Additional 20.625 Gb/s		♦	
280	25 to 29 GBaud		♦	
Impulse response correction				
IRC	Impulse response correction to provide ideal channel response		♦	♦
Electrical Bandwidth				
030	33 GHz electrical bandwidth			♦
050	50 GHz electrical bandwidth			♦
Amplification				
30A	30 GHz amplified		♦	

Table 4 Options (Sheet 2 of 2)

Option	Description	N1090A	N1092A/B/C/D/E	N1094A/B
Pattern Lock				
PLK	Pattern lock trigger hardware. Requires option LOJ.		♦	♦
Timebase and Sampling Rate				
LOJ	Low jitter timebase		♦	♦
STB	Standard timebase		♦	♦
FS1	Fast sampling rate		♦	♦
Channel				
EEC	Add an electrical channel	♦		
Software				
200	Enhanced jitter analysis software, fixed perpetual license		♦	♦
201	Advanced waveform analysis software, fixed perpetual license		♦	♦
300	Advanced amplitude analysis/RIN/Q-factor, fixed perpetual license		♦	♦
401	Advanced eye analysis software, fixed perpetual license		♦	♦
500	Productivity package, fixed perpetual license		♦	♦
9FP	PAM-N analysis software, fixed perpetual license		♦	♦
ALL	Internal software options		♦	♦
EFP	FlexEye independent channel acquisition and control		♦	♦
SIM	InfiniiSim-DCA waveform transformation software, fixed perpetual license		♦	♦

Table 5 Available Rack Mount Options for all DCA-M Products

Option	Description
1CM	Rack-mount kit for a single DCA-M
1CN	Rack-mount kit for mounting any combination of two of the N109X-series DCA-M products.



Accessories

Table 6 Supplied Accessories

Item	Quantity					
	N1090A	N1092A	N1092B	N1092D	N1094A	N1094B
USB cable, USB-A plug to right-angle USB-B plug (3m long)	1	1	1	1	1	1
RF SMA 50-ohm cap (male)	2	1	1	1	1*	1†
FC fiber-optic dust cap	1	1	2	4		

*. 3 with option 030

†. 5 with option 030

Safety and Regulatory Information

This product has been designed and tested in accordance with accepted industry standards, and has been supplied in a safe condition. The documentation contains information and warnings that must be followed by the user to ensure safe operation and to maintain the product in a safe condition.

Only Keysight approved accessories shall be used.

This is a sensitive measurement apparatus by design and may have some performance loss when exposed to ambient continuous electromagnetic phenomenon.

The N1090A is in compliance with CAN/CSA-C22.2 No. 61010-1 and UL Std. 61010-1 and with IEC 61010-1.

The N1092A/B/D is in compliance with CAN/CSA-C22.2 No. 61010-1 and UL Std. 61010-1 and with IEC 61010-1.

The N1094A/B is in compliance with CAN/CSA-C22.2 No. 61010-1 and UL Std. 61010-1 and with IEC 61010-1.

COMPLIANCE WITH CANADIAN EMC REQUIREMENTS

This ISM device complies with Canadian ICES-001.

Cet appareil ISM est conforme a la norme NMB du Canada.

Acoustic statement: (European Machinery Directive)

Acoustic noise emission
LpA < 70 dB
Operator position
Normal operation mode per ISO 7779

WARNING **This is a Safety Protection Class I Product (provided with a protective earthing ground incorporated in the power cord). The mains plug shall only be inserted in a socket outlet provided with a protective earth contact. Any interruption of the protective conductor inside or outside of the product is likely to make the product dangerous. Intentional interruption is prohibited.**

WARNING **To prevent electrical shock, disconnect the Keysight Technologies Model N1090A, N1092A/B/C/D/E, N1094A/B from mains before cleaning. Use a dry cloth or one slightly dampened with water to clean the external case parts. Do not attempt to clean internally.**

WARNING **If this product is not used as specified, the protection provided by the equipment could be impaired. This product must be used in a normal condition (in which all means for protection are intact) only.**

WARNING **No operator serviceable parts inside. Refer servicing to qualified personnel. To prevent electrical shock do not remove covers.**

WARNING **For continued protection against fire hazard, replace fuses, and or circuit breakers only with same type and ratings. The use of other fuses, circuit breakers or materials is prohibited.**

Environmental Specifications

The following table lists the environmental specifications. Performance specifications for the DCA-M are documented in FlexDCA's help system.

Table 7 Environmental Specifications

Item	Description
Use	For indoor use only
Temperature	
Operating	10°C to +40°C (50°F to +104°F)
Non-operating	–40°C to +70°C (–40°F to +158°F)
Altitude	Up to 4,600 meters (15,000 ft)
Maximum relative humidity	80% for temperatures up to 31°C decreasing linearly to 50% relative humidity at 40°C
Weight (Characteristic)	
N1090A	6.20 kg (13.68 lb)
N1092A	6.071 kg (13.384 lb)
N1092B	6.159 kg (13.578 lb)
N1092C	6.1 kg (13.5 lb)
N1092D	6.344 kg (13.986 lb)
N1092E	6.1 kg (13.5 lb)
N1094A	6.0 kg (13.4 lb)
N1094B	6.1 kg (13.6 lb)
Without front connectors and rear feet	88.26 mm H x 207.40 mm W x 485 mm D (3.48 inch x 8.17 inch x 19.01 inch)
With front connectors and rear feet	103.31 mm H x 219.56 mm W x 517.80 mm D (4.07 inch x 8.64 inch x 20.39 inch)
With front cover and rear feet	110.18 mm H x 219.56 mm W x 550.71 mm D (4.34 inch x 8.64 inch x 21.68 inch)

Instrument Markings

Table 8 Instrument Markings

Marking	Description
	The AC symbol is used to indicate the required nature of the line module input power.
	This symbol indicates that the power line switch is ON.
	This symbol indicates that the power line switch is in the OFF position.
	The instruction documentation symbol. The product is marked with this symbol when it is necessary for the user to refer to the instruction in the documentation
	The chassis ground symbol. The chassis ground symbol is used to indicate a chassis connection.
	The CE indicates of the European Community. ISM GRP 1-A: This is a symbol of an Industrial Scientific and Medical Group 1 Class A product (CISPR 11, Clause 4) ICES/NMB-001: Cet appareil ISM est conforme a la norme NMB du Canada. This is a marking to indicate product compliance with the Canadian Interference-Causing Equipment Standard (ICES-001)
	The CSA indicates Canadian Standards Association.
	The RCM mark indicates the Australian Communications and Media Authority.
	This symbol indicates separate collection for electrical and electronic equipment, mandated under EU law as of August 13, 2005. All electric and electronic equipment are required to be separated from normal waste for disposal (Reference WEEE Directive, 2002/96/EC).
	This symbol indicates the Environmental Protection Use Period (EPUP) for the product's toxic substances for the China RoHS requirements.
	Packaging recycling symbol.
	South Korean Class A EMC Declaration A 급 기기 (업무용 방송통신기자재) This equipment is Class A suitable for professional use and is for use in electromagnetic environments outside of the home. 이 기기는 업무용 (A 급) 전자파적합기기로서 판 매자 또는 사용자는 이 점을 주 의하시기 바라 며 , 가정외의 지역에서 사용하는 것을 목적으 로 합니다 .

Returning the instrument for service

If the probe is found to be defective we recommend sending it to an authorized service center for all repair and calibration needs. Perform the following steps before shipping the probe back to Keysight Technologies for service.

1 Contact your nearest Keysight sales office for information on obtaining an RMA number and return address.

2 Write the following information on a tag and attach it to the malfunctioning equipment.

- Name and address of owner
- Product model number (for example, N1092D)
- Product Serial Number (for example, MYXXXXXXXX)
- Description of failure or service required

NOTE Use original packaging or comparable.

3 Protect the probe by wrapping in plastic or heavy paper.

4 Pack the probe in the original carrying case or if not available use bubble wrap or packing peanuts.

5 Place securely in sealed shipping container and mark container as "FRAGILE".

NOTE If any correspondence is required, refer to the product by serial number and model number.

Contacting Keysight Technologies

For technical assistance, contact your local Keysight Call Center.

- In the Americas, call 1 (800) 829-4444
- In other regions, visit http://www.keysight.com/find/assist

Before returning an instrument for service, you must first call the Call Center at 1 (800) 829-4444.

2 Installing

PC Requirements

A PC that is controlling the N109XA-series oscilloscope, or any other DCA-M, must meet the following requirements.

Single channel setup

- Intel I3 processor or better
- 4 GB memory
- Windows 7 (32 or 64 bit)

Parallel test setup (multiple DCA-Ms or multiple channels)

- Intel I5 or better
- 8 GB memory
- Windows 7 (64 bit)

The communication API between your system controller and the PC is SCPI over LAN, either VXI-11 or HiSlip. If NI-VISA or IO Libraries are used to communicate with GPIB instruments, the switch to SCPI/LAN is very simple. It is important to note that there is no need to do any USB programming. This is all handled by the FlexDCA interface.

Select a Controller Setup

Figure 6, Figure 7, and Figure 8 illustrate three different setups for controlling a DCA-M. The USB cable is provided with the DCA-M. The N1010A FlexDCA application (*PC not provided by Keysight)* directly controls the DCA-M.

Setup 1. Using a PC

Figure 6 shows a DCA-M connected to a PC via USB. FlexDCA is running on the PC. This is a typical setup.

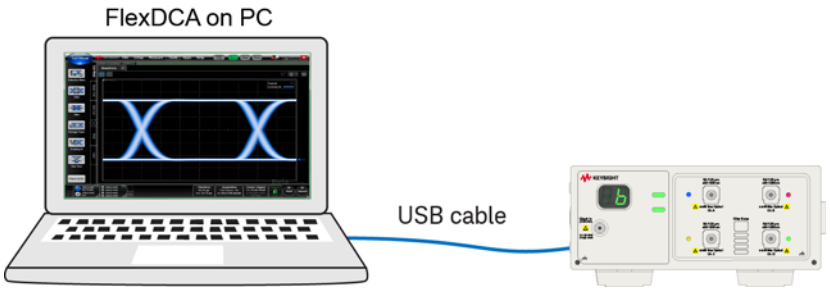


Figure 6 DCA-M connected to a PC

Setup 2. Using a controller connected to a PC test system

Figure 7 shows a DCA-M connected to a PC via USB. FlexDCA on this PC is being controlled over LAN by the primary test system's controller.

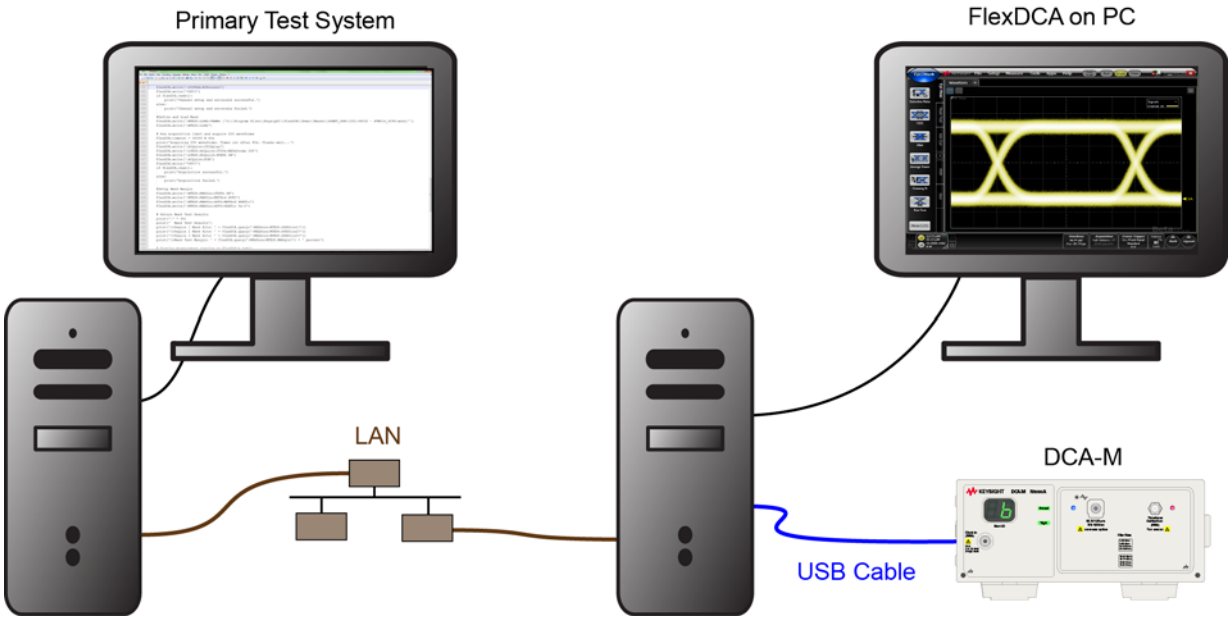


Figure 7 DCA-M connected to an a PC (with FlexDCA) controlled by a test system PC

Setup 3. Using an 86100D

Figure 8 shows a DCA-M connected to an 86100D via USB. FlexDCA is running on the 86100D.

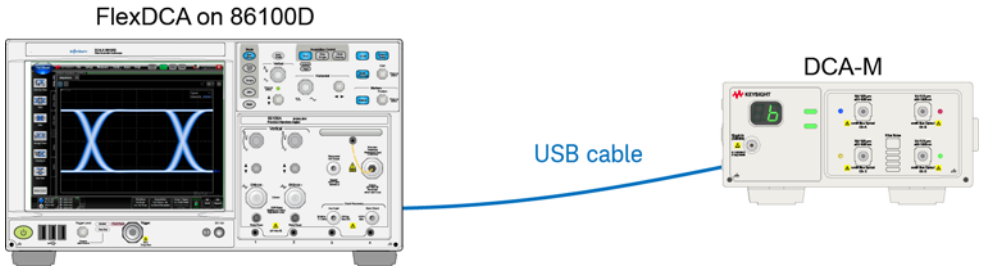


Figure 8 DCA-M connected to an 86100D

Flex-on-Flex setup does not work

Figure 9 shows a setup that *does not* work. This is known as a “Flex-on-Flex” connection, because, N1010A FlexDCA on the PC is connected to and controlling FlexDCA on the 86100D. A Flex-on-Flex setup works great for standard 86100D modules, but not for DCA-M instruments. The DCA-M in this picture *cannot be* identified, accessed, or controlled from FlexDCA on the PC. The solution is to connect the DCA-M directly to the PC.

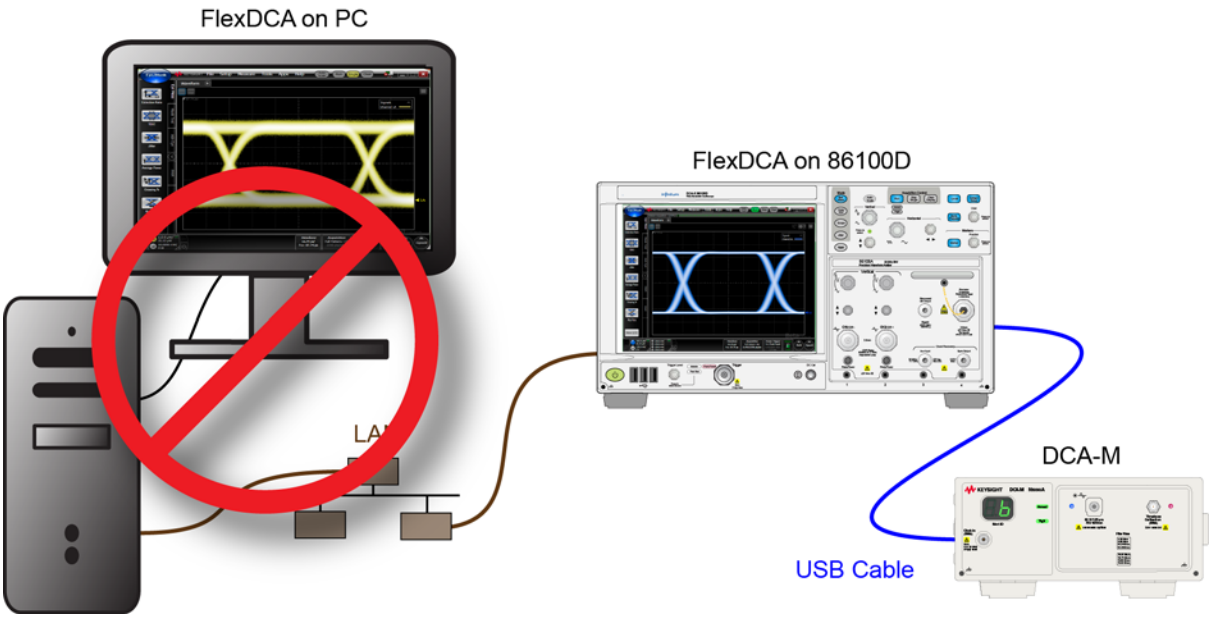


Figure 9 DCA-M cannot be “seen” with a Flex-on-Flex connection

To Install the DCA-M

Before attempting the following procedure, read "Safety and Regulatory Information" on page 17.

CAUTION This product is designed for use in INSTALLATION CATEGORY II and POLLUTION DEGREE 2 environment.

CAUTION CAUTION, VENTILATION REQUIREMENTS: When installing the instrument(s) into a cabinet consideration shall be given to the convection flow into and out of the cabinet. Consideration shall also be given to the individual instruments to avoid having the heated discharge of one instrument, now becoming the cooling intake air for another instrument. Another area of concern is verification that the maximum ambient operating temperature of the instrument(s) is not exceeded by cabinet installation. Keysight recommends forced air convection whenever an instrument(s) are installed in a cabinet and further recommends that the maximum operating temperature of the cabinet be reduced 10°C from the lowest, of the maximum operating temperature of a single instrument. If there are any concerns or special requirements a Keysight Field Engineer should be consulted to assure instrument(s) temperature compliance and performance.

Procedure

1 Confirm that your site satisfies the LINE power requirements shown in Table 9.

Table 9 LINE Power Specifications

Item	Description
Nominal voltage and/or range	100/120 VAC, 50/60/400 Hz
	220/240 VAC, 50/60 Hz
Power in Watts	290 Watts Maximum
The instruments can operate with mains supply voltage fluctuations up to ± 10% of the nominal voltage.	

WARNING **The Mains wiring and connectors shall be compatible with the connector used in the premise electrical system. Failure, to ensure adequate earth grounding by not using the correct components may cause product damage, and serious injury.**

WARNING **Use Keysight supplied power cord or one with same or better electrical rating.**

CAUTION This instrument has auto-ranging line voltage input, be sure the supply voltage is within the specified range.

NOTE Install the instrument so that the detachable power cord is readily identifiable and is easily reached by the operator. The detachable power cord is the instrument disconnecting device. It disconnects the mains circuits from the mains supply before other parts of the instrument.

NOTE The main power cord can be used as the system disconnecting device. It disconnects the mains circuits from the mains supply.

2 Connect the supplied LINE power cord.

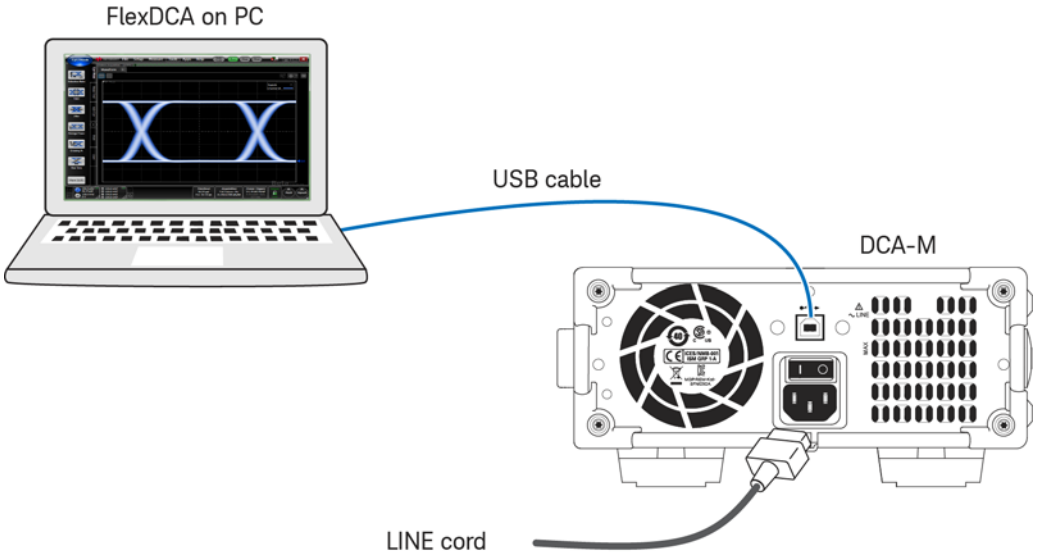


Figure 10 DCA-M Connected to PC

3 If you will be controlling the DCA-M from a PC, install FlexDCA on the PC and start FlexDCA.

4 Use the supplied USB cable to connect the DCA-M to the PC or 86100D, depending on the setup. See Figure 10. If connecting to an 86100D, be sure to connect the USB cable to one of the 86100D's *rear-panel* USB ports.

5 Turn on the DCA-M. A device driver that is installed with FlexDCA will automatically detect the DCA-M. The DCA-M is installed in FlexDCA's first available virtual slot. The slot number is shown on the DCA-M's front-panel LED readout. Available FlexDCA slots depend on the installation:

- If the DCA-M is connected to a PC *and* an 86100D is *not* connected, slots 1 through 8 are available.
- I the DCA-M is connected to an 86100D, slots 5 through 8 are available.

6 Locate the slot button for your installed N109X-series in FlexDCA's slot button tray as shown in Figure 11. Click the gear button is you want to re-assign the DCA-M to a different slot. This can also be accomplished remotely as shown in "Example 2. Re-assign a DCA-M to a different slot" on page 86.

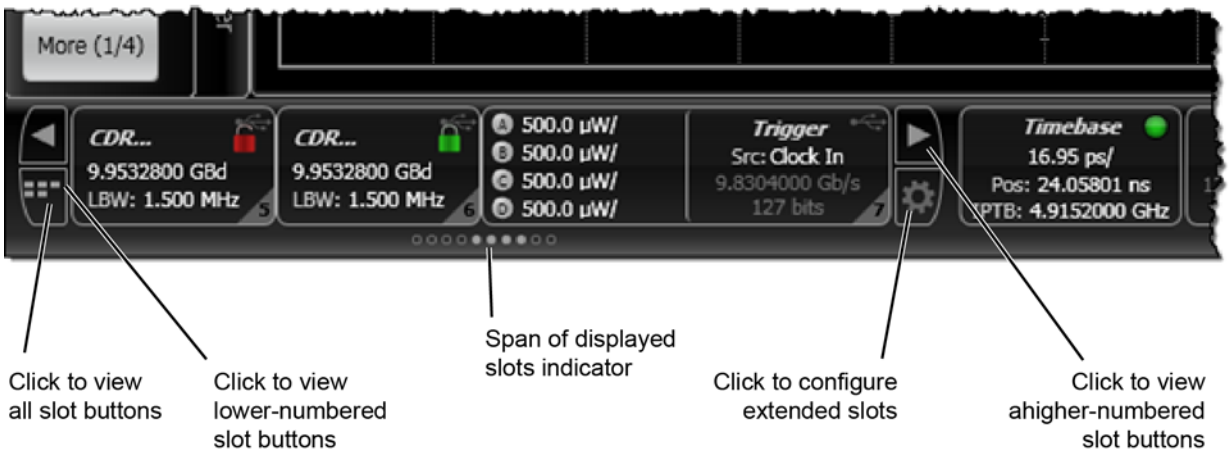


Figure 11 Slot buttons at bottom of FlexDCA application

7 If the DCA-M has an optical input, connect an optical signal to the DCA-M's front-panel fiber-optic connector.

8 Connect a clock signal to the DCA-M's front-panel **Clock In** SMA connector.

As with the 86100D, use a reference clock that is synchronous with the input signal being measured. The clock input range is 100 MHz to 32 GHz. The DCA-M's front-panel **Trig'd** light should turn from orange to green indicating that the signal is locked. If the oscilloscope's acquisition is in stopped or single mode, the **Trig'd** indicator will not be lit.

9 In FlexDCA's Trigger Setup dialog box, select the General Trigger Setup tab. In the Source field, select Front Panel.

NOTE **Although the amplitude of the clock can be 2 Vpp maximum, for best results keep the amplitude below 500 mVpp.**

10 To learn more about using the DCA-M, read Chapter 3, “Using" and refer to FlexDCA’s help system.

NOTE To avoid damaging a DCA-M's front-panel fiber-optic connector, use proper connection techniques. Refer to “Cleaning the Fiber-Optic Connectors" on page 39.

3 Using

A N109X-series oscilloscope is controlled using the FlexDCA user interface that is running on a PC or an 86100D. Because, this is the 86100D's graphical user interface, 86100D users will be familiar with operating an N109X-series oscilloscope.

Multiple DCA-Ms can be connected to FlexDCA, but delta-time measurements between two channels *cannot* be compared because each DCA-M (and 86100D if connected) uses an independent trigger.

NOTE **The N1010A FlexDCA interface is free and can be downloaded at "www.keysight.com/find/flexdca_download".**

Remote programs previously developed using the 86100 FlexDCA interface can be leveraged directly to control an N109X-series oscilloscope. Use FlexDCA SCPI programming tools to simplify conversion of legacy 86100-based automation to FlexDCA compatible code, as noted in Chapter 4, "Programming".

To Setup the Oscilloscope

This example shows an N1092D-series sampling oscilloscope. You can substitute any N109X-series oscilloscope. The input **Clock In** signal must be a synchronous clock or sub-rate clock. The N109X-series oscilloscope will trigger with any clock signal between 100 MHz and 32 GHz.

The settings made in this procedure can also be made using a program as shown in the Python example, "Example 3. Perform eye measurement with N109X" on page 65.

In this procedure, the following assumptions are made for simplicity. You can change these parameters to any acceptable values.

Data signal: . . . . . 10.3125 GBd
N1092D slot: 5

1 In FlexDCA, click **Setup** > **Default Setup**.
2 Connect the N109X-oscilloscope as shown in Figure 12. If connecting to an 86100D, be sure to connect the USB cable to an 86100D *rear-panel* USB port.

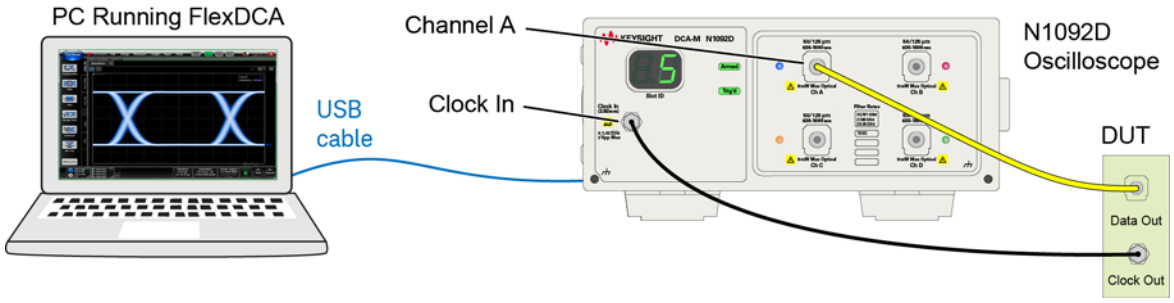


Figure 12 Typical Connection Setup

3 In FlexDCA's signals palette, turn all channels off except for channel 5A.
4 Click **Run**.
5 Click **Setup** > **Mode** > **Eye/Mask**.
6 Click **Auto Scale**.
7 The N109X-series oscilloscope's front-panel **Trig'd** light should be green. If not, confirm that the amplitude of the signal at channel 5A is not too low due to the "pass through" loss in the N1077A. Consult the specifications listed in the oscilloscope user's guide.

Configure pattern lock (optional)

The N109X-series oscilloscope must have option PLK.

8 Click **Setup** > **Trigger Setup** and select the Pattern Lock tab. In the **Data Rate** field, turn off **Auto Detect** and select 10.3125 GBd.

9 In the General Setup tab, click the **Pattern Lock** button.

To Setup the Oscilloscope with Clock Recovery

This example shows an N1077A-SMS that is connected to an N1092D-series sampling oscilloscope. You can substitute any N109X-series oscilloscope. If you are using an N1077A-SXT, you will need an external optical splitter. If you are using an N1076A, you will connect your electrical data signal to the N1076A's front-panel **Data In +** port.

The input **Clock In** signal must be a synchronous clock or sub-rate clock. The N109X-series oscilloscope will trigger with any clock signal within the range specified in the FlexDCA help system.

To learn more about using the N1076/7A clock recovery DCA-M, download the N1076/7A user's guide from *www.keysight.com*.

The settings made in this procedure can also be made using a program as shown in the Python example, "Example 4. Perform eye measurement with N109X & N1076/7A" on page 68.

In this procedure, the following assumptions are made for simplicity. You can change these parameters to any acceptable values.

Data signal: . . . . . 10.3125 GBd
N1092D slot: 5
N1076/7A slot:. 6

1 In FlexDCA, click **Setup** > **Default Setup**.

2 Connect the N1076/7A as shown in Figure 13. If connecting to an 86100D, be sure to connect the USB cable to an 86100D *rear-panel* USB port.

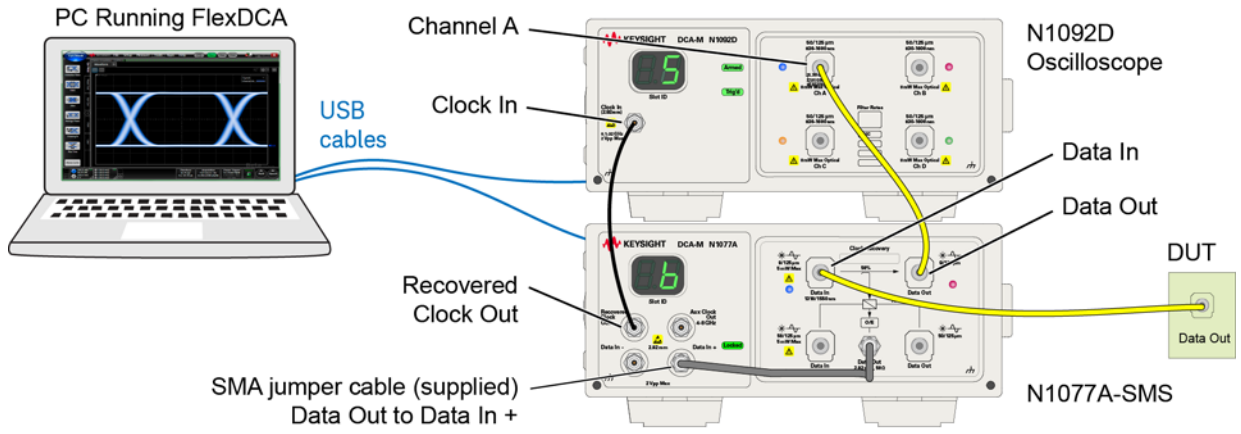


Figure 13 Typical Connection Setup

Configure clock recovery

3 In FlexDCA, click **Setup** > **Trigger Setup** and select the General Trigger Setup tab. Confirm that the trigger Source is set to **Front Panel**.

4 Click **Setup** > **Modules** > **(Slot 6): Clock Recovery**. The dialog box shown Figure 14 on page 35 is displayed.

5 For the **Input Source**, select **Electrical Data +**, which is used for single-ended or optical signal.

6 In the **Data Rate** field, enter 10.3125 GBd.

7 In the **Clock Recovery** field, click the **Lock** button to lock clock recovery. The N1076/7A front-panel locked light should be green. If not, confirm the input signal's amplitude and baud rate against the values described in Table 2 on page 7.

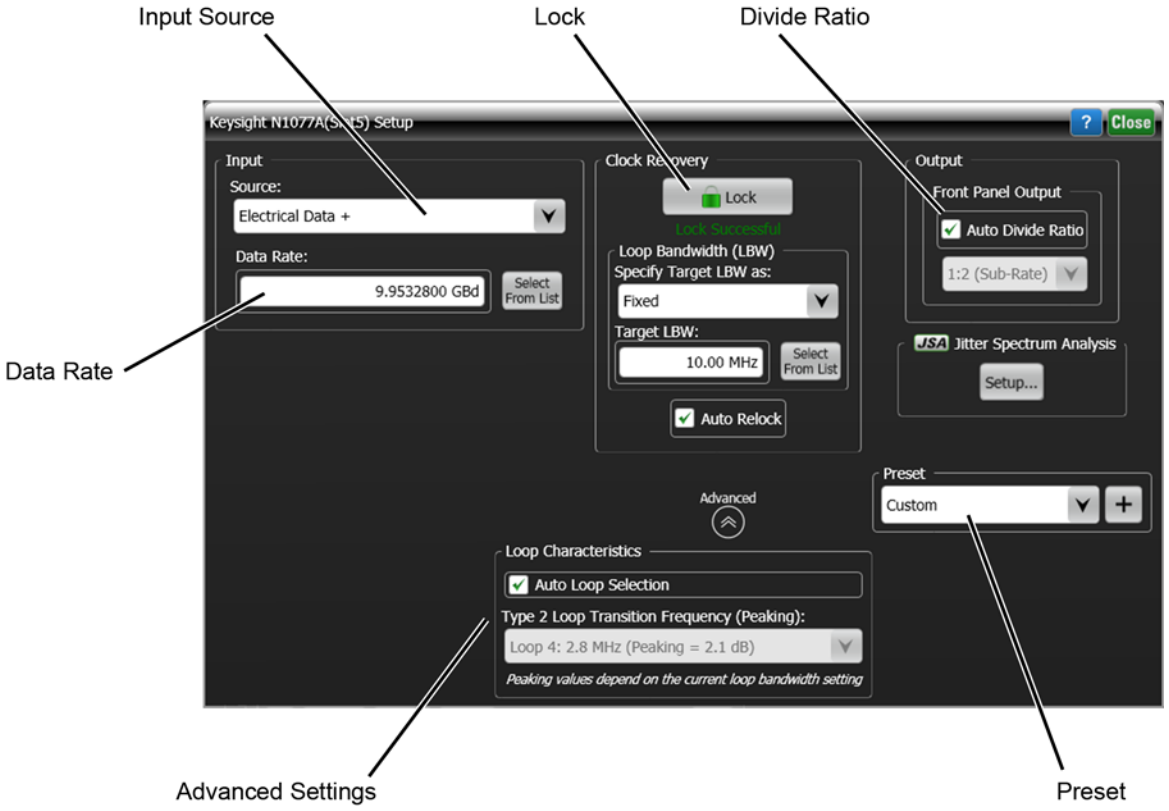


Figure 14 Clock Recovery Setup Dialog Box

Configure N109X-series oscilloscope

8 In FlexDCA's signals palette, turn all channels off except for channel 5A.

9 Click **Run**.

10 Click **Setup** > **Mode** > **Eye/Mask**.

11 Click **Auto Scale**.

12 The N109X-series oscilloscope's front-panel Trig'd light should be green. If not, confirm that the amplitude of the signal at channel 5A is not too low due to the "pass through" loss in the N1077A. Consult the specifications listed in the oscilloscope user's guide.

Configure pattern lock (optional)

The N109X-series oscilloscope must have option PLK.

13 Click **Setup** > **Trigger Setup** and select the Pattern Lock tab. In the **Data Rate** field, turn off **Auto Detect** and select 10.3125 GBd.

14 In the General Setup tab, click the **Pattern Lock** button.

Reference Clock

Like the 86100D, for proper display of the input signal the input **Clock In** signal must be a synchronous clock or sub-rate clock. The N109X-series oscilloscope will trigger with any clock signal between 100 MHz and 32 GHz. The clock input range for the N1090A is 500 MHz to 12 GHz, and the N1092 is 0.455 to 28.5 GHz. A timebase calibration, previously performed at service centers, can be performed by users.

Unlike an 86100D, the trigger setting for the N109X-series can only be selected to be Front Panel or Free Run.

Table 10 Valid Clock Input

Feature	N1090-Series	N1092-Series	N1094-Series
Recommended	< 500 mVp-p to prevent crosstalk between trigger and channel input		
Maximum non-destruct voltage		2.6 Vp-p	2.6 Vp-p
Frequency (full or sub-rate clocks)	455 MHz to 12 GHz, full or sub-rate clocks	500 MHz to 28.5 GHz, full or sub-rate clocks*	500 MHz to 28.5 GHz, full or sub-rate clocks[1]
Frequency (sub-rate clocks only)†	100 MHz to 455 MHz	100 MHz to 500 MHz‡	100 MHz to 500 MHz[3]

* Pattern lock mode on or off.

† Provided that the data rate exceeds 455 Mb/s and the clock divide ratio is a power of two.

‡ Pattern lock off. Provided that the data rate exceeds 500 Mb/s and the clock divide ratio is a power of two.

Pattern Lock

On N1092-series and N1094-series oscilloscopes, option PLK adds pattern lock capability. The following measurement features require pattern lock:

- Option 200 (Advanced Jitter Analysis)
- Option 201 (Advanced Waveform Analysis)
- Option 300 (Advanced Amplitude Analysis/RIN)
- Option SIM (InfiniiSim-DCA Waveform Transformation Toolset).

Reference Filters

The rates of all installed reference filters are listed on the front panel. These rates correspond to internal, calibrated reference filters that are selected as an instrument option at the time of order. These reference filters allow for optical

transmitter compliance testing at the listed filter rates. The available options cover a variety of standard data rates from 1.25 to 11.3 Gb/s as shown in Table 4 on page 13.

Aligning Waveforms

Because a DCA-M's acquisition system is independent of the 86100D, if you are using multiple DCA-Ms or a DCA-M with 86100D receiver modules, the input waveforms may not align after an Auto Scale.

The waveforms will align If option 500, Productivity Package Software, is installed on the 86100D or DCA-M *and* Rapid Eye is turned on. Rapid Eye is enabled from the Acquisition Setup dialog box's Waveform tab.

If option 500 is *not* installed, you can manually add software delay to channels to align them. To align the eyes, open the channel's Setup dialog box, click **Advanced**, and enter the Software Delay.

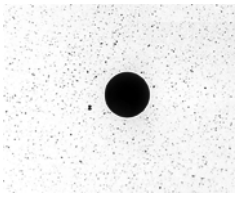
Cleaning the Fiber-Optic Connectors

Treat all fiber-optic connectors like the high-quality lens of an expensive camera. Damage to the connectors on calibration and verification devices, test ports, cables, and other devices can:

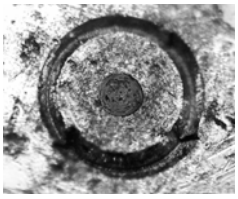
- Degrade measurement accuracy and repeatability and
- Cause expensive damage to instruments.

Because fiber-optic connectors are susceptible to damage that is not immediately obvious to the naked eye, it is very easy to make bad measurements without being aware of a connector problem. Learning about proper handling and cleaning techniques will help you to avoid any degradation in connector performance. With glass-to-glass interfaces, any damage of the ferrule or end of the fiber, any stray particles, or finger oil can have a significant effect on connector performance.

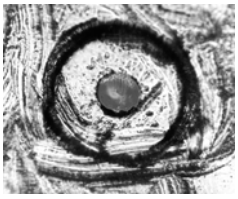
This picture shows the end of a clean, problem-free fiber-optic connector. The dark center circle is the fiber's 125 µm core and cladding which carries the light. The surrounding area is the soft nickel-silver ferrule.



This picture shows a fiber end that is dirty from neglect or improper cleaning. Loose particles or oils are smeared and ground into the end of the fiber causing light scattering and poor reflection. Not only is the precision polish lost, but this action can also grind off the glass face and destroy the connector.



This picture shows physical damage to the glass fiber end caused by either repeated connections made without removing loose particles from the fiber end or by using improper cleaning tools. This damage can be severe enough to transfer the damage from the connector end to a good connector that comes in contact with it.



The cure for these problems is disciplined connector care. Visual inspection of fiber ends can be helpful. Contamination or imperfections on the cable end face can be detected as well as cracks or chips in the fiber itself. Use a microscope (100X to 200X magnification) to inspect the entire end face for contamination, raised metal, dents in the metal, and any other imperfections. Visible imperfections not touching the fiber core may not affect performance, unless the imperfections keep the fibers from contacting.

WARNING **Always remove both ends of fiber-optic cables from any instrument, system, or device before visually inspecting the fiber ends. Disable all optical sources before disconnecting fiber-optic cables. Failure to do so may result in permanent injury to your eyes.**

CAUTION Improper connector care, cleaning, or use of mismatched cable connectors can invalidate the published specifications and damage connectors. Clean all cables before applying to any connector. Repair of damaged connectors due to improper use is not covered under warranty.

To clean a front-panel fiber-optic input connector

Use caution as fiber-optic end surfaces can easily be damaged due to improper cleaning techniques. Repairs can be expensive. To access and clean the fiber-optic end surface:

1 Lift the receptacle latch as shown in the following picture.

2 Carefully pull off the receptacle without touching the ferrule or fiber end.

3 Clean the removed receptacle.

4 To clean the fiber-optic end, use a professional fiber-optic cleaning product. Many products are available and are easily located via an internet search on "*fiber optic cleaning products*". You can purchase tools designed specifically for the type of fiber-optic connector or cable that you are using.

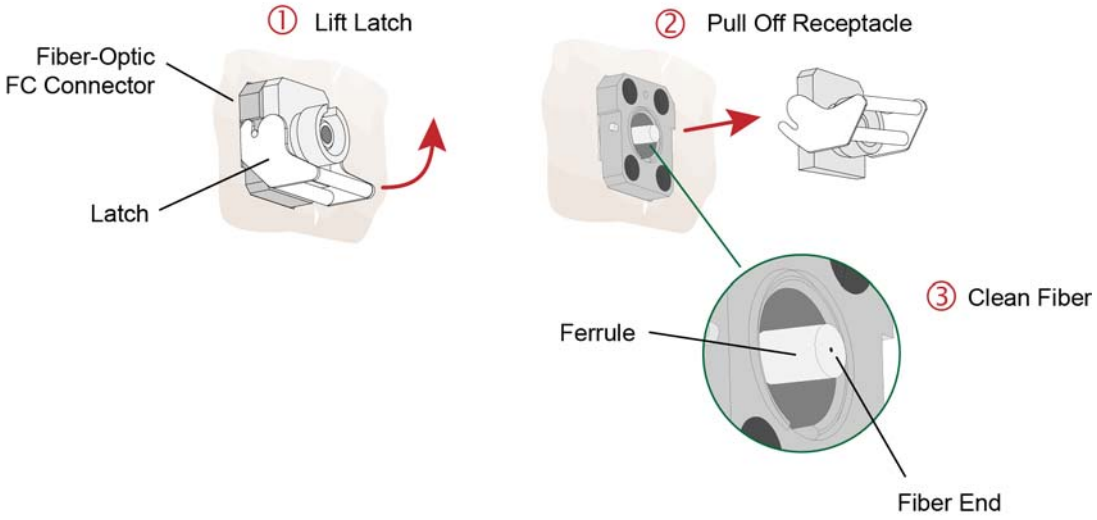


Figure 15 Removing the Receptacle to Expose the Fiber End

NOTE To avoid damaging a DCA-M's front-panel fiber-optic connector, use proper connection techniques.

4 Programming

The N109X-series oscilloscopes are remotely controlled using the SCPI commands that are documented in FlexDCA's help system. The commands are sent to N1010A FlexDCA (on a PC) or FlexDCA (on the 86100D). The most common setup is shown in Figure 16 on page 42 which shows the program running on a PC which controls one or more N109X-series oscilloscopes via N1010A FlexDCA.

This chapter explains the minor differences between programming N109X-series and 86100D sampling oscilloscope and provides a few simple examples. More example can be found in FlexDCA's help system.

To help in developing programs, use the following FlexDCA SCPI programming tools:

- SCPI Recorder (page 48)
- Interactive SCPI Command Tree (page 50)

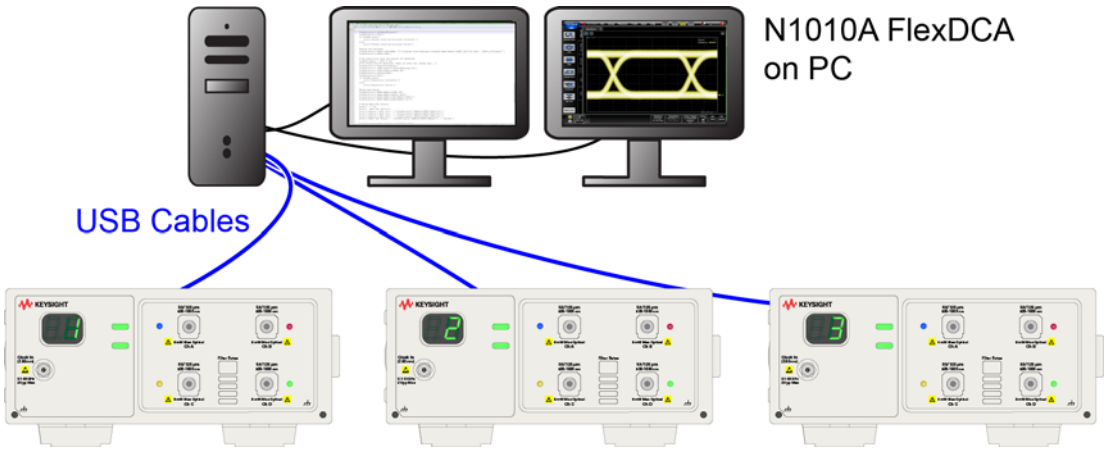


Figure 16 Setup for Remote Control of the DCA-M on PC

FlexDCA commands t work the same regardless if they control an N109X-series or 86100D. For example, the command:

```
:CHANnel2A:YSCale 1.0E-1
```

changes the channel's vertical scale on a N109X-series just like on an 86100D. Refer to FlexDCA's help system for information comprehensive information on remote control. The help includes a topic for every command.

Unique DCA-M Subsystems

:EMODules Subsystem

FlexDCA's **:EMODules** (extended modules) SCPI command subsystem is unique to DCA-M instruments such as N109X-series oscilloscopes and N1076/7A clock recovery. Use this subsystem to assign DCA-M oscilloscopes to FlexDCA slots (1 through 8). On an 86100D, only slots 5 through 8 are available.

Table 11 :EMODules Commands

Command	Description
:EMODules:DCAM:DEVice?	Queries the identification string of an DCA-M extended module.
:EMODules:SLOT:CMEThod?	Queries the connection method for the module in the slot. USB is returned if a DCA-M is connected and NONE if one is not connected.
:EMODules:SLOT:CONNect	Sets the communication connection with a DCA-M.
:EMODules:SLOT:DISConnect	Disconnects an extended module, simulated module, or mainframe from the indicated slot.
:EMODules:SLOT:SELection	Assigns an extended module, simulated module, or mainframe to a slot.
:EMODules:SLOT:STATe	Queries the state of the extended module connection.

:SLOT Subsystem for N109X-Series Oscilloscopes

FlexDCA's **:SLOT** SCPI command subsystems is unique to DCA-M instruments. Use this subsystem to control the trigger setting of a DCA-M. Refer to Table 12 on page 44.

N109X-series oscilloscopes with option PLK installed have the ability to pattern lock. Because these oscilloscopes use different internal hardware than the 86100D, the **:SLOT** subsystem is used rather than the **:TRIGger** subsystem for pattern lock.

Table 12 :SLOT Commands for N109X-Series Oscilloscopes

Command	Description
:SLOT:TRIGger:MODe?	Queries the trigger mode, which is always clock (CLOC) and cannot be set or changed (:TRIGger:MODe command). You can query CLOC (clock) with either the :TRIGger:MODe? or :SLOT:TRIGger:MODe? queries.
:SLOT:TRIGger:MRATe?	Queries the measured clock rate at the Clock In input of the DCA-M extended module.
:SLOT:TRIGger:SOURce?	The DCA-M oscilloscope's trigger source can either be front panel or free run as set by this command's FPANel or FRUN arguments. Do not use the CHANnel or SLOT arguments.
Commands of for Oscilloscopes with Option PLK (Pattern Lock)	
:SLOT:TRIGger:BRATe	Enters or queries the bit-rate of the trigger source.
:SLOT:TRIGger:BRATe:AUTodetect	Enables or disables automatic bit-rate detection.
:SLOT:TRIGger:DCDRatio	When using pattern lock, enters or queries the trigger-divide ratio.
:SLOT:TRIGger:DCDRatio:AUTodetect	Enables or disables automatic trigger-divide-ratio detection.
:SLOT:TRIGger:MRATe?	Queries the measured clock rate at the DCA-M's front-panel Clock In input connector.
:SLOT:TRIGger:PLANgth	Enters the length of the pattern in bits.
:SLOT:TRIGger:PLANgth:AUTodetect	Queries the measured clock rate at the Clock In input of the DCA-M extended module.
:SLOT:TRIGger:TRACking	Turns on pattern lock tracking where pattern lock settings are the same (linked) for all DCA-M oscilloscopes and the 86100D. The clock and data signals should be the same rate and pattern length.

Aligning Waveforms

Because a DCA-M's acquisition system is independent of the 86100D, if your are using multiple DCA-Ms or a DCA-M with 86100D receiver modules, the input waveforms may not align after an Auto Scale (**:SYSTem:AUToscale)**. To enable **:SYSTem:AUToscale** to align waveforms:

- If option 500, Productivity Package Software, is installed on the 86100D or DCA-M, turn Rapid Eye is turned on, and the waveforms will align. Option 500 is the most convenient way to align the waveforms and maintains very fast throughput. To enable Rapid Eye and align waveforms, use these commands.

```
:ACQuire:REYE:ALIGn ON
:ACQuire:REYE ON
:SYSTem:AUToscale;*OPC?
```

- If option 500 is *not* installed, you can use the **:CHANnel:TDELay** command to add software delay to a channels, but the acquisition throughput will be reduced.

```
:CHANnel2A:TDELay 45.68e-12
:SYSTem:AUToscale;*OPC?
```

Precision Time Base

Because the DCA-M has different hardware than the 86100D, the DCA-M *does not work* with a the 86100D's Internal Precision Time Base (PTB). However, a DCA-M with option LOJ (Low Jitter Timebase) *does provide* PTB performance without the need for special commands. Therefore, the following FlexDCA commands *do not work* with DCA-M oscilloscopes:

```
:CALibrate:FRAMe:PTIMebase:STARt
:CALibrate:FRAMe:PTIMebase:STATus?
:TIMebase:PTIMebase:RFRequency
:TIMebase:PTIMebase:RMEThod
:TIMebase:PTIMebase:RTReference
:TIMebase:PTIMebase:STATe
:TIMebase:PTIMebase:VPTBsignals
```

Recommended Best Practices

The following techniques explicitly set values for selecting an optical channel's wavelength or filter rate. These methods are recommended for both N109X-series and 86100D oscilloscopes.

Optical wavelength selection commands

:CHANnel:WAVelength

To specify or query the input channel's wavelength, it is recommended that you *do not* use the **:CHANnel:WAVelength** command. For example,

```
:CHANnel2A:WAVelength WAVelength2
```

This is because the wavelength argument can differ between modules. For example, the **WAVelength2** argument might specify a 1310 nm wavelength on an DCA-M oscilloscope and 1550 nm on an standard receiver module.

Instead, explicitly select the wavelength using the **:CHANnel:WAVelength:VALue** command. For example,

```
:CHANnel2A:WAVelength:VALue 1.310E-6
```

This is the most robust and reliable technique for making these selections and the easiest to read! This command *works for both* traditional modules and the DCA-M oscilloscopes. If the wavelength is not within 1% of a supported wavelength, an error will be generated.

:CHANnel:WAVelength:VALue:VSET?

You can query the available wavelengths for setting using the **:CHANnel:WAVelength:VALue:VSET?** query. This query returns a comma separated list of available wavelengths in meters for the selected optical channel. For example,

```
"1.555E-6,850E-9,1.310E-6,1.550E-6,…"
```

Filter rate selection commands

:CHANnel:FSELect

To specify or query the input channel's reference filter, it is recommended that you *do not* use the **:CHANnel:FSELect** command. For example,

```
:CHANnel2A:FSELect:FILTer2
```

This is because the filter argument can differ between modules and module options. For example, the **FILTer2** argument can specify a different filter based on DCA-M oscilloscope or standard module and with different reference filter rate options. For example, it might be 8.5 Gb/s in one case and 9.953280 Gb/s in another.

Instead, explicitly select the wavelength using the **:CHANnel:FSELect:RATE** command. For example,

```
:CHANnel2A:FSELect:RATE 10.3125E9
```

This is the most robust and reliable technique for making these selections and the easiest to read! This command *works for both* traditional modules and the DCA-M oscilloscopes. If the wavelength is not within 1% of a supported reference filter rate, an error will be generated.

:CHANnel:FSELect:RATE:VSET?

You can query the available reference filters for setting using the **:CHANnel:**FSELect**:**RATE**:VSET?** query. This query returns a comma separated list of available filters in b/s for the selected optical channel. For example,

```
"155.000E+6,622.000E+6,1.250000E+9,2.488000E+9,…"
```

:CHANnel:FILTer OFF

The **:CHANnel:FILTer OFF** command does *not* work with DCA-M oscilloscopes. Instead, select the highest rate filter using either of the following two commands:

:CHANnel:FSELect:RATE <*highest rate filter*>

:CHANnel:FSELect:RATE:MAXimum

Reference filters are low-pass filters and the highest rate filter is essentially the same bandwidth as not having a filter installed. Refer to Table 4 on page 13 for a list of reference filters provided for each option.

Using the SCPI Recorder

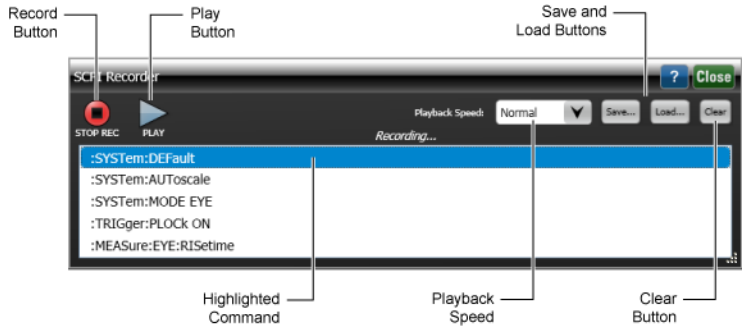
FlexDCA's SCPI recorder is a powerful tool that enables you to quickly discover, record, and learn about the commands needed to perform almost any task remotely. To open the SCPI recorder in FlexDCA, click **Tools** > **SCPI Programming Tools** > **SCPI Recorder**.

When you've recorded your commands, you can copy them to the Windows clipboard, play them back, or save them as a macro. Click **REC** and the recorder performs the following tasks:

1 Captures most FlexDCA setting changes (mouse clicks or keyboard presses).

2 Translates the setting into an equivalent SCPI remote-programming command.

3 Records the SCPI command within the dialog box.



Click **REC** to begin recording. The button turns red to alert you that any mouse clicks or keyboard presses will be recorded. When you have captured all your commands, click the record button again to end the recording.

Click **Play** to play back your recording and observe the commands as they are executed. Playback always starts at the highlighted command, which is shown in a blue box as seen in the above figure. Click any command in the dialog box to highlight that command. Use Playback Speed to control the rate that the commands are played: Slow, Normal, or Fast.

During playback, click **PAUSE** to temporarily stop the playback.

During playback, click **Device Clear** when a command is waiting a unacceptable length of time to complete. This can happen, for example, during limit testing if you use a command that locks the remote interface until 100 failed samples are detected, which could take a very long time.

```
:LTESt:MTESt:FAILures 100; *OPC?
```

Click **Clear** to delete your recorded commands from the dialog box.

Click **Save** to save your recorded commands into a script file (.scpi) that can be reloaded into the dialog box at any time by clicking Load. SCPI script file are ASCII files, so you can edit them as well as create new script files using a text editor. SCPI scripts are often small snippets of code that may depend on FlexDCA being set to very specific settings. As a result, your scripts may not work in the same manner in different situations. This can be alleviated by starting your scripts with commands that configure starting conditions. For example, the first few lines in the following code configures FlexDCA from a default setup before making a rise time measurement. To include comments in your scripts, precede the comments with two forward-slash characters as shown in this example:

```
:SYSTem:DEFault
:TRIGger:SOURce FPANel
:TRIGger:BWLimit CLOCk
:SYSTem:MODE EYE         // Place FlexDCA in Eye/Mask mode
:SYSTem:AUToscale
:TRIGger:PLOCk ON
:MEASure:EYE:RISetime // Measure risetime
:MEASure:EYE:RISetim?
```

NOTE You can use the Windows clipboard to copy your recorded commands. Click on a command to select it. Shift click to select a range of commands. Ctrl click to select non-adjacent commands. Ctrl-A selects all commands. Ctrl-C copies all selected commands. In another application, use Ctrl-V to paste the commands. You can also right-click and then click Copy.

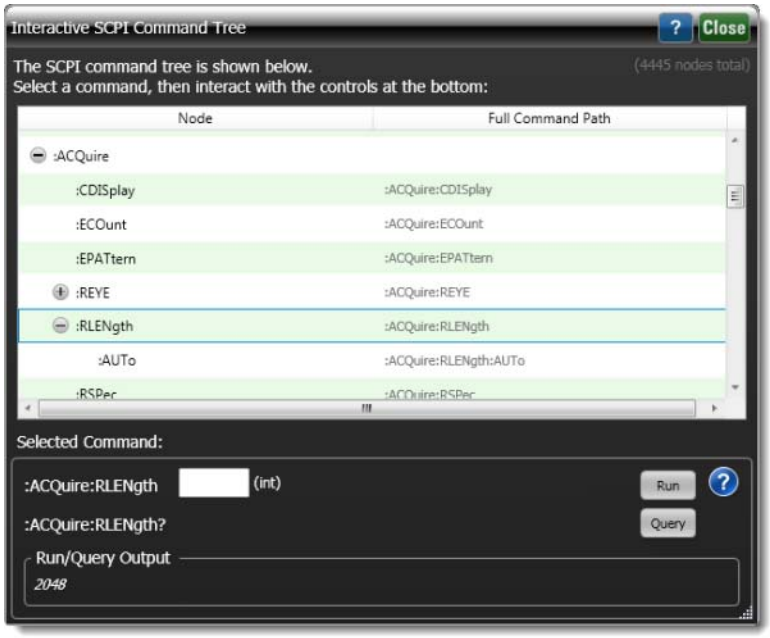
NOTE The default user data folder for saving script files is \SCPI Scripts.

NOTE When FlexDCA is installed, a registry entry is created to associate script files (.SCPI) with Notepad. This makes it easier to open script files.

Using the Interactive SCPI Command Tree

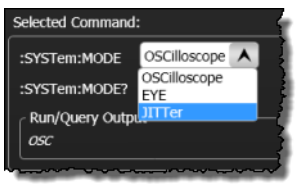
FlexDCA's SCPI Remote Command Tree Viewer lists all SCPI command subsystems and their commands. You can quickly traverse the tree and test the interaction between any command and FlexDCA.

To open the SCPI recorder in FlexDCA, click **Tools** > **SCPI Programming Tools** > **Interactive SCPI Command Tree**.



Click the icon to open the help topic for the selected command.

To learn about a command or subsystem, scroll down the command tree to locate the commands that interest you, such as the **:SYSTem** subsystem. Click on the expand button to view the available commands. You can also expand an entry by double clicking on the row. The following figure shows the **:SYSTem:MODE** command. Notice that the drop-down list provides all of the available arguments for the command. You can make a selection, and click Run to execute the command. Click **Query** to display the current value in the Run/Query Output field. Some commands, like **:ACQuire:RLENgth** shown in the above figure, take an integer (int) or real (double) value.



FlexDCA Configurations

This section describes the four basic setup configurations for controlling FlexDCA via the LAN as illustrated in Figure 17 on page 52. Select the configuration that matches your needs and run the example program.

For each setup, there is a very simple Python script that establishes the connection to FlexDCA on the PC or 86100D and prints the response of FlexDCA's ***IDN?** query. An N109X-series oscilloscope is not required. The four simple example scripts show how similar the task is for each setup. If you would like to install Python, refer to "To Install Python 3.5 and 2.7" on page 60.

When FlexDCA is being remotely controlled, front panel control is locked out. If you click **Local** on FlexDCA, the remote control link disconnects and control is returned to FlexDCA user interface.

The version of both the PC's FlexDCA and the 86100D's FlexDCA must be A.05.30 or above and must be the same version.

Environment

All of the examples shown in this section were written for Python 3.5 and 2.7 and were all tested using the following environment:

- Windows 7 PC
- Keysight IO Libraries Suite installed for VISA (Virtual Instrument Software Architecture) library. Go to *http://www.keysight.com/find/iosuite*.
- Python 3.5. Anaconda Python distribution (Windows 32-bit) from Continuum Analytics installed using the graphical installer. Python 2.7 also activated as explained in "To Install Python 3.5 and 2.7" on page 60. For information on Anaconda, go to *https://www.continuum.io/downloads*.
- PyVISA which is a Python front end for the VISA library. The correct version of PyVISA is provided with Anaconda. To learn about PyVISA, go to *http://pyvisa.sourceforge.net/pyvisa/* and to *pyvisa.readthedocs.org*.

NOTE **All of the examples shown in this section present a Python script that establishes a LAN connection using the HiSLIP interface. GPIB connections are not presented. Some firewall applications might block SICL/LAN communications.**

NOTE **If you are viewing this page as a PDF file, select and copy code examples to the Windows clipboard.**

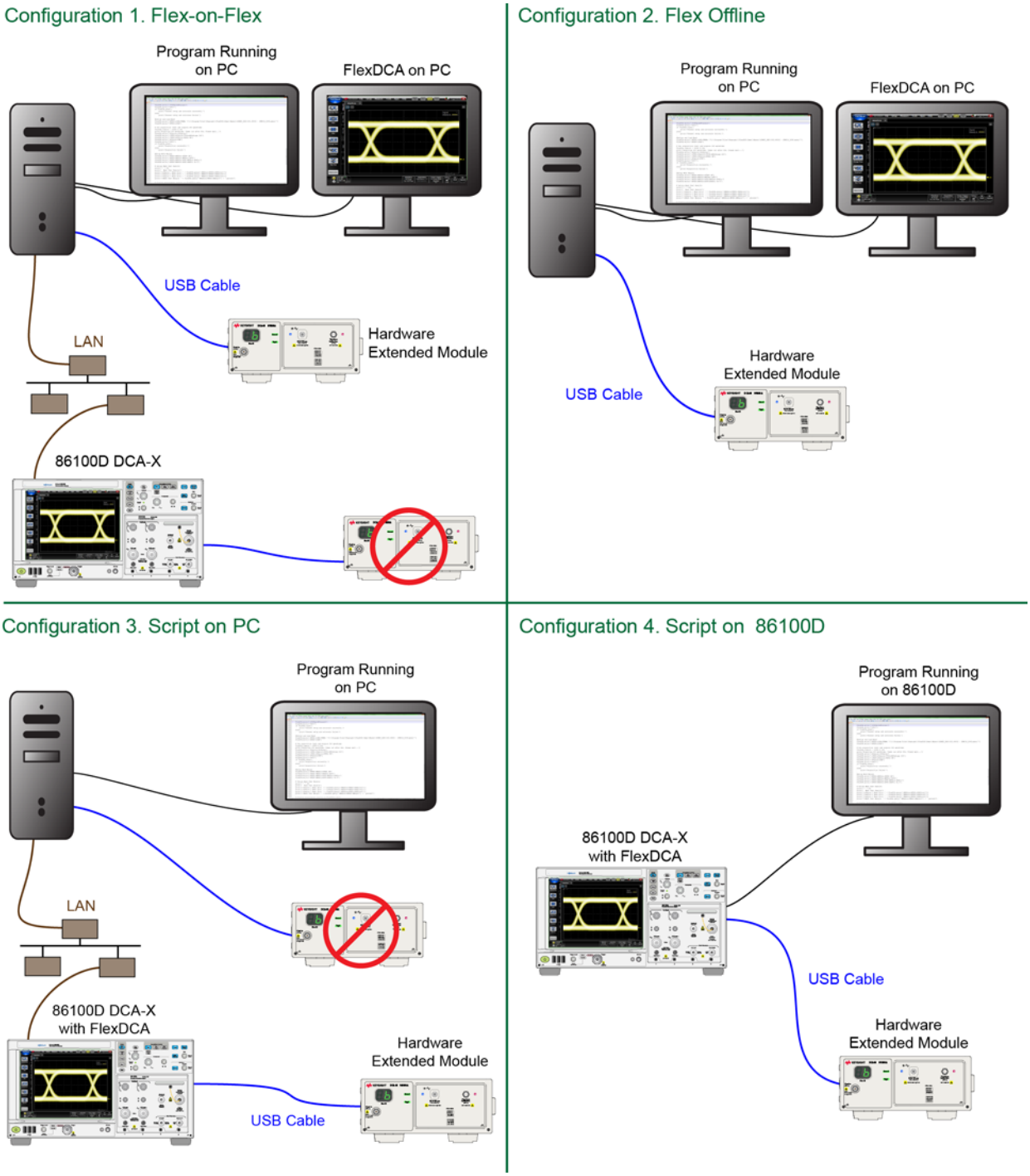
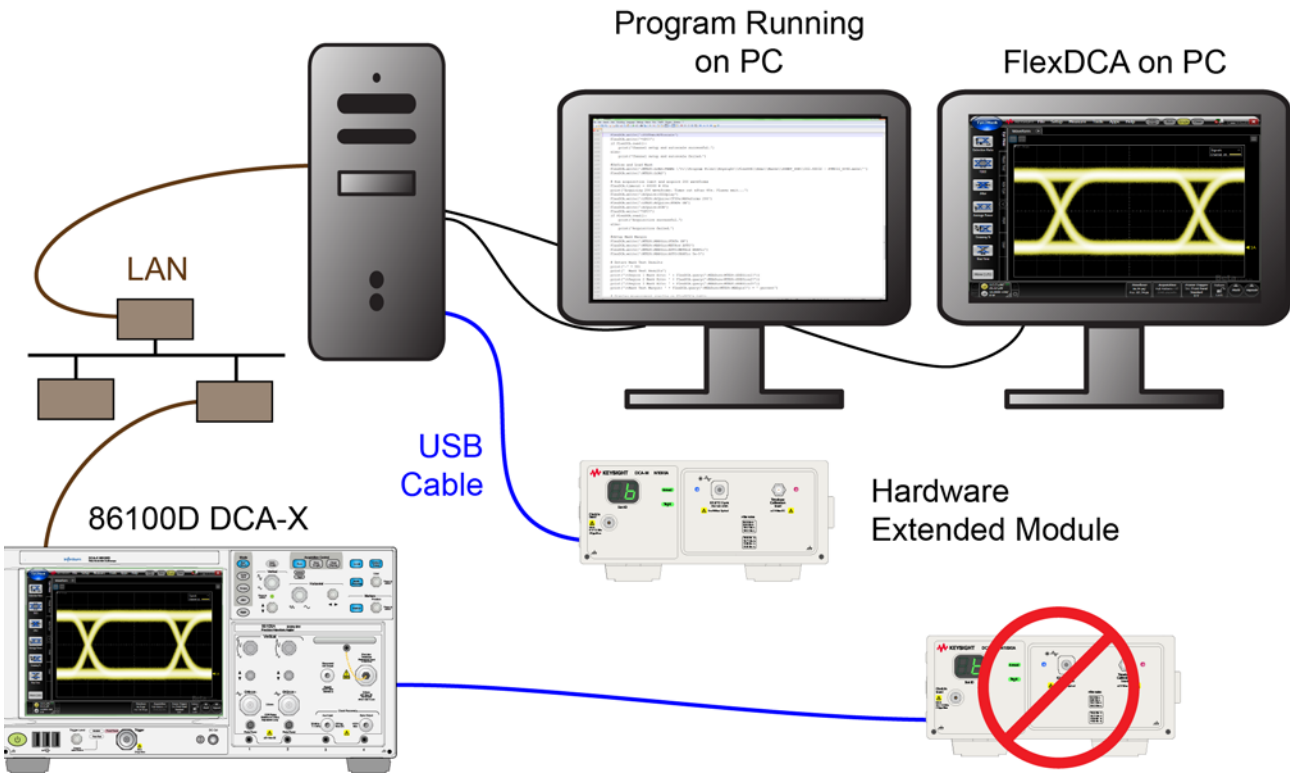


Figure 17 The Four Configurations

Configuration 1. Flex-on-Flex



In this setup, the script runs on the PC and directly controls FlexDCA on the PC. In the script, the line shown in red opens a connection between the script and the PC.

The text shown in green is the 86100D's computer name. Replace this green text with the name of your 86100D. In 86100D's FlexDCA, click the menu commands **Help** > **About FlexDCA**. In the About 86100D dialog box, locate the System Information field and scroll the field's listing until the Computer Name listing appears.

Program Listing 1. Configuration 1, Flex-on-Flex

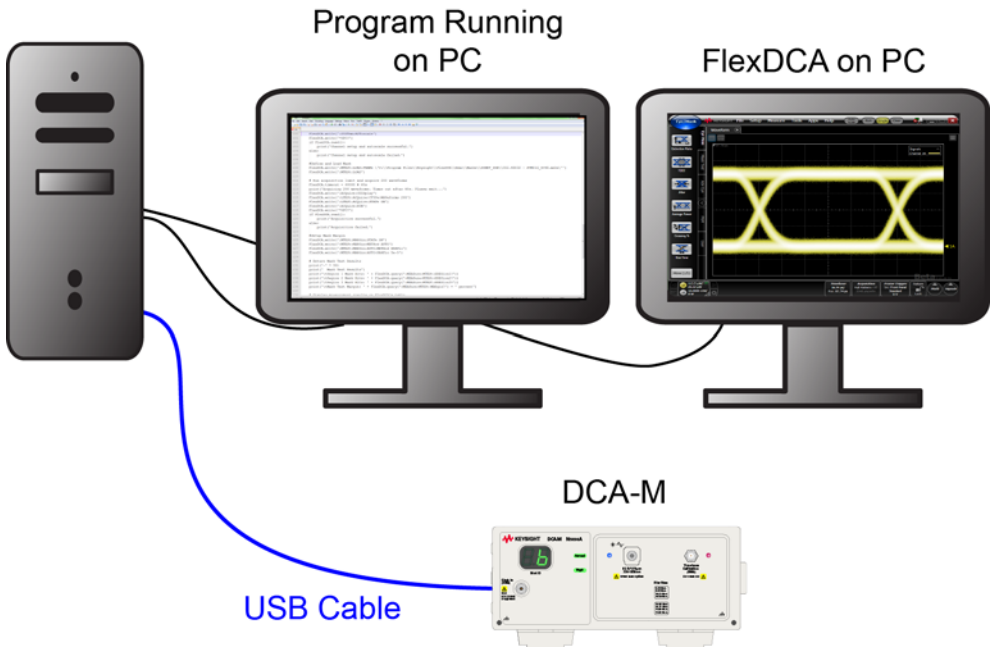
```
import visa # import VISA library
rm = visa.ResourceManager() # Create an instance of PyVISA's ResourceManager
print('\n\tConnecting to FlexDCA on PC. Please wait...')
flex=rm.open_resource('TCPIP0::localhost::hislip0,4880::INSTR')
flex.timeout=20000 # wait up to 20s for response from FlexDCA
flex.read_termination = '\n'
s = flex.query('*IDN?')
print('\tFlexDCA connection established to:\n\t' + s) # Identify FlexDCA
print('\n\tConnecting FlexDCA to 86100D. Please wait...')
flex.write(':RDCA:CONNect:METhod LAN')
flex.write(':RDCA:CONNect:MODE STANdard')
flex.write(':RDCA:CONNect:ACTion TBSettings')
flex.write(':RDCA:CONNect:TSETtings ON') # Pull state upon connect
flex.write(':RDCA:DISConnect:TSETtings ON') # Push state upon disconnect
flex.write(':RDCA:CONNect:HOST "K-86100D-20108"')
flex.query(':RDCA:CONNect;*OPC?')
s = flex.query('*IDN?')
print('\tFlexDCA connection established to:\n\t' + s) # Identify FlexDCA
flex.query(':RDCA:DISConnect;*OPC?')
print('\n\tFlexDCA on 86100D is disconnected.')
flex.write(':SYSTem:GTLocal')
flex.close()
print('\tFlexDCA on PC is disconnected.')
```

To manually make the Flex-on-Flex connection

This script establishes the connection between the PC's FlexDCA and 86100D's FlexDCA using the commands in the **:RDCA** subsystem. You can perform this task manually using the following steps:

1. In 86100D's FlexDCA, click the menu commands **Help** > **About FlexDCA**. In the About 86100D dialog box, locate the **System Information** field and scroll the field's listing until the Computer Name listing appears.
2. Write down the computer name (for example, "K-86100D-20108"). Close the dialog box.
3. On the PC, start FlexDCA and click the menu commands **Setup** > **DCA Connection Setup**. In the dialog box, select LAN. In the Hostname field, enter the computer name that you located in step 2. Click Connect to manually connect FlexDCA with the 86100D.

Configuration 2. FlexDCA Offline

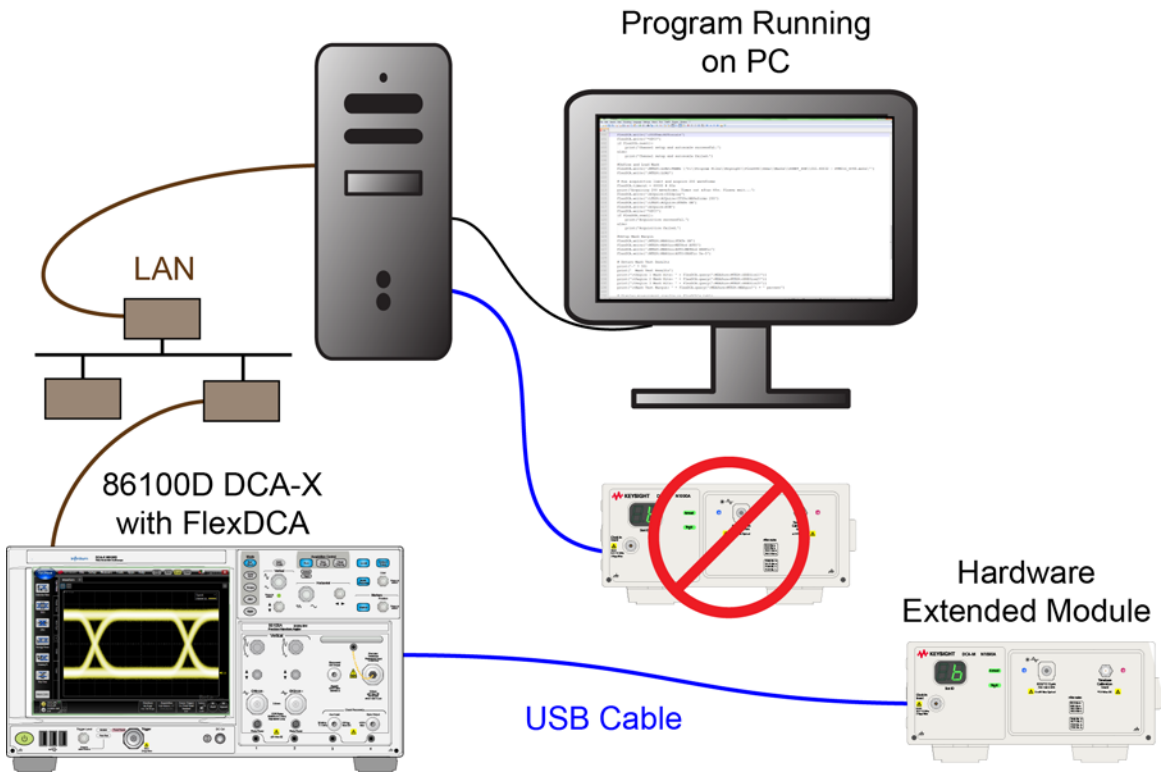


In this setup, N1010A FlexDCA is being used *offline* without a connection to an 86100D. In the script, the VISA address shown in red is used to open a connection between the script and the PC.

Program Listing 2. Configuration 2, FlexDCA Offline

```
import visa # import VISA library
rm = visa.ResourceManager() # Create an instance of PyVISA's ResourceManager
flex=rm.open_resource('TCPIP0::localhost::hislip0,4880::INSTR')
flex.read_termination = '\n'
s = flex.query('*IDN?')
print('\n\tFlexDCA connection established to:\n\t' + s) # Identify FlexDCA
flex.write(':SYSTem:GTLocal')
flex.close()
print('\tFlexDCA connection closed.')
```

Configuration 3. Script on PC

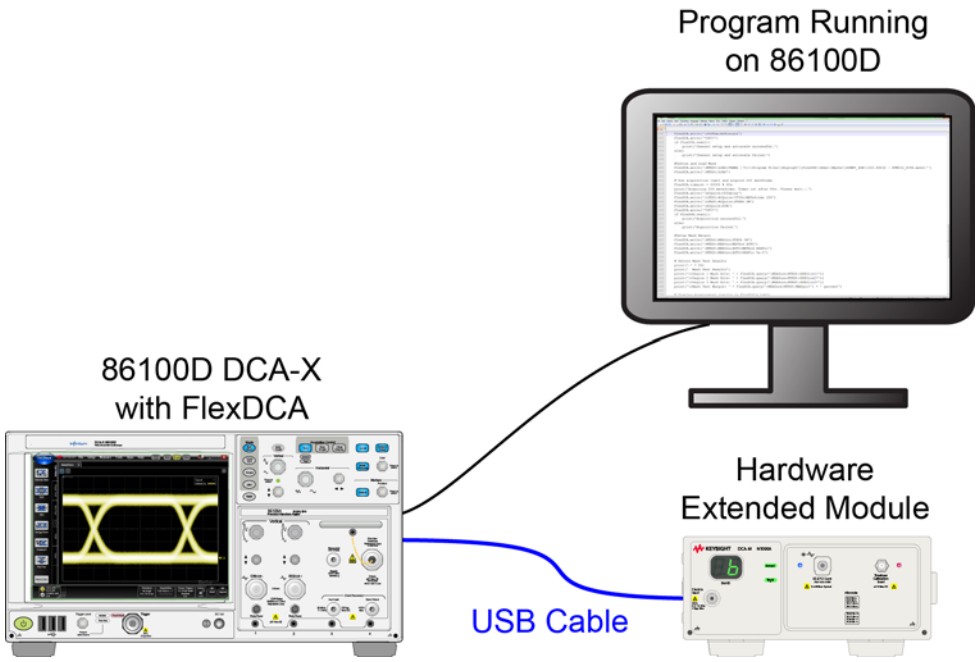


In this setup, the script runs on the PC and controls an 86100D. In the script, replace the 86100D's computer name (shown in red) with the name of your 86100D. In 86100D's FlexDCA, click the menu commands **Help** > **About FlexDCA**. In the About 86100D dialog box, locate the System Information field and scroll the field's listing until the Computer Name listing appears.

Program Listing 3. Configuration 3, Script on PC

```
import visa # import VISA library
rm = visa.ResourceManager() # Create an instance of PyVISA's ResourceManager
flex=rm.open_resource('TCPIP0::K-86100D-20108::hislip0,4880::INSTR')
flex.read_termination = '\n'
s = flex.query('*IDN?')
print('\n\tConnection established to:\n\t' + s) # Identify FlexDCA
flex.write(':SYSTem:GTLocal')
flex.close()
print('\tConnection closed.')
```

Configuration 4. Script on 86100D



In this setup, the script runs on the 86100D. In the script, the following line opens a connection between the script on the 86100D's Windows and FlexDCA. The VISA address is shown in red in the following script.

Program Listing 4. Configuration 4, Script on 86100D

```
import visa # import VISA library
rm = visa.ResourceManager() # Create an instance of PyVISA's ResourceManager
flex=rm.open_resource('TCPIP0::localhost::hislip0,4880::INSTR')
flex.read_termination = '\n'
s = flex.query('*IDN?')
print('\n\tConnection established to:\n\t' + s) # Identify FlexDCA
flex.write(':SYSTem:GTLocal')
flex.close()
print('\tConnection closed.')
```

Finding the VISA Address for HiSLIP

The VISA Address for the HiSLIP interface is easy to locate and is used to identify the LAN address between your program and FlexDCA.

1 On FlexDCA, click the menu commands **Tools** > **SCPI Programming Tools**. The SCPI Server Setup dialog box opens as shown below.

2 From the dialog box, write down the complete listed VISA Address for HiSLIP. As an alternative, you can highlight the string using the mouse cursor and press Ctrl-C to copy it to the clipboard for later use. In the above picture, the string for HiSLIP would be:

TCPIP0::2UA2421081::hislip0,4880::INSTR

You can substitute the IP address as shown here:

TCPIP0::141.121.89.103::hislip0,4880::INSTR

If your program is running on the same PC or 86100D as FlexDCA, you can simplify this string by substituting *localhost* for the computer name. For example, you would modify the above VISA address to be the following string:

TCPIP0::localhost::hislip0,4880::INSTR

Example Programs

This chapter presents several example programs for controlling the N109X-series oscilloscopes. These programs will run in Python 2.7 or 3.X and are intended to show basic tasks that you can adapt to programs that are written in another language. These examples demonstrate how to establish a LAN connection to FlexDCA and how to use FlexDCA commands and techniques to accomplish specific FlexDCA tasks. These examples are not provided to teach the Python language.

CAUTION **Do not modify indents at the beginning of any lines within the program. The indentation of code lines in Python is critical to the ability of the code to run. Indents define code blocks.**

Getting the Code Examples

You can copy the code examples from the PDF version of this documents, but the most reliable and easiest method for getting the source code is by copying it from FlexDCA's help system:

1. Open FlexDCA's help and under "Programming", click the *Python Examples* link.
2. In the *Programming Examples* topic, click on one script that you are interested in.
3. In the script topic, download the script from the "Get Example Script" sidebar. From the opened window, copy the code listing and paste it in your Python editor.

Example Code Environment

All of the examples shown in this chapter were written in Python 3.4 and were all tested using the following environment:

- Windows 7 PC
- Python 3.5.1. Anaconda Python distribution 2.4.0 (Windows 32-bit) from Continuum Analytics installed using graphical installer.
- Keysight IO Libraries Suite installed for VISA (Virtual Instrument Software Architecture) library.
- PyVISA which is a Python front end for the VISA library. Go to http://pyvisa.sourceforge.net/pyvisa/. For more information, go to *pyvisa.readthedocs.org*.

To Install Python 3.5 and 2.7

Anaconda automatically installs the code for both Python 3.x and 2.x. Starting with step 5, the following procedure installs Python 2.7. The example Python scripts in this book are relatively simple, so you should be able to convert them to 2.x is you prefer to work in Python 2.x.

1. Install Keysight IO Libraries Suite. Download the suite from *http://www.keysight.com/find/iosuite*.
2. Go to *https://www.continuum.io/downloads* and download the graphical Python 3.5 Anaconda Python distribution (Windows 32-bit). Don't worry, you'll get both Python 3.5 and 2.7.
3. Open Windows Command Prompt. To locate Command Prompt, search for *cmd* on the Start menu.
4. In Windows Command Prompt, enter the following commands to confirm the installation and to install PyVISA for Python 3.5.

```
cd c:\
conda info
pip install pyvisa
```

5. Enter the following commands to create a create a Python 2.7 environment.

```
conda create -n py27 python=2.7 anaconda
```

6. When the Python 2.7 installation is complete, activate Python 27 and install PyVISA for Python 2.7.

```
activate py27
pip install pyvisa
```

7. Use the following command to start Spyder, Anaconda's Python script editor. Since you've just activated Python 2.7, Spyder will open using Python 2.7. In Spyder, locate the IPython Console pane. The version of Python running will be listed at the top of IPython Console.

```
spyder
```

8. To use Python 3.5, enter the following commands in the Windows Command Prompt.

```
deactivate py27
spyder
```

General Information on the Scripts

print() statements

The Python 3.x **print()** statement is used to provide messages on the script's progress to the user. This works in Python 2.x as long as you do not pass multiple strings to the function.

Message timing

When the **print()** statement is used as described above, under certain conditions your message may appear on screen sometime later than they should and this can be very confusing. For example, if you acquiring waveforms as shown here:

```
print('Acquiring 100 waveforms.')
FlexDCA.write(:ACQuire:RUN';*OPC?)
```

You'll watch FlexDCA capture the waveforms and then write the statement "Acquiring 100 waveforms" to the screen! The solution is to flush the stdout buffer. In Python 3.x, this can be done by:

```
print('Acquiring 100 waveforms.', flush=True)
FlexDCA.write(:ACQuire:RUN';*OPC?)
```

In Python 2.x and 3.x, the same task can be accomplished with:

```
print('Acquiring 100 waveforms.')
sys.stdout.flush()
FlexDCA.write(:ACQuire:RUN';*OPC?)
```

Reading user input

The input statements for getting a response from a user are different between Python versions. In Python 3.x, use the **input()** function. In Python 2.x, use the **raw_input()** function. In versions of the scripts in steps 3 and 4, you must comment out one of two input lines before running the script. For example, to run the script in Python 3.x:

```
# PYTHON VERSION SPECIFIC CODE!
input('Press any key to continue!') # Python 3.x
#raw_input('Press any key to continue!') # Python 2.x
```

Example 1. Identify all connected DCA-Ms and modules

The following Python script discovers all DCA-M oscilloscopes, standard modules, and simulated modules and prints their model numbers to the console. The script uses the **:SYST:MODel? SLOT** query to return the model number of a standard or DCA-M module. The **:EMODules:SLOT:SELection?** query is used to return the string "DCAM" if a DCA-M module is installed in the slot. If **EMPT** is returned, the slot is empty. These commands are shown in the color red in the script.

Program Listing 5. DCA_M_find.py

```
# -*- coding: utf-8 -*-

import visa # import VISA library

slots = {1:'',2:'',3:'',4:'',5:'',6:'',7:'',8:''}

def discover_modules(flex, slots):
    """ Locates each hardware, simulated, or DCA-M by slot."""
    for slot in range(1,9): # slot numbers
        model = flex.query(':SYST:MODel? SLOT' + str(slot))
        if model in 'Not Present':
            continue
        elif model in 'N1010A': # Simulated module found
            slots[slot] = flex.query(':EMODules:SLOT' + str(slot) + ':SELection?')
        else: # standard module found
            slots[slot] = model
    if list(slots.values()).count('') == 8:
        print('No modules found!')
        return {}
    else: return slots

def report_modules_found(flex, slots):
    """ Print list of installed modules."""
    print('\nInstalled modules:')
    simulated = {'DEM':'DEM (Simulated dual electrical)',
                 'DOM':'DOM (Simulated dual optical)',
                 'QEM':'QEM (Simulated quad electrical)',
                 'OEM':'OEM (Simulated electrical/optical)'}
    for slot in slots:
        model = slots[slot]
        if model in simulated: #eg. 'DEM' is a key
            print('- Slot '+ str(slot) +': '+ simulated[model[0:3]])
        elif model: # non-simulated module exists
            print('- Slot '+ str(slot) +': '+ model) # eg, N1092A

rm = visa.ResourceManager() # Create an instance of PyVISA's ResourceManager
flex=rm.open_resource('TCPIP0::K-86100D-20108::hislip0,4880::INSTR')
flex.read_termination = '\n'
flex.timeout = 30000 # Connection timeout 30s
slots = discover_modules(flex, slots)
if slots:
   report_modules_found(flex, slots)
flex.write(':SYSTem:GTLocal')
flex.close()
```

Example 2. Re-assign a DCA-M to a different slot

The script in Program Listing 6 moves an existing N109X-series oscilloscope from slot 5 to slot 8. The script in Program Listing 7 first disconnects any existing DCA-Ms and deletes any simulated modules. The script then connects up three DCA-Ms (dictionary values) in the desired slots (dictionary keys).

If you try these examples, be sure to substitute the correct device names for your DCA-Ms (shown in red).

Program Listing 6. DCA_M_reassign.py

```
import visa # import VISA library

def move_DCAM(flex):
    """ Moves DCA-M from slot 5 to slot 8."""
    flex.write(':EMODules:SLOT5:DISConnect')
    flex.write(':EMODules:SLOT8:SELection DCAM')
    flex.write(':EMODules:DCAM8:DEVice "N1092D-US56999999"')
    flex.write(':EMODules:SLOT8:CONNect')


rm = visa.ResourceManager() # Create an instance of PyVISA's ResourceManager
flex=rm.open_resource('TCPIP0::localhost::hislip0,4880::INSTR')
flex.read_termination = '\n'
move_DCAM(flex)
flex.write(':SYSTem:GTLocal')
flex.close()
```

Program Listing 7. DCA_M_reorder.py

```
# -*- coding: utf-8 -*-

import visa # import PyVISA library
import sys

my_dcam_dict = {'5':'"N1077A-US56030014"',
             '6':'"N1090A-US54420111"',
             '7':'"N1092D-US56140205"'}

def remove_all_dcam(flex):
    for slot in '12345678':
        model = flex.query(':EMODules:slot'+slot+':SELection?')
        if model in ['DCAM','DEMini','QEMini','OEMini','DOMini']:
            if 'CONN' in flex.query(':EMODules:SLOT'+slot+':STATe?'):
                print('Disconnecting DCA-M in slot '+slot+' ...')
                sys.stdout.flush()
                flex.query(':EMODules:SLOT'+slot+':DISConnect;*OPC?')
    print('All DCA-M and simulated modules disconnected.\n')
    sys.stdout.flush()

def connect_dcam(flex, my_dcam_dict):
    """ Assigns my DCA-Ms to slots. """
    for key, value in my_dcam_dict.items():
        flex.write(':EMODules:DCAM'+key+':DEVice '+value)
        flex.write(':EMODules:SLOT'+key+':SELection DCAM')
        flex.query(':EMODules:SLOT'+key+':CONNect;*OPC?')
        print(value[1:7] + ' connected to slot '+key)
        sys.stdout.flush()

rm = visa.ResourceManager() # Create an instance of PyVISA's ResourceManager
flex=rm.open_resource('TCPIP0::localhost::hislip0,4880::INSTR')
flex.read_termination = '\n'
flex.timeout =  10000 # 10s
remove_all_dcam(flex)
connect_dcam(flex, my_dcam_dict)
print('All DCA-Ms have been re-assigned to new slots.')
flex.write(':SYSTem:GTLocal')
flex.close()
```

Example 3. Perform eye measurement with N109X

This Python example uses an N109X-series DCA-M oscilloscope to perform an eye measurement (bit rate) on channel 6A. The programs performs the following tasks:

- Places FlexDCA to its default settings. This places the FlexDCA triggering to front panel which is required to view a valid eye diagram, and then makes all the required setting to view an eye diagram.
- Queries the N109X to see if it supports pattern locking (option PLK) and then turns on pattern locking if available.
- Returns the measured trigger rate that is input to the N109X.
- Runs an acquisition limit test and then activates the bit-rate measurement on the data.

About the :TRIGger:MRATe? query

The **:TRIGger:MRATe?** query returns the baud rate of the N109X's clock input signal. For this query to work, at least one of the N109X's input channels must be turned on, but a signal does not need to be connected to the channel. Also, the N109X's data acquisition must be in Run mode, or, if in single mode, a single acquisition must first be run to ensure valid data for the measurement.

To run this program

- Run this program on a PC that has FlexDCA running. In Program Listing 8 program listing, FlexDCA is identified by the red text: **localhost**.

  If you change localhost to an 86100D, you can run the program on a PC to control FlexDCA on an 86100D. Connect the N109X to the 86100D.
- Install an N109X-series oscilloscope to FlexDCA's slot 6. Change the **DCAMSLT** constant if you want to use a different slot.
- Data in is expected to be 10.3125 Gb/s, but you can change this to any acceptable rate for the oscilloscope.
- Connect valid signal to channel **6A** on the N1090X.
- Connect a clock to the N1090X's **Clock In**.
- Display an eye diagram on FlexDCA.

Program Listing 8. DCA_M_109X_pc.py (Sheet 1 of 2)

```
# -*- coding: utf-8 -*-

import visa # import VISA libraryN
import sys

DCAMSLT = '6' # N109X's slot
DRATE = '10.312500E+9' # input data rate

def setup_86100():
    """ Default all settings. Set trigger and turn on channel. """
    flex.query(':SYSTem:DEFault;*OPC?')
    flex.write(':TRIGger:SOURce FPANel') #
    for slot in range(1,9): # Only channel 6A on.
        for channel in ['A','B','C','D']:
            flex.write(':CHANnel'+str(slot)+channel+':DISPlay OFF')
    flex.write(':CHAN'+DCAMSLT+'A:DISPlay ON') # Turn channel 'A' on

def setup_dcam():
    """ Configure DCA-M and query measured clock input rate."""
    flex.write(':ACQuire:SINGle')
    clockrate = flex.query(':SLOT'+DCAMSLT+':TRIG:MRATe?')
    if '-1.00' in clockrate or '0.00' in clockrate:
        print('DCA-M is not triggered. Check inputs.')
        return False
    flex.query(':ACQ:RUN;*OPC?')
    flex.query(':SYSTem:MODE EYE;*OPC?')
    return True

def pattern_lock():
    """ If DCA-M pattern lock option is installed, configure acquisition
    test to acquire pattern waveforms. """
    if 'PLK' in flex.query(':SYST:OPTions? SLOT'+DCAMSLT):
        flex.write(':TRIGger:BRATe:AUTodetect OFF')
        flex.write(':TIMebase:BRATe ' + DRATE)
        flex.write(':SLOT'+DCAMSLT+':TRIGger:TRACking ON')
        flex.query(':TRIGger:PLOCk ON;*OPC?')
        print('Pattern lock on.')
        return True
    return False
```

Program Listing 8. DCA_M_109X_pc.py (Sheet 2 of 2)

```
def make_measurement(patternlock):
    """ Perform acquisition limit test and bit-rate measurement. """
    flex.query(':AUToscale;*OPC?')
    flex.write(':ACQuire:SINGle;:ACQuire:CDISplay')
    if patternlock:
        flex.write(':LTESt:ACQuire:CTYPe:PATTerns 20')
        print('Acquiring 20 pattern waveforms. Please wait...')
    else:
        flex.write(':LTESt:ACQuire:CTYPe:WAVeforms 100')
        print('Acquiring 100 waveforms. Please wait...')
    sys.stdout.flush() # Flush print message
    flex.write(':LTESt:ACQuire:STATe ON')
    flex.query(':ACQuire:RUN;*OPC?')
    flex.write(':LTESt:ACQuire:STATe OFF')
    flex.write(':MEASure:EYE:BITRate')
    print('Measured:')
    print('  Eye Bit Rate: ' + flex.query(':MEASure:EYE:BITRate?') + ' b/s')
    print('  Clock rate: ' + flex.query(':SLOT'+DCAMSLT+':TRIG:MRATe?') + ' baud
(Bd)')


rm = visa.ResourceManager() # Instance of PyVISA's ResourceManager
flex=rm.open_resource('TCPIP0::localhost::hislip0,4880::INSTR')
flex.timeout = 20000 # Set connection timeout to 20s
flex.read_termination = '\n'
flex.write_termination = '\n'
s = flex.query('*IDN?')
print('\nConnection established to:\n' + s) # Identify flex
setup_86100()
if setup_dcam():
    make_measurement(pattern_lock())
flex.write(':SYSTem:GTLocal')
flex.close()
```

Example 4. Perform eye measurement with N109X & N1076/7A

This Python example uses an N109X-series DCA-M oscilloscope in slot 6 and N1076/7A clock recovery in slot 5 to perform an eye measurement (bit rate) on channel 6A. It is the same as "Example 3. Perform eye measurement with N109X" on page 65, but adds clock recovery. The program performs the following tasks:

- Places FlexDCA to its default settings. This places the FlexDCA triggering to front panel which is required to view a valid eye diagram.
- Configures the clock recovery.
- Configures N109X.
- Queries the N109X to see if it supports pattern locking (option PLK) and then turns on pattern locking if available.
- Returns the measured trigger rate that is input to the N109X.
- Runs an acquisition limit test and then activates the bit-rate measurement on the data.

About the :TRIGger:MRATe? query

The **:TRIGger:MRATe?** query returns the baud rate of the N109X's clock input signal. For this query to work, at least one of the N109X's input channels must be turned on, but a signal does not need to be connected to the channel. Also, the N109X's data acquisition must be in Run mode, or, if in single mode, a single acquisition must first be run to ensure valid data for the measurement.

To run this program

- Run this program on a PC that has FlexDCA running. In Program Listing 9 program listing, FlexDCA is identified by the red text: **localhost**.

  If you change localhost to an 86100D, you can run the program on a PC to control FlexDCA on an 86100D. Connect the N109X to the 86100D.
- Install an N109X-series oscilloscope to FlexDCA's slot 6. Change the **DCAMSLT** constant if you want to use a different slot.
- Install an N1076/7A clock recovery to FlexDCA's slot 5. Change the **RECSLT** constant if you want to use a different slot.
- Data in is expected to be 10.3125 Gb/s (**DRATE** constant). But you can change this to any acceptable rate for the oscilloscope.
- Connect valid signal to channel **6A** on the N1090X.
- Connect a clock to the N1090X's **Clock In**.
- Display an eye diagram on FlexDCA.

Program Listing 9. DCA_M_109X_1077_pc.py (Sheet 1 of 2)

```
# -*- coding: utf-8 -*-

import visa # import VISA library
import sys

RECSLT = '5' # N1077A's slot
DCAMSLT = '6' # N1092D's slot
DRATE = '10.312500E+9' # input data rate

def setup_86100():
    """ Default all settings. Set trigger and turn on channel. """
    flex.query(':SYSTem:DEFault;*OPC?')
    flex.write(':TRIGger:SOURce FPANel') #
    for slot in range(1,9): # Only channel 6A on.
        for channel in ['A','B','C','D']:
            flex.write(':CHANnel'+str(slot)+channel+':DISPlay OFF')
    flex.write(':CHAN'+DCAMSLT+'A:DISPlay ON') # Turn channel 'A' on

def setup_n1077a():
    """ Configure N1077A clock recovery. """
    flex.write(':CRECovery'+RECSLT+':SOURce ELECtrical') # set input type
    flex.write(':CRECovery'+RECSLT+':CRATe ' + DRATE) # set input data rate
    flex.write(':CRECovery'+RECSLT+':CLBandwidth 5.692E+6') # set PLL loop BW
    flex.query(':CRECovery'+RECSLT+':RELock;*OPC?') # Lock clock recovery

def setup_dcam():
    """ Configure DCA-M and query measured clock input rate."""
    flex.write(':ACQuire:SINGle')
    clockrate = flex.query(':SLOT'+DCAMSLT+':TRIG:MRATe?')
    if '-1.00' in clockrate or '0.00' in clockrate:
        print('DCA-M is not triggered. Check inputs.')
        return False
    flex.query(':ACQ:RUN;*OPC?')
    flex.query(':SYSTem:MODE EYE;*OPC?')
    return True

def pattern_lock():
    """ If DCA-M pattern lock option is installed, configure acquisition
    test to acquire pattern waveforms. """
    if 'PLK' in flex.query(':SYST:OPTions? SLOT'+DCAMSLT):
        flex.write(':TRIGger:BRATe:AUTodetect OFF')
        flex.write(':TIMebase:BRATe ' + DRATE)
        flex.write(':SLOT'+DCAMSLT+':TRIGger:TRACking ON')
        flex.query(':TRIGger:PLOCk ON;*OPC?')
        print('Pattern lock on.')
        return True
    return False
```

Program Listing 9. DCA_M_109X_1077_pc.py (Sheet 2 of 2)

```
def make_measurement(patternlock):
    """ Perform acquisition limit test and bit-rate measurement. """
    flex.query(':AUToscale;*OPC?')
    flex.write(':ACQuire:SINGle;:ACQuire:CDISplay')
    if patternlock:
        flex.write(':LTESt:ACQuire:CTYPe:PATTerns 20')
        print('Acquiring 20 pattern waveforms. Please wait...')
    else:
        flex.write(':LTESt:ACQuire:CTYPe:WAVeforms 100')
        print('Acquiring 100 waveforms. Please wait...')
    sys.stdout.flush() # Flush print message
    flex.write(':LTESt:ACQuire:STATe ON')
    flex.query(':ACQuire:RUN;*OPC?')
    flex.write(':LTESt:ACQuire:STATe OFF')
    flex.write(':MEASure:EYE:BITRate')
    print('Measured:')
    print('  Eye Bit Rate: ' + flex.query(':MEASure:EYE:BITRate?') + ' b/s')
    print('  Clock rate: ' + flex.query(':SLOT'+DCAMSLT+':TRIG:MRATe?') + ' baud
(Bd)')


rm = visa.ResourceManager() # Instance of PyVISA's ResourceManager
flex=rm.open_resource('TCPIP0::localhost::hislip0,4880::INSTR')
flex.timeout = 20000 # Set connection timeout to 20s
flex.read_termination = '\n'
flex.write_termination = '\n'
s = flex.query('*IDN?')
print('\nConnection established to:\n' + s) # Identify flex
setup_86100()
setup_n1077a()
if setup_dcam():
    make_measurement(pattern_lock())
flex.write(':SYSTem:GTLocal')
flex.close()
```

Example 5. Pattern acquisition with save

The script in Program Listing 10 acquires a pattern waveform in oscilloscope mode. An N109X-series oscilloscope must be installed in slot 5. If the DCA-M oscilloscope is queried to ensure that option PLK (pattern lock) is installed. N1090A oscilloscopes are not allowed as they do not have the option PLK.

The pattern waveform is saved in a file with the internal ".wfmx" format. This format can be re-imported into FlexDCA's waveform memory if needed. To run this script, the following items must be set:

- A DCA-M that has option PLK (pattern lock capability).
- The DCA-M installed in slot 5.
- Channel 5A is used. If the DCA-M is installed in a different slot or if a different channel is used, edit the script for the correct values.

Program Listing 10. DCA_M_pattern_acquisition.py (Sheet 1 of 2)

```
# -*- coding: utf-8 -*-

import sys
import time
import visa # import VISA library

def setup(chan):
    print('Configuring basic settings. Please wait...')
    sys.stdout.flush() # Flush print message in stdout
    flex.write(':TRIGger:SOURce FPANel')
    flex.write(':CHANnel' + chan + ':DISPlay ON')
    flex.write(':ACQuire:RUN')
    flex.query(':SYSTem:MODE OSCilloscope;*OPC?')
    flex.query(':TRIGger:PLOCk ON;*OPC?')
    flex.query(':SYSTem:AUToscale;*OPC?')
    flex.write(':ACQuire:EPATtern ON') # Acquire entire pattern
    flex.write(':ACQuire:SPBit:MODe MANual') # Select samples/bit
    flex.write(':ACQuire:SPBit 31.99212598') # 32 samples/bit
    flex.write(':ACQuire:SMOothing AVERage') # Average smoothing selected
    PatternLength = flex.query(':TRIGger:PLENgth?') # Get pattern length
    flex.write(':TIMebase:BRANge ' + str(PatternLength)) # View entire pattern

def run_acquisition_limit_test(chan):
    """ Configures and runs acquisition limit test on 32 patterns.  """
    # Specify pattern waveform file (.wfmx internal format)
    file = '"%USER_DATA_DIR%\\Waveforms\\Pattern_Waveform_Internal.wfmx"'
    flex.write(':DISK:WAVeform:FFORmat INTernal')
    flex.write(':LTESt:ACQuire:SWAVeform:RESet') # do not save waveforms
    flex.write(':LTESt:ACQuire:SWAVeform:CHANnel'+ chan +':STATe DISK')
    flex.write(':LTESt:ACQuire:SWAVeform:CHANnel'+ chan +':FNAMe '+ file)
    flex.write(':LTESt:ACQuire:CTYPe:PATTerns 32')
    flex.write(':LTESt:ACQuire:CTYPe PATTerns')
    flex.write(':ACQuire:ECOunt 32')
    flex.write(':ACQuire:SINGle;:ACQuire:CDISplay')
    flex.write(':LTESt:ACQuire:STATe ON')
    print('Acquiring 32 patterns. Timeout set to 20s.\nPlease wait...')
    sys.stdout.flush() # Flush print message in stdout
    flex.query(':ACQuire:RUN;*OPC?')
    flex.write(':LTESt:ACQuire:STATe OFF')
    flex.write(':ACQuire:STOP')
    s = flex.query(':LTESt:ACQuire:SWAVeform:CHANnel'+ chan +':FNAMe?')
    print('File saved: ' + s)
```

Program Listing 10. DCA_M_pattern_acquisition.py (Sheet 2 of 2)

```
#_____________________________________________________
#   Main script

channel = '5A'
rm = visa.ResourceManager() # Create an instance of PyVISA's ResourceManager
flex=rm.open_resource('TCPIP0::localhost::hislip0,4880::INSTR')
flex.read_termination = '\n'
flex.write_termination = '\n'
flex.timeout = 20000 # Connection timeout 20s
s = flex.query('*IDN?')
print('\tFlexDCA connection established to:\n\t' + s) # Identify flex
sys.stdout.flush() # Flush print message in stdout
DCAM = ['N1092A','N1092B','N1092C','N1092D','N1092E','N1094A','N1094B']

while True:
    model = flex.query(':SYSTem:MODel? SLOT' + channel[0])
    print('\nModel: ', model)
    if model not in DCAM: break
    options = flex.query(':SYSTem:OPTions? SLOT' + channel[0])
    print('Options: ', options)
    if 'PLK' not in options: break
    setup(channel)
    run_acquisition_limit_test(channel)
    flex.write(':SYSTem:GTLocal')
    time.sleep(0.5)
    flex.close()
    break
print('\nProgram has finished.')
```

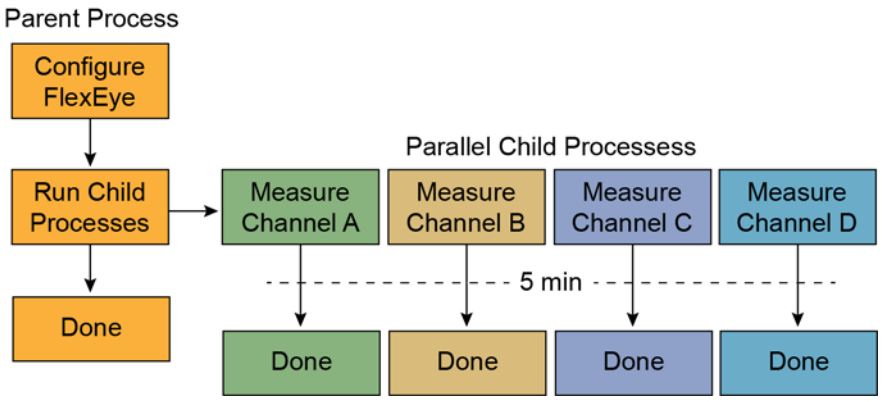
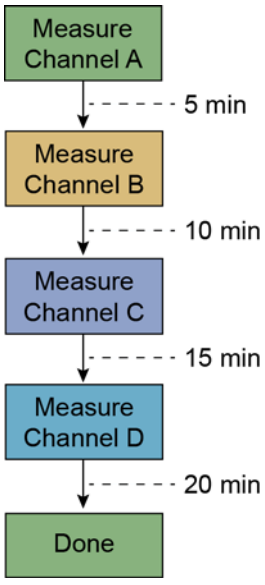
Example 6. FlexEye Independent Eye Acquisition

What is FlexEye

This example uses Eye mode's FlexEye Independent Eye Acquisition and Analysis to greatly reduce Eye measurement test times in computer-controlled environments. FlexEye runs measurements asynchronously by turning selected input channels into independent FlexDCA sampling oscilloscopes known as sessions. A child process is run on each session and performs the measurements as shown in the illustrations in the following table.

As a FlexEye example, let's say that you wrote a program that measures the change in eye width over 5 minutes and that you need to run this measurement on four channels (four DUTs). Traditionally, you would perform each measurement sequentially for a total test time of 20 minutes. With FlexEye, these same measurements can be performed asynchronously by launching four asynchronous child processes. As all measurements are simultaneously run, the total test time for all channels is just slightly over 5 minutes. There is a Python 2.7 example program that demonstrates just such a task.

Table 13 FlexEye vs. Traditional Measurements

With FlexEye (Asynchronous Measurements)	Traditional (Synchronous Measurements)
	

What the example does

This program simply performs an eye width and eye height measurement for each session waveform and reports the readings along with the time required to run each session.

To run the example

For information about this example, complete with example code, follow these steps:

1. Open FlexDCA's help system (Version A.05.50 or above).
2. Select the Search tab in the help's left pane.
3. Enter "FlexEye testing: with DCA-M module". Be sure to include the quote (") characters. Then, click **List Topics**.
4. Select the topic title in the search list and click **Display**.

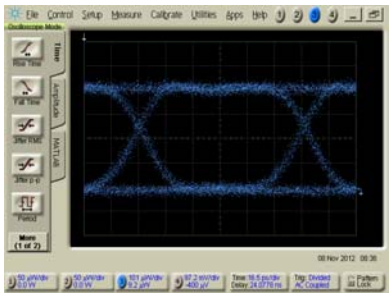
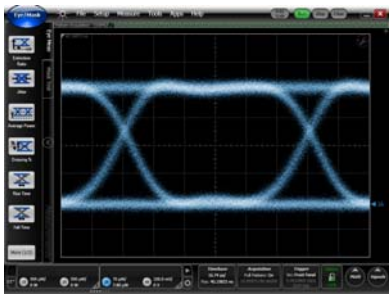
Read the displayed topic and click the links in the sidebar to copy the source code. You'll need *FlexEye_DCA_M.py*, *keyflex.py* (with is library), and *FlexEye_DCA_M_child.py*.

You will need to install at least one DCA-M module with two or more eye diagrams displayed.

5 Converting Legacy Programs

This chapter is written to help those programmers who have a program that controls an 86100D in legacy Graphical User Interface (GUI) and now wish to update their program for an 86100D running FlexDCA. DCA-M instruments cannot be controlled using legacy commands.

Table 14 86100D GUIs

86100D Running Legacy GUI	86100D Running FlexDCA GUI
	

Converting——At a Glance

By converting your program, you'll gain the following benefits:

- Compliant FlexDCA programming commands.
- An actively supported code base.
- Access to powerful, new features that are supported only by FlexDCA.
- Faster measurements in many situations.
- Ability to use new DCA-M instruments, such as the N109X-series DCA-M Sampling Oscilloscopes.

The following list includes some of the differences that you should consider when converting to FlexDCA commands.

- When FlexDCA is installed on a PC, use the **:RDCA:CONNect** command to establish a connection between FlexDCA and the 86100D DCA. Your program will be able to control any DCA-M that is connected to the PC (via USB). However, in this setup (Flex-on-Flex) the program will *not* be able to control a DCA-M that is connected to the 86100D.
- The *OPC common command works with a different set of programming commands as noted in "Command Synchronization (*OPC)" on page 84 and in FlexDCA's help system.
- FlexDCA does not have an equivalent command to legacy's **:DIGitize** root-level command.
- FlexDCA instrument states are saved and recalled from files instead of registers. Use **:DISK:SETUP:SAVE** and **:DISK:SETUP:RECall**.
- Screen images are saved to a file instead of sent to a printer. There is no printer support in FlexDCA. Instead, save the image to a file (**:DISK:SIMage:SAVE**) and use your programming languages support to send the image to the printer.
- Root level commands are dispersed within the **:ACQuire**, **:DISK**, **:STATus**, and **:SYSTem** commands. Refer to Table 24 on page 104.
- Limit test commands in the legacy **:ACQuire** subsystem have been moved to FlexDCA's **:LTESt** subsystem.
- FlexDCA commands do not support creating folders or deleting files and folders, or changing the current working directory. Instead, use the commands that are provided with your programming language.
- When saving a pattern waveform to a file, the entire pattern is always saved. There is no command for saving a range of bits as there is with the legacy commands (**:PWAVeform:RANGe**).
- FlexDCA's math functions do not include the differentiate (**:FUNCtion:DIFF**) or peeling (**:FUNCtion:PEELing**) commands.
- The legacy **:MARKer:REFerence** command specifies the marker reference for TDR and TDT markers, but in FlexDCA, the **:MARKer:REFerence** command specifies the reference marker for delta marker measurements. Although these two commands share the same name, their function is very different.

Conversion Tools

Because FlexDCA responds to a very different set of remote commands than those used to control the 86100D in legacy configuration, Keysight has developed the following two tools within FlexDCA to help you with your conversion:

- SCPI Recorder which records you selections and actions in the FlexDCA GUI and creates the equivalent FlexDCA programming commands. Refer to "Using the SCPI Recorder" on page 48.
- Interactive SCPI Command Tree which gives you a GUI access to all of the commands with the ability to select and enter arguments and query the results. Refer to "Using the Interactive SCPI Command Tree" on page 50.
- Refer to FlexDCA's online help for information on remote control. The help includes a topic for every command.

FlexDCA Measurements Often Require More Commands

Legacy SCPI commands used command arguments to modify the command's behavior. But in FlexDCA, most arguments have been moved to child or sibling commands. This has resulted in an increase in the number of commands that are required to perform some tasks.

For example, consider an ratio eye-width measurement. Legacy code required the following command:

```
:MEASure:CGRade:EWIDth? CHANnel2A RATio
```

The equivalent command in FlexDCA would be:

```
:MEASure:EYE:EWIDth:SOURce CHANnel2A
:MEASure:EYE:EWIDth:FORMat RATio
:MEAsure:EYE:EWIDth?
```

FlexDCA's SCPI Remote Command Tree Viewer makes it easy to view any available sibling commands. Refer to "Using the Interactive SCPI Command Tree" on page 50.

Queries

Command queries in FlexDCA are formatted the same as for the legacy 86100D commands. For example,:

```
:TIMEBASE:SCALE?
:MEASURE:EYE:RISETIME?
```

As with legacy commands, the output queue must be read before the next program message is sent. For example, when you send the query **:MEASURE:EYE:RISETIME?** you must follow it with an input statement. If you send another command or query before reading the result of a query, the output buffer is cleared and the current response is lost. This also generates a query-interrupted error in the error queue. If you execute an input statement before you send a query, it will cause the computer to wait indefinitely.

If a measurement cannot be made because of the lack of data, because the source signal is not displayed, the requested measurement is not possible (for example, a period measurement on an FFT waveform), or for some other reason, 9.99999E+37 is returned as the measurement result.

You can send multiple queries to FlexDCA within a single program message, but you must also read them back within a single program message. This can be accomplished by either reading them back into a string variable or into multiple numeric variables. When you read the result of multiple queries into string variables, each response is separated by a semicolon. For example, the response of the query:

```
:MEASURE:EYE:RISETIME?;FALLTIME?
```

would be:

```
<risetime>;<falltime>
```

Returning Measurement Data

FlexDCA responds to measurement queries differently from the legacy GUI in the following two measurement situations:

- Measurements on eye diagrams (Eye/Mask mode).
- Measurements in Oscilloscope mode with averaging on.

The legacy GUI controlled measurement timing, but FlexDCA is data centered and requires a *OPC?:

When an eye measurement is queried in legacy mode, the legacy GUI waits for a number of data points to be captured, before returning a measurement value. The number of data points is controlled by the legacy **:MEASure:CGRade:COMPlete** command which set the hit count that the color grade database's peak-numbers-of-hits must equal before a measurement is executed. In FlexDCA, there is no such measurement timing control. Instead, when you send a measurement query, a measurement is immediately made and its value returned *regardless* if the waveform data is adequate or not. The returned value could be an incorrect value or even **NaN** (Not a Number).

In FlexDCA, the solution is to use an acquisition limit test followed by an *OPC? to force the program to wait until a specified number of waveforms are captured before making measurements. This is shown in the following lines of Python code:

```
FlexDCA.write(':ACQuire:STOP;:ACQuire:CDISplay')
FlexDCA.write(':ACQuire:ECOunt 100') # Average 100 waveforms
FlexDCA.write(':LTESt:ACQuire:CTYPe:WAVeforms 100') # capture 100 waveforms
FlexDCA.write(':LTESt:ACQuire:CTYPe WAVeforms')
FlexDCA.write(':LTESt:ACQuire:STATe ON')
FlexDCA.query(':ACQuire:RUN;*OPC?') # run limit test
FlexDCA.write(':LTESt:ACQuire:STATe OFF')
FlexDCA.write(':MEASure:EYE:LIST:CLEar')
FlexDCA.write(':MEASure:EYE:AMPLitude') # measure eye amplitude
```

When the command form of a measurement is used, the measurement result is displayed and returned:

```
:MEASure:EYE:EWIDth
```

When the query form of the measurement is used, the measurement is made one time and the measurement result is returned.

```
:MEASure:EYE:EWIDth?
```

If the current acquisition is complete, the measurement is made and the result is returned. If the current acquisition is incomplete and FlexDCA is running, acquisitions will continue to occur until the acquisition is complete. The acquisition will then be measured and the result returned.

If the current acquisition is incomplete and FlexDCA is stopped, the measurement result will be "9.91E+37 (NaN)".

You can also return the measurement results of every measurement that is listed in all displayed measurement results panes. Use the **:MEASure:RESults?** query. The following is an example of three measurements returned in Eye mode. Line breaks have been added to make the string easier to read:

```
Name=Eye Width[Time],Source=1A,Current=9.458E-11,Min=9.436E-11,Max=9.4
58E-11,Count=1.9E+1,Name=Fall Time,Source=1A,Current=2.366E-11,Min=2.3
66E-11,Max=2.366E-11,Count=3.9E+1,Name=Rise Time,Source=1A,Current=2.3
66E-11,Min=2.366E-11,Max=2.366E-11,Count=4.4E+1
```

Command Synchronization (*OPC)

Use the following two commands synchronize command execution with FlexDCA: *OPC (command and query) and *WAI.

Table 15 Commands that are Compatible with *OPC (Sheet 1 of 2)

Legacy	FlexDCA
:ACQuire:LTESt	–
:ACQuire:RUNTil	:ACQuire:RUN (When a limit test is turned on.)
:AUToscale (Jitter mode only)	:SYSTem:AUToscale when FlexDCA is in :ACQuire:RUN mode.
–	:CRECovery:RELock
–	:DIFF:DMODe
–	:DIFF:DSTRacking
:DIGitize	–
–	:DISK:DWIZard:SAVE
–	:DISK:SIMage:SAVE
:DISK:PWAVeform:SAVE	:DISK:WAVeform:SAVE
:MTESt:RUNTil	:ACQuire:RUN
:PRINt	–
–	:FEYE:STATe
–	:MTESt:LOAD
–	
–	:RDCA:CONNect Establishes a connection between FlexDCA and the 86100D DCA.
:SINGle	:ACQuire:SINGle
	:SYSTem:AUToscale When FlexDCA is in :ACQuire:RUN mode.
	:SYSTem:DEFault :SYSTem:FACTory :SYSTem:MODE
	:SYSTem:PAUTodetect When using pattern lock, detects bit rate, pattern length, and trigger divide ratio.
–	:TDR:DESKew:SDONe?
:TIMebase:PRECision	:PTIMebase:STATe ON

Table 15 Commands that are Compatible with *OPC (Sheet 2 of 2)

Legacy	FlexDCA
:TIMebase:PRECision:RFRequency	:TIMebase:PTIMebase:RFRequency
:TIMebase:PRECision:TREFerence	–

The ***OPC** command sets bit 0 in the Standard Status Event Register when all pending device operations have finished. (Use ***ESR?** to query the register.) The ***OPC** command is useful for verifying the completion of commands that could take a variable amount of time or commands executed in parallel with other commands, such as **:SYSTem:AUToscale**. The ***OPC** command does not stop the execution of the remote script.

The ***OPC?** query allows synchronization between the computer and FlexDCA by using the message available (MAV) bit (bit 4 in the Status Byte) or by reading the output queue. (Use ***STB?** to query the Status Byte register.) Unlike the ***OPC** command, the ***OPC?** query does not affect the **OPC** event bit in the Standard Event Status Register. The execution of the remote script is halted and therefore the ***OPC?** query should be used judiciously. For example, running a acquisition limit test with the command string,

```
:LTESt:MTESt:FAILures 100; *OPC?
```

locks the remote interface until 100 failed samples are detected, which could take a very long time. Under these circumstances, you must send a device clear or close to re-start the 86100D.

The ***WAI** command is similar to the ***OPC?** query as it also blocks the execution of the remote script until all pending operations are finished. ***WAI** is particularly useful if the host computer is connected to FlexDCA and one or more instruments. This command will not block the GPIB bus, allowing the computer to continue issuing commands to FlexDCA or an instrument that is not executing the ***WAI** command.

Commands for Defining Waveforms

Defining Eye Diagram Boundaries

```
:MEASure:EBOundary:DEFault
:MEASure:EBOundary:LEFT
:MEASure:EBOundary:RIGHt
```

Defining Custom Threshold Levels

Table 16 Commands to Enter Custom Threshold Levels

Level	**Levels Defined by** Percentage	**Levels Defined by** Unit Amplitude

Upper Level	`:MEASure:THReshold:DISTal`	`:CHANnel{N}:UDEFined:DISTal`
		`:DIFF{1:4}:UDEFined:DISTal`
		`:EMEMory{1:8}:UDEFined:DISTal`
		`:WMEMory{1:8}:UDEFined:DISTal`
		`:FUNCtion{1:16}:UDEFined:DISTal`

Middle Level	`:MEASure:THReshold:MESial`	`:CHANnel{N}:UDEFined:MESial`
		`:DIFF{1:4}:UDEFined:MESial`
		`:EMEMory{1:8}:UDEFined:MESial`
		`:WMEMory{1:8}:UDEFined:MESial`
		`:FUNCtion{1:16}:UDEFined:MESial`

Lower Level	`:MEASure:THReshold:PROXimal`	`:CHANnel{N}:UDEFined:PROXimal`
		`:DIFF{1:4}:UDEFined:PROXimal`
		`:EMEMory{1:8}:UDEFined:PROXimal`
		`:WMEMory{1:8}:UDEFined:PROXimal`
		`:FUNCtion{1:16}:UDEFined:PROXimal`

Defining Waveform Top and Base Levels

Table 17 Commands Used to Enter Top and Base Levels

Waveform Type	**Commands**
Channels	`:CHANnel{N}:UDEFined:BASe` `:CHANnel{N}:UDEFined:TOP`
Differential Waveforms	`:DIFF{1:4}:UDEFined:BASe` `:DIFF{1:4}:UDEFined:TOP`
Eye Memory	`:EMEMory{1:8}:UDEFined:BASe` `:EMEMory{1:8}:UDEFined:TOP`
Waveform Memory	`:WMEMory{1:8}:UDEFined:BASe` `:WMEMory{1:8}:UDEFined:TOP`
Function	`:FUNCtion{1:16}:UDEFined:BASe` `:FUNCtion{1:16}:UDEFined:TOP`

Common Child Commands

As seen in the SCPI Command Tree Viewer, most FlexDCA commands have lower-level commands (children). Some child commands are unique while others are common and repeated under many commands. For example, **:CHANnel:BANDwidth** command has four child commands: **DEFault**, **NEXT**, **PREVious** and **VSET?**.

```
:CHANnel<number>:BANDwidth {HIGH | MID | LOW}
:CHANnel<number>:BANDwidth:DEFault
:CHANnel<number>:BANDwidth:NEXT
:CHANnel<number>:BANDwidth:PREVious
:CHANnel<number>:BANDwidth:VSET?
```

Commands with a parameter argument, generally have **DEFault**, **NEXT**, **PREVious**, and **VSET?** child commands. Commands with a numerical argument, such as **:TIMebase:SCALe**, generally include **DEFault**, **MINimum**, **MAXimum**,and **STEP** child commands.

Common Child Commands for Measurements and Queries

Unless otherwise noted, the following common child commands and queries are available for all oscilloscope, eye, and jitter/noise measurements. The first five queries return the values that are shown in the displayed measurement results tables except *Current*. To return the *Current* measurement value, which is the value for the last waveform acquisition, use the parent command. For example, to return the *Current* Oscilloscope mode risetime measurement, use the **:OSCilloscope:RISetime?** query.

If smoothing is turned on (**:ACQure:SMOothing**), the *Current*, *Mean*, *Minimum*, *Maximum*, and *Std Dev* measurements are an average of all previous waveform or pattern acquisitions and are not available until the terminal acquisition count has been reached (**:ACQuire:ECOunt**). For example, in 16-count smoothing, the *Mean* measurement is reported after the first 16 acquisitions have completed and the displayed *Count* value is then incremented from 0 to 1. Each subsequent acquisition increments the *Count* value.

NOTE **Before these values can be displayed in the results table, the table must be configured in full width mode.**

Common Query for Status Message

The common query for status messages is **:SREason?**. This common query returns the status message that results from sending the parent command. Status messages are shown at the bottom of the FlexDCA's display and typically indicates an error such as incorrect parameter. If the parent command did not result in a message, the query returns a null string.

Table 18 Common Child Commands and Queries

Command	Description
MEAN?	Returns the Mean measurement. It is calculated over the number of waveform acquisitions returned by the COUNt? query.
MIN?	Returns the Minimum measurement. It is calculated over the number of waveform acquisitions returned by the COUNt? query.
MAX?	Returns the Maximum measurement. It is calculated over the number of waveform acquisitions returned by the COUNt? query.
SDEViation?	Returns the standard diviation (Std Dev) measurement, which is only displayed in the result table for Oscilloscope Mode measurements. It is calculated over the number of waveform acquisitions returned by the COUNt? query.
COUNt?	Returns the number of waveform acquisitions (Count) that have occurred for the parent measurement.
LOCation?	Returns the location (index) of the measurement in the displayed results table. This query is available for all Oscilloscope mode, Eye mode, Jitter/Amplitude mode, JSA, mask test, and histograms measurements. The active measurement list is indexed starting at 1. As a result, the first (top most) measurement in the list is identified as 1, the second measurement from the top is identified as 2, and so forth.
REGion	Select a time span over which an oscilloscope measurement is made.
SOURce	This command specifies the source waveform on which to perform the measurement.
STATus?	Queries the status of the measurement. This common child query has its own help topic. The response can be either CORR, INV, or QUES. CORR indicates that the measurement result is correct. INV indicates that the measurement result is invalid. A question mark (?) is shown in the displayed results table instead of a measurement result. QUES indicates that the measurement result is questionable. A question mark (?) is shown in the displayed results table next to the measurement result.
STATus:REASon?	Queries the explanation why the STATus child command indicates an unsuccessful measurement. If the STATus child command indicates a successful measurement, STATus:REASon returns a null string. This common child query has its own help topic.

Commands that Modify the Measurement Results Table

FlexDCA measurement results are shown in a displayed Results table. Use the **:MEASure** subsystem commands shown in Table 19 to modify the table results. the **:LIST** commands identify a listed result by an index number rather than the name of the measurement. The index 1, identifies the first (top most) measurement in the list. The second measurement from the top is identified as 2, and so forth. To get the index value for a specific measurement, you use the **:LOCation?** child command for the targeted measurement. For example:

```
:MEASure:OSCilloscope:FALLtime:LOCation?
```

The `:LIST:CLear` command clears the entire active measurement list in the displayed results table. The `:LIST:DEFault` command for Jitter Mode's **:AMPLitude** and **:JITTer** results resets the results table to show the default active measurement list in the table. The **:LIST:REMove** command removes the selected measurement from the active measurement list, which is displayed in the results table. Use the :LOCation? query for a specific measurement to determine the index of the measurement in the table. The **:LIST:SELect** command removes the selected measurement from the active measurement list, which is displayed in the results table. Use the **:LOCation?** query for a specific measurement to determine the index of the measurement in the table..

Table 19 Commands by Instrument Mode

Mode	Commands
Oscillosope	:MEASure:OSCilloscope:LIST:CLEar :MEASure:OSCilloscope:LIST:REMove :MEASure:OSCilloscope:LIST:SELect
TDR	:MEASure:TDR:LIST:CLEar :MEASure:TDR:LIST:REMove :MEASure:TDR:LIST:SELect
Eye/Mask	:MEASure:EYE:LIST:CLEar :MEASure:EYE:LIST:REMove :MEASure:EYE:LIST:SELect
Jitter Mode (jitter results)	:MEASure:JITTer:LIST:CLEar :MEASure:JITTer:LIST:DEFault :MEASure:JITTer:LIST:REMove :MEASure:JITTer:LIST:SELect
Jitter Mode (noise results)	:MEASure:AMPLitude:LIST:CLEar :MEASure:AMPLitude:LIST:DEFault :MEASure:AMPLitude:LIST:REMove :MEASure:AMPLitude:LIST:SELect

Definite-Length Block Response Data

The definite-length block response data format allows any type of device-dependent data to be transmitted over the system interface as a series of 8-bit binary data bytes. This is particularly useful for sending large quantities of data or 8-bit extended ASCII codes. The syntax is a pound sign (#) followed by a non-zero digit representing the number of following digits (decimal integer) that states the number of 8-bit data bytes being sent. This is followed by the actual data. For example, for transmitting 3000 bytes of data, the syntax would be: #43000<*3000 bytes of data*>. The leftmost 4 represents the number of digits in the number of bytes, and 3000 represents the number of bytes to be transmitted. Byte order can affect the ability of your programs to correctly interpret block data.

Endiannes of Returned Binary Data

To specify the byte order of binary block data that is returned in response to a query, use the **:SYSTem:BORDer** command. This setting applies to both **:WAVeform** and **:MEASure** subsystem data.

NOTE **FlexDCA Revision. The ability to specify endiannes requires FlexDCA revision A.04.00 and above. Prior to revision A.04.00, the endianness of returned block data was always LSB (Least Significant Byte) first.**

:MEASure Subsystem Queries That Return Block Data

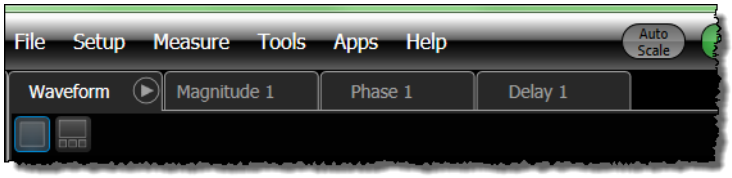
```
:MEASure:AMPLitutde:ISIVsbit?
:MEASure:AMPLitutde:ISIVsbit:BITS?
:MEASure:JITTer:DDJVsbit?
:MEASure:JITTer:DDJVsbit:BITS?
:MEASure:JITTer:EBITs?
:MEASure:JITTer:PATTern?
```

:WAVeform Subsystem Queries That Return Block Data

```
:WAVeform:EYE:INTeger:DAta?
:WAVeform:YFORmat:DOUBle:YDATa?
:WAVeform:YFORmat:FLOat:YDATa?
:WAVeform:YFORmat:WORD:YDATa?
:WAVeform:XYFormat:DOUBle:YDATa?
:WAVeform:XYFormat:DOUBle:XDATa?
:WAVeform:XYFormat:FLOat:YDATa?
:WAVeform:XYFormat:FLOat:XDATa?
```

FlexDCA Waveform Content Windows

Waveform content windows are a FlexDCA feature that is not available in the 86100D legacy GUI. The legacy GUI only supports one content window type that defined the Y-axis in linear amplitude units (V or W) and the X-axis in time. With FlexDCA many different kinds content windows are available to display a wide variety of waveform types.



Depending on the type of waveform, waveforms can be displayed in one of eight different types of content windows. Table 20 on page 93 lists these windows along with the waveform types that they can contain. Table 21 on page 95 lists the available commands and Queries for scaling a content window's axis.

All content windows can show multiple waveforms of the same type. In all content windows except the *Waveform* window, multiple waveforms have the same scaling. In the *Waveform* window, each waveform can have a different Y-axis scaling.

Assigning Waveforms to Content Windows

When created, waveforms are automatically displayed in content window number 1 for the correct type. As shown in the above table, multiple content windows (up to four total) are available for some types. To move a waveform to an additional content window use the SCPI commands shown in the table. If the additional content window does not exist, it will be automatically created.

NOTE

Although there are no direct front-panel controls to create additional Waveform, Freq-Mag, Phase, and GrpDelay content window, you can assign existing waveforms to new windows by using the Interactive SCPI Command Tree.

Axis Scaling

The **:TIMebase**, **:GRAPh**, and **:TRACe** subsystems are used to create and scale content windows and their waveforms as shown in Table 21 on page 95. All time-domain content windows share the same X-axis time scales, which are configured in the **:TIMebase** subsystem. This includes the Waveform, Time-Ohms, Time-Volts, and Time-% content windows. The Y-axis for time-domain content windows and both axis in frequency domain content windows, are all scaled using commands in the **:GRAPh** subsystem.

Traces contain the waveform data that is displayed in graphs. In the **:TRACe** subsystem, the x-axis and y-axis values can only be queried, because they are set by the x-axis and y-axis values of their assigned graph.

Use the **:TRACe** subsystem to perform the following tasks:

- Create a trace.
- Assign a T-domain or S-parameter to the trace.
- Change a trace's color.
- Define the trace amplitude levels for measurements.

Use the **:GRAPh** subsystem to perform the following tasks:

- Scale *both* axis of frequency-domain graphs.
- Scale the Y-axis only of time-domain graphs.

Table 20 Content Window Types (Sheet 1 of 2)

Availability in Instrument Mode					Number of Available Windows		
Scope	Eye	TDR	Jitter	Waveform Types	Via Front Panel	Via Remote Control	SCPI Command Used to Assign Waveform to Content Window
Waveform Window							
●	●	●		Channels Diff. Channels Most Functions Waveform Memory Eye/Mask Memory	1	1	None
Notes: This is the default window for most waveforms. Y-Axis: Volts or Watts (linear scale). X-Axis: seconds.							
Freq-Mag Window							
●	●	●		Output of FFT math function,	1	4	`:FUNCtion<N>:CWINdowÂ MAGNitude<N>`
				TDR/TDT S-Parameters	4	4	`:TRACe<N>:CWINdow MAGNitude<N>`
Notes: Y-Axis: dBV or dBm (logarithmic scale). X-Axis: seconds or Hertz. Use the GRAPh subsystem commands to autoscale, position, and scale waveforms.							
GrpDelay Window							
●	●	●		Output of FFT math function,	1	4	`:FUNCtion<N>:CWINdow GDELay<N>`
				TDR/TDT S-Parameters	4	4	`:TRACe<N>:CWINdow GDELay<N>`
Notes: Y-Axis: seconds. X-Axis: Hertz. Use the GRAPh subsystem commands to autoscale, position, and scale waveforms.							
Phase Window							
●	●	●		Output of FFT math function,	1	4	`:FUNCtion<N>:CWINdow PHASe<N>`
				TDR/TDT S-Parameters	4	4	`:TRACe<N>:CWINdow PHASe<N`
Notes: Y-Axis: phase (°). X-Axis: Hertz. Use the GRAPh subsystem commands to autoscale, position, and scale waveforms.							

Table 20 Content Window Types (Sheet 2 of 2)

Availability in Instrument Mode					Number of Available Windows		
Scope	Eye	TDR	Jitter	Waveform Types	Via Front Panel	Via Remote Control	SCPI Command Used to Assign Waveform to Content Window
Time-Ohms Window							
		●		TDR/TDT T-Domain Responses	4	4	`:TRACe<N>:CWINdowÂ TOHMs<N>`
Notes: Y-Axis: Ohms. X-Axis: Time. Use the GRAPh subsystem commands to autoscale, position, and scale waveforms.							
Time-Volts Window							
		●		TDR/TDT T-Domain Responses	4	4	`:TRACe<N>:CWINdow TVOLts<N>`
Notes: Y-Axis: Volts. X-Axis: Time. Use the GRAPh subsystem commands to autoscale, position, and scale waveforms.							
Time-% Window							
		●		TDR/TDT T-Domain Responses	4	4	`:TRACe<N>:CWINdow TVOLts<N>`
Notes: Y-Axis: %. X-Axis: Time. Use the GRAPh subsystem commands to autoscale, position, and scale waveforms.							
Graphs Window							
			●	Channel and Jitter Data Memory	–	–	None
Notes: Used for display of all Jitter Mode graphs.							

Table 21 Commands and Queries for Axis Scaling (Sheet 1 of 2)

Graph Axis	:TIMebase Subsystem	:GRAPh Subsystem	:TRACe Subsystem
NOTES: Bold Text: Commands that set axis values. Normal Text: Queries that read axis values. Example time values are provided as aid in understanding commands.			
Frequency Domain Content Windows (for S-parameters and FFT operator)			
X-Axis	–	**:X:AUToscale**	–
	–	**:X:STARt (Hz)**	:XLEFt?
	–	**:X:SPAN (Hz)**	–
	–	–	:XSCale?
	–	:X:STOP (Hz)	:XRIGht?
	–	:X:CENTer (Hz)	:XPOSition? (always at center of axis)
Y-Axis	–	**:Y:AUToscale**	–
	–	**:Y:OFFSet**	:Y:OFFSet?
	–	**:Y:SCALe**	:Y:SCALe?
	–	–	:YTOP?
	–	–	:YBOTtom?
Time Domain Content Windows (for time-domain parameters)			
X-Axis	**:POSition (100 ps)** (with timebase reference set to left)	:X:STARt (100 ps)	:XLEFt? (100 ps)
	:XRANge (2000 ps)	:X:SPAN (2000 ps)	–
	:SCALe (200 ps/div)	–	:XSCale? (200 ps/div)
	–	:X:STOP (2100 ps)	:XRIGht? (2100 ps)
	–	:X:CENTer (1100 ps)	:XPOSition? (1100 ps) (always at center of axis)
Y-Axis	–	**:Y:AUToscale**	–
	–	**:Y:OFFSet**	:YOFFset?
	–	**:Y:SCALe**	:YSCale?
	–	–	:YTOP?
	–	–	:YBOTtom?
Waveform Content Window (Channels, Memories, Functions, ...)			

Table 21 Commands and Queries for Axis Scaling (Sheet 2 of 2)

Graph Axis	:TIMebase Subsystem	:GRAPh Subsystem	:TRACe Subsystem
X-Axis	**:POSition (100 ps)** (with timebase reference set to left)	–	–
	:XRANge (2000 ps)	–	–
	:SCALe (200 ps/div)	–	–
Y-Axis	Y-axis values set in **:CHANnel**, **:DIFF**, **:WMEMory**, and etc subsystems.		

Unavailable :CWINDow Commands

The following :CWINdow commands are reserved for future use with Waveform content windows. Since only one Waveform content window is available at this time, currently there is no need to use these commands.

```
:CHANnel<N>:CWINdow
:DIFF<N>:CWINdow
:EMEMory<N>:CWINdow
:JDMemory<N>:CWINdow
:WMEMory<N>:CWINdow
```

Example of Using Content Windows

This example creates four FFT functions and assigns the output of functions 1 and 2 to Group Delay content window 1 and functions 3 and 4 to Group Delay window 2.

```
  // Configure first FFT function.
:FUNCtion1:FOPerator FFT  // Create an FFT function.
:FUNCtion1:OPERand1 CHAN1A  // Assign input waveform.
:SPRocess1:FFT:DISPlay GDELay  // Select output waveform type.
:FUNCtion1:CWINdow GDELay1  // Assign output waveform to content
window.
  // Configure second FFT function.
:FUNCtion2:FOPerator FFT
:FUNCtion2:OPERand1 CHAN1B
:SPRocess2:FFT:DISPlay GDELay
:FUNCtion2:CWINdow GDELay1
  // Configure third FFT function.
:FUNCtion3:FOPerator FFT
:FUNCtion3:OPERand1 CHAN2A
:SPRocess3:FFT:DISPlay GDELay
:FUNCtion3:CWINdow GDELay2
  // Configure fourth FFT function.
:FUNCtion4:FOPerator FFT
:FUNCtion4:OPERand1 CHAN2B
:SPRocess4:FFT:DISPlay GDELay
:FUNCtion4:CWINdow GDELay2
// Configure content windows to be zoom tiled.
:DISPlay:WINDow:GDELay1:DMODe ZTILe
```

```
:DISPlay:WINDow:GDELay2:DMODe ZTILe
// Select the waveform to be displayed in the zoom-tiled Group Delay
content window.
:DISPlay:WINDow:GDELay1:ZSIGnal FUNCtion1  // View function 1 waveform.
:DISPlay:WINDow:GDELay2:ZSIGnal FUNCtion3  // View function 3 waveform.
// Autoscale waveforms in each Group Delay content window.
:GRAPh:GDELay1:AUToscale
:GRAPh:GDELay1:AUToscale
```

FlexDCA Commands Specific to Instrument Options

Option 500, Rapid Eye Commands

```
:ACQuire:REYE
:ACQuire:REYE:INTerval
```

Option IRC Commands

```
:CHANnel:SIRC
:CHANnel:SIRC:AUPDates
:CHANnel:SIRC:CFOLder
:CHANnel:SIRC:CFUPdate
:CHANnel:SIRC:CSTatus?
:CHANnel:SIRC:CSTatus:REASon?
:CHANnel:SIRC:DSTore
:CHANnel:SIRC:FBANdwidth
:CHANnel:SIRC:FRATe
:CHANnel:SIRC:USTatus?
:CHANnel:SIRC:USTatus:REASon?
```

Option SIM Commands

```
:SPRocess:CONVolve:FMODel
:SPRocess:CONVolve:FNAMe
:SPRocess:CONVolve:FRELoad
:SPRocess:CONVolve:PEXTract
:SPRocess:CONVolve:PORDer
:SPRocess:DCONvolve:DRESult
:SPRocess:DCONvolve:FMODel
:SPRocess:DCONvolve:FNAMe
:SPRocess:DCONvolve:FRELoad
:SPRocess:DDEConvolve:BANDwidth
:SPRocess:DDEConvolve:BANDwidth:AUTo
:SPRocess:DDEConvolve:BLIMit
:SPRocess:DDEConvolve:DRESult
:SPRocess:DDEConvolve:FMODel
:SPRocess:DDEConvolve:FNAMe
:SPRocess:DDEConvolve:FRELoad
:SPRocess:DDEConvolve:PORDer
:SPRocess:DDEMbed
:SPRocess:DDEMbed:BANDwidth
:SPRocess:DDEMbed:BANDwidth:AUTo
:SPRocess:DDEMbed:BLIMit
:SPRocess:DDEMbed:FNAMe
:SPRocess:DEConvolve:BANDwidth
:SPRocess:DEConvolve:BANDwidth:AUTo
:SPRocess:DEConvolve:BLIMit
:SPRocess:DEConvolve:FMODel
:SPRocess:DEConvolve:FNAMe
:SPRocess:DEConvolve:FRELoad
:SPRocess:DEConvolve:PEXTract
:SPRocess:DEConvolve:PORDer
:SPRocess:DEMBed:BANDwidth
```

```
:SPRocess:DEMBed:BANDwidth:AUTo
:SPRocess:DEMBed:BLIMit
:SPRocess:DEMBed:FNAMe
```

Option JSA Commands

```
:CRECovery:JSANalysis:ACQuire
:CRECovery:JSANalysis:FFTMagnitude:ECOunt
:CRECovery:JSANalysis:FFTMagnitudeSMOothing
:CRECovery:JSANalysis:INTegrate:F
:CRECovery:JSANalysis:OPTimize:EMBedded
:CRECovery:JSANalysis:OPTimize:INPut
:CRECovery:JSANalysis:PLLorder:FGAIn
:CRECovery:JSANalysis:PLLorder:FPOLe
:CRECovery:JSANalysis:PLLorder:FZERo
:CRECovery:JSANalysis:PLLorder:JTF:BWIDth
:CRECovery:JSANalysis:PLLorder:JTF:PEAKing
:CRECovery:JSANalysis:PLLorder:MODE
:CRECovery:JSANalysis:PLLorder:OJTF:BWIDth
:CRECovery:JSANalysis:PLLorder:OJTF:DFACtor
:CRECovery:JSANalysis:PLLorder:ORDer
:CRECovery:JSANalysis:RJCorrection
:CRECovery:JSANalysis:SPECtrum
:CRECovery:JSANalysis:SPECtrum:ASCii:YDATa?
:CRECovery:JSANalysis:SPECtrum:DOUBle:YDATa?
:CRECovery:JSANalysis:SPECtrum:FLOat:YDATa?
:CRECovery:JSANalysis:SPECtrum:PEAKs:ALL?
:CRECovery:JSANalysis:SPECtrum:POINts?
:CRECovery:JSANalysis:SPECtrum:XINCrement?
:CRECovery:JSANalysis:SPECtrum:XORigin?
:MEASure:CRECovery:DJ
:MEASure:CRECovery:DJ:LOCation?
:MEASure:CRECovery:DJ:STATus?
:MEASure:CRECovery:DJ:STATus:REASon?
:MEASure:CRECovery:RJ
:MEASure:CRECovery:RJ:LOCation?
:MEASure:CRECovery:RJ:STATus?
:MEASure:CRECovery:RJ:STATus:REASon?
:MEASure:CRECovery:TJ
:MEASure:CRECovery:TJ:LOCation?
:MEASure:CRECovery:TJ:STATus?
:MEASure:CRECovery:TJ:STATus:REASon?
:MEASure:JITTer:RJ
:DISPlay:JSANalysis:SCRaph
:LTESt:MEASure:CRECovery:LOCation:FAILures
:LTESt:MEASure:CRECovery:LOCation:FREGion
:LTESt:MEASure:CRECovery:LOCation:LLIMit
:LTESt:MEASure:CRECovery:LOCation:STATe
:LTESt:MEASure:CRECovery:LOCation:ULIMit
:LTESt:MEASure:CRECovery:LOCation:UPACtion
```

Comparison of Command Subsystems

Table 22 shows the differences between the subsystems for the legacy commands and the subsystems for FlexDCA's commands. Refer to "Legacy-to-FlexDCA Command Subsystems" on page 103 for tables that compare the individual commands.

Table 22 Legacy Versus FlexDCA Subsystems (Sheet 1 of 3)

Legacy Subsystem	FlexDCA Subsystem
Common Commands*	Common commands are documented in Table 23 on page 103.
Root Level Commands†	There are no FlexDCA root level commands. Instead similar commands are dispersed within the :ACQuire, :DISK, :STATus, and :SYSTem commands. Refer to Table 24 on page 104.
`:ACQuire`	:ACQuire (Refer to Table 25 on page 106.) :LTESt (Refer to Table 34 on page 116.)
`:CALibrate`	:CALibrate (Refer to Table 26 on page 107.)
`:CHANnel`	:CHANnel (Refer to Table 27 on page 108.)
–	:CMODe This subsystem controls common-mode waveforms in TDR/TDT mode.
`:CRECovery`	:CRECovery (Refer to Table 28 on page 109.)
–	:DIFF This subsystem controls differential electrical waveforms.
`:DISK`	:DISK (Refer to Table 29 on page 110.) There are no commands for creating folders or deleting files and folders, or changing the current working directory. Instead, use the commands that are provided with your programming language.
`:DISPlay`	:DISPlay (Refer to Table 30 on page 111.)
–	:EMEMory This subsystem loads and displays gray-scale and color-grade database waveforms into one of eight eye memories.
–	:EMODules This subsystem assigns extended modules to internal DCA slots and simulated slots.
`:FUNCtion`	:FUNCtion (Refer to Table 31 on page 113.) All legacy functions are supported except differentiate (:DIFF) and (:PEELing).
–	:GRAPh This subsystem controls waveform scaling and position (with some exceptions) for frequency and time domain content windows.
`:HARDcopy`	There is no :HARDcopy subsystem in FlexDCA. Refer to Table 32 on page 114. Use the :DISK subsystem to save an screen image to a graphics file (:DISK:SIMage:SAVE) or as a Documentation Wizard file (:DISK:DWIZard:SAVE), which is a comprehensive zip file for which you can specify a broad range of contents.
`:HISTogram`	:HISTogram

Table 22 Legacy Versus FlexDCA Subsystems (Sheet 2 of 3)

Legacy Subsystem	FlexDCA Subsystem
–	`:JDMemory` This subsystem allows you to use jitter data offline.
–	`:JSAMemory` This subsystem allows you to view, use, or query the data within the JSA Memory database.
–	`LLINe` This subsystem to load and display a Limit Line Test file in oscilloscope or TDR modes.
`:LTESt`	:LTESt (Refer to Table 34 on page 116.)
`:MARKer`	:MARKer (Refer to Table 35 on page 118.)
`:MEASure`	:MEASure (Refer to Table 37 on page 121.)
`:MTESt`	:MTESt (Refer to Table 36 on page 119.)
–	`:PTIMebase` This subsystem controls features available for the 86107A precision timebase module and 86108A Precision Waveform Analyzer module.
–	`:RDCA` This subsystem controls the remote connection between N1010A FlexDCA running on a PC and the 86100D.
–	`:SLOT` This subsystem controls the trigger setting of a hardware extended modules that have triggering capability. For example, the N1090A DCA-M single optical channel sampling oscilloscope.
–	`:SOURce` This subsystem configures the source waveforms of simulated extended modules.
`:SPARameter`	
`:SPRocessing`	:SPRocess. The name of the signal processing subsystem has been shortened. (Refer to Table 38 on page 129.)
–	:STATus (Refer to Table 24 on page 104.) The Status subsystem commands are used to detect when a specific event has occurred. Many of these commands were root-level legacy commands.
`:SYSTem`	:SYSTem (Refer to Table 39 on page 130.)
`:TDR`	:TDR (Refer to Table 40 on page 131.)
`:TIMebase`	:TIMebase (Refer to Table 41 on page 132.)
–	`:TRACe` Use the TRACe subsystem commands in TDR/TDT Mode to create traces for displaying T-domain parameters and S-parameters. This includes specifying the waveform content window in which to display the trace. Up to 128 traces can be defined. Refer to "FlexDCA Waveform Content Windows" on page 91 to learn about content windows.
`:TRIGger`	:TRIGger (Refer to Table 42 on page 133.)
`:WAVeform`	:WAVeform (Refer to Table 43 on page 134.)

Table 22 Legacy Versus FlexDCA Subsystems (Sheet 3 of 3)

Legacy Subsystem	FlexDCA Subsystem
`:WMEMory`	:WMEMory (Refer to Table 44 on page 135.)

* Common commands do not reside in any subsystem and begin with an asterisk (*) character.

† Root-level commands do not reside in any subsystem and do *not* begin with an asterisk.

Legacy-to-FlexDCA Command Subsystems

This section contains tables that compare legacy / FlexDCA commands by SCPI subsystem. Look up your legacy command in these tables and a you'll find an equivalent or similar FlexDCA command, if available. In many cases, there is not a one-to-one correspondence between 86100D and FlexDCA commands. For some commands, the FlexDCA version may only provide a query form. There are no root-level commands in FlexDCA. Instead the equivalent commands are dispersed within subsystems as listed in the table. DCA-M instruments cannot be controlled using legacy commands.

Table 22 on page 100 lists all of the command *subsystems* in legacy and in FlexDCA. There are additional subsystems in FlexDCA that are not in legacy.

Table 23 Common Commands

86100D Legacy Command	Similar FlexDCA Command
*CLS	*CLS
*ESE	*ESE
*ESR?	*ESR?
*IDN?	*IDN?
*LRN?	:DISK:SETUP:SAVE The instrument setup data is saved to a file rather than returned to the controller.
*OPC	*OPC
*OPT?	*OPT?
*RCL	:DISK:SETUP:RECall The instrument setup data is recalled from a file rather than a register. Instrument setups are not saved to registers in FlexDCA.
*RST	*RST or :SYSTem:DEFault
*SAV	:DISK:SETUP:SAVE The instrument setup data is saved to a file rather than saved to a register. Instrument setups are not saved to registers in FlexDCA.
*SRE	*SRE
*STB?	*STB?
*TRG	*TRG
*TST?	—
*WAI	*WAI

Table 24 Root Level Commands (Sheet 1 of 2)

86100D Legacy Command	Similar FlexDCA Command
AEEN	:STATus:ACQuisition:ENABle
ALER?	:STATus:ACQuisition:EVENt?
AUToscale	:SYSTem:AUToscale;*OPC?
BLANk	:SYSTem:BLANk
CDISplay	:ACQuire:CDISplay
COMMents	–
CREE	:STATus:CRECovery:ENABle
CRER?	:STATus:CRECovery:EVENt
DIGitize	–
JEE	:STATus:JITTer:ENABle
JER?	:STATus:JITTer:EVENt
LER?	:STATus:LOCal:EVENt
LTEE	:STATus:LTESt:ENABle
LTER?	:STATus:LTESt:EVENt
MODel?	:SYSTem:MODel?
MTEE	:STATus:MTESt:ENABle
MTER?	:STATus:MTESt:EVENt
OPEE	:STATus:OPERation:ENABle
OPER?	:STATus:OPERation:EVENt
PTEE	:STATus:PTIMebase:ENABle
PTER?	:STATus:PTIMebase:EVENt
PRINt	FlexDCA has no print capability. Save screen images to a file (:DISK:SIMage:SAVE) and then print the file.
RECall:SETup	:DISK:SETup:RECall
RUN	:ACQuire:RUN
SERial	:SYSTem:SERial? {FRAMe | SLOT{1:8}}
SINGle	:ACQuire:SINGle
STOP	:ACQuire:STOP
STORe:SETup	:DISK:SETup:SAVE
STORe:WAVeform	:DISK:WAVeform:SAVE

Table 24 Root Level Commands (Sheet 2 of 2)

86100D Legacy Command	Similar FlexDCA Command
TER?	:STATus:PTIMebase:EVENt
UEE	:STATus:USER:ENABle
UER?	:STATus:USER:EVENt
VIEW	:SYSTem:VIEW

Table 25 :ACQuire Commands

86100D Legacy Command	Similar FlexDCA Command
:ACQuire:AVERage	:ACQuire:SMOothing
:ACQuire:BEST	:ACQuire:FEEDthru
:ACQuire:COUNt	:ACQuire:ECOunt
:ACQuire:EYELine	:ACQuire:WRAPping
:ACQuire:LTESt	:LTESt:ACQuire:SIMage:STATe :LTESt:ACQuire:SWAVeform:CHANnel{1A:8D}:STATe :LTESt:ACQuire:SWAVeform:CMODe{1A:8C}:STATe :LTESt:ACQuire:SWAVeform:DIFF{1A:8C}:STATe :LTESt:ACQuire:SWAVeform:FUNCtion{1:16}:STATe
:ACQuire:POINts	:ACQuire:RLENgth
:ACQuire:POINts Auto	:ACQuire:RLENgth:MODE AUTomatic
:ACQuire:RUNTil	:LTESt:ACQuire:CTYPe:PATTerns :LTESt:ACQuire:CTYPe:SAMPles :LTESt:ACQuire:CTYPe:WAVeforms :LTESt:ACQuire:STATe ON;*OPC?
:ACQuire:SSCReen	:LTESt:ACQuire:SIMage:STATe
:ACQuire:SSCReen:AREA	:LTESt:ACQuire:SIMage:GONLy
:ACQuire:SSCReen:IMAGe	:LTESt:ACQuire:SIMage:INVert :LTESt:ACQuire:SIMage:MONochrome
:ACQuire:SWAVeform	:LTESt:ACQuire:SWAVeform:CHAN:STATe :LTESt:ACQuire:SWAVeform:CMODe:STATe :LTESt:ACQuire:SWAVeform:DIFF:STATe :LTESt:ACQuire:SWAVeform:FUNCtion:STATe
:ACQuire:SWAVeform:RESet	:LTESt:ACQuire:SWAVeform:RESet

Table 26 :CALibrate Commands (Sheet 1 of 2)

86100D Legacy Command	Similar FlexDCA Command
:CALibrate:CANCel	:CALibrate:CANCel
:CALibrate:CONTinue	:CALibrate:CONTinue
:CALibrate:ERATio:DLEVel?	:CALibrate:DARK:CHANnel{1A:8D}:STARt
:CALibrate:ERATio:STARt	:CALibrate:DARK:CHANnel{1A:8D}:STARt
:CALibrate:ERATio:STATus?	:CALibrate:DARK:CHANnel{1A:8D}STATus?
:CALibrate:FRAMe:LABel	–
:CALibrate:FRAMe:STARt	:CALibrate:FRAMe:PTIMebase:STARt For 86100D Option PTB precision timebase only.
:CALibrate:FRAMe:TIME?	:CALibrate:FRAMe:TIMebase:STATus:TIME?
:CALibrate:MODule:LRESistance	:CALibrate:LRESistance
:CALibrate:MODule:OCONversion?	:CALibrate:OPTical:CONVersion?
:CALibrate:MODule:OPOWer	:CALibrate:OPTical:USER:CHANnel{1A:8D}:OPOWer For N1090A DCA-M modules only.
:CALibrate:MODule:OPTical	:CALibrate:OPTical:USER:CHANnel{1A:8D}:STARt For N1090A DCA-M modules only.
:CALibrate:MODule:OWAVelength	:CALibrate:OPTical:USER:CHANnel{1A:8D}:WAVelength For N1090A DCA-M modules only.
:CALibrate:MODule:STATus?	:CALibrate:SLOT{1:8}:STATus?
:CALibrate:MODule:TIME?	:CALibrate:CHANnel{1A:8D}:STATus:DETails? :CALibrate:CHANnel{1A:8D}:STATus:TIME? :CALibrate:CHANnel{1A:8D}:STATus:DTEMperature? :CALibrate:SLOT{1:8}:STATus:DETails? :CALibrate:SLOT{1:8}:STATus:TIME? :CALibrate:SLOT{1:8}:STATus:DTEMperature?
:CALibrate:MODule:VERTical	:CALibrate:SLOT{1:8}:STARt
:CALibrate:OUTPut	:CALibrate:OUTPut
:CALibrate:PROBe	:CALibrate:PROBe:CHANnel{1A:8D}:STARt
:CALibrate:RECommend?	:CALibrate:SLOT{1:8}:CRECovery:STATus? :CALibrate:DARK:CHANnel{1A:8D}:STATus? :CALibrate:SLOT{1:8}:STATus? :CALibrate:SLOT{1:8}:VERTical:STATus?
:CALibrate:SAMPlers	–
:CALibrate:SDONe?	:CALibrate:SDONe?
:CALibrate:SKEW	:CALibrate:SKEW:SLOT{1:8}:STARt
:CALibrate:SKEW:AUTO	–

Table 26 :CALibrate Commands (Sheet 2 of 2)

86100D Legacy Command	Similar FlexDCA Command
:CALibrate:STATus?	:CALibrate:SLOT:STATus?

Table 27 :CHANnel Commands

86100D Legacy Command	Similar FlexDCA Command
:CHANnel:BANDwidth	:CHANnel{1A:8D}:BANDwidth
:CHANnel:CONNector	:CHANnel{1A:8D}:CONNector
:CHANnel:DISPlay	:CHANnel{1A:8D}:DISPlay
:CHANnel:DSKew	:CHANnel{1A:8D}:SKEW Manual skew entry for N1045A and 86118A-H01 modules.
:CHANnel:DSKew:AUTO	:CALibrate:SKEW:SLOT{1:8}:STARt Automatic deskew for N1045A module.
:CHANnel:DSKew:AUTO:STEP	–
:CHANnel:DSKew:LCALibrate	–
:CHANnel:FDEScription?	:CHANnel{1A:8D}:FSELect:RATe:VSET?
:CHANnel:FILTer	:CHANnel{1A:8D}:FILTer FILTer{1:12}
:CHANnel:FSELect	:CHANnel{1A:8D}:FILTer FILTer{1:12}
:CHANnel:OFFSet	:CHANnel{1A:8D}:YOFFset
:CHANnel:PROBe	:CHANnel{1A:8D}:ATTenuator:STATe :CHANnel{1A:8D}:TRANsducer:STATe
:CHANnel:PROBe:CALibrate	:CALibrate:PROBe:CHANnel{1A:8D}:STARt
:CHANnel:PROBe:SELect	:CHANnel{1A:8D}:PROBe:ID
:CHANnel:RANGe	–
:CHANnel:SCALe	:CHANnel{1A:8D}:YSCale
:CHANnel:TDRSkew	:TDR:DESKew:SCHannel
:CHANnel:UNITs	:CHANnel{1A:8D}:TRANsducer:UNITs
:CHANnel:UNITs:ATTenuation	:CHANnel{1A:8D}:TRANsducer:GAIN
:CHANnel:UNITs:OFFSet	:CHANnel{1A:8D}:TRANsducer:OFFSet
:CHANnel:WAVelength	:CHANnel{1A:8D}:WAVelength

Table 28 :CRECovery Commands

86100D Legacy Command	Similar FlexDCA Command
:CRECovery:ARELock	–
:CRECovery:ARELock:CANCel	–
:CRECovery:ARELock:STATe?	–
:CRECovery:CFRequency?	:CRECovery:CFRequency?
:CRECovery:CLBandwidth	:CRECovery:CLBandwidth
:CRECovery:CRATe	:CRECovery:CRATe
:CRECovery:INPut	:CRECovery:SOURce
:CRECovery:LBANdwidth	:CRECovery:LBANdwidth
:CRECovery:LBWMode	:CRECovery:LBWMode
:CRECovery:LOCKed?	:CRECovery:LOCKed?
:CRECovery:LSELect	:CRECovery:LSELect
:CRECovery:LSELect:AUTomatic	:CRECovery:LSELect:AUTomatic
:CRECovery:ODRatio	:CRECovery:ODRatio
:CRECovery:ODRatio:AUTO	:CRECovery:ODRatio:AUTomatic
:CRECovery:PEAKing?	:CRECovery:JSANalysis:PLLoop:JTF:PEAKing?
:CRECovery:RATE	:CRECovery:RATe
:CRECovery:RDIVider	:CRECovery:RDIVider
:CRECovery:RELock	:CRECovery:RELock
:CRECovery:SPResent?	–
:CRECovery:TDENsity?	:CRECovery:TDENsity?
:CRECovery:T2TFrequency?	–

Table 29 :DISK Commands

86100D Legacy Command	Similar FlexDCA Command
:DISK:BFILe?	:DISK:FILE:READ
:DISK:CDIRectory	–
:DISK:DELete	–
:DISK:DIRectory?	–
:DISK:LOAD	:DISK:EYE:RECall :DISK:JDATabase:RECall :DISK:WAVeform:RECall :DISK:SETup:RECall
:DISK:MDIRectory	–
:DISK:PWAVeform:LOAD	:DISK:WAVeform:RECall
:DISK:PWAVeform:PPBit	:ACQuire:SPBit :ACQuire:SPBit:MODe
:DISK:PWAVeform:RANGe	:TIMebase:BRANge
:DISK:PWAVeform:RANGe:STARt	:TIMebase:BPOSition
:DISK:PWAVeform:RANGe:STOP	:TIMebase:BRANge
:DISK:PWAVeform:SAVE	:DISK:WAVeform:SAVE For pattern waveforms, the entire waveform is always saved; a range of bits cannot be specified.
:DISK:PWD?	Use the commands that are provided with your programming language.
:DISK:SIMage	:DISK:SIMage:SAVE
:DISK:SPARameter:SAVE	:DISK:SPARameter:PSET :DISK:SPARameter:SAVE
:DISK:STORe	:DISK:EYE:SAVE :DISK:JDATabase:SAVE :DISK:WAVeform:SAVE :DISK:SETup:SAVe
:DISK:TFILe?	:DISK:FILE:ASCii:READ

Table 30 :DISPlay Commands (Sheet 1 of 2)

86100D Legacy Command	Similar FlexDCA Command
:DISPlay:CGRade:LEVels?	–
:DISPlay:CONNect	:DISPlay:CONNect
:DISPlay:DATA?	:DISK:SIMage:FNAMe :DISK:SIMage:SAVE;*OPC? :DISK:BFIL?
:DISPlay:DCOLor	:SYSTem:DEFault :DISPlay:GRATicule:GRID:INTensity
:DISPlay:ETUNing	:DISPlay:ETUNing
:DISPlay:GRATicule	:DISPlay:GRATicule:GRID:STATe :DISPlay:GRATicule:HDIVisions :DISPlay:GRATicule:VDIVisions
:DISPlay:JITTer:BATHtub:YSCale	:DISPlay:JITTer:BATHtub:YSCale
:DISPlay:JITTer:GRAPh	:DISPlay:JITTer:GRAPh:TYPE
:DISPlay:JITTer:HISTogram:YSCale	:DISPlay:JITTer:HISTogram:YSCale
:DISPlay:JITTer:LAYout	:DISPlay:JITTer:LAYout
:DISPlay:JITTer:PJWFrequency	–
:DISPlay:JITTer:PJWTracking	–
:DISPlay:JITTer:SHADe	–
:DISPlay:LABel	–
:DISPlay:LABel:DALL	–
:DISPlay:PERSistence	:DISPlay:PERSistence
:DISPlay:RRATe	–
:DISPlay:SCOLor	:CHANnel:COLor :DIFF:COLor :EMEMory:COLor :FUNCtion:COLor :WMEMory:COLor
:DISPlay:SINTegrity:BATHtub:YSCale	:DISPlay:JITTer:BATHtub:YSCale
:DISPlay:SINTegrity:GRAPh	:DISPlay:JITTer:GRAPh
:DISPlay:SINTegrity:HISTogram:YSCale	:DISPlay:JITTer:HISTogram:YSCale
:DISPlay:SINTegrity:LAYout	:DISPlay:JITTer:LAYout
:DISPlay:SINTegrity:LEVel	:DISPlay:AMPLitude:LEVel
:DISPlay:SINTegrity:SHADe	–

Table 30 :DISPlay Commands (Sheet 2 of 2)

86100D Legacy Command	Similar FlexDCA Command
:DISPlay:SPARameter:GRAPh	When a waveform content window is in zoom-tiled mode, selects the waveform to display. Use the following commands to select the waveform to display: :DISPlay:WINDow:GDELay:ZSIGnal :DISPlay:WINDow:MAGNitude:ZSIGnal :DISPlay:WINDow:PHASe:ZSIGnal :DISPlay:WINDow:T:OHMS:ZSIGnal :DISPlay:WINDow:T:PERCent:ZSIGnal :DISPlay:WINDow:T:VOLTs:ZSIGnal :DISPlay:WINDow:TIME:ZSIGnal
:DISPlay:SPARameter:LAYout	Places a waveform content window in zoom-tiled mode: :DISPlay:WINDow:GDELay:DMODe :DISPlay:WINDow:MAGNitude:DMODe :DISPlay:WINDow:PHASe:DMODe :DISPlay:WINDow:T:OHMS:DMODe :DISPlay:WINDow:T:PERCent:DMODe :DISPlay:WINDow:T:VOLTs:DMODe :DISPlay:WINDow:TIME:DMODe
:DISPlay:SPARameter:SHADe	–
:DISPlay:SSAVer	–

Table 31 :FUNCtion Commands

86100D Legacy Command	Similar FlexDCA Command
:FUNCtion:ADD	:FUNCtion:FOPerator ADD
:FUNCtion:DIFF	:FUNCtion:FOPerator DIFFerence
:FUNCtion:DISPlay	:FUNCtion:DISPlay
:FUNCtion:FUNCtion	:FUNCtion:OPERand
:FUNCtion:HORizontal	:FUNCtion:HTRacking
:FUNCtion:HORizontal:POSition	:FUNCtion:XPOSition
:FUNCtion:HORizontal:RANGe	:FUNCtion:XSCale
:FUNCtion:INVert	:FUNCtion:FOPerator INVert
:FUNCtion:MAGNify	:FUNCtion:FOPerator AMPLify
:FUNCtion:MAXimum	:FUNCtion:FOPerator MAX
:FUNCtion:MINimum	:FUNCtion:FOPerator MIN
:FUNCtion:MULTiply	:FUNCtion:FOPerator MULTiply
:FUNCtion:OFFSet	:FUNCtion:FOPerator ADD :FUNCtion:OPERand CONStant :FUNCtion:CONStant
:FUNCtion:PEELing	–
:FUNCtion:RANGe	:FUNCtion:YSCale
:FUNCtion:SUBTract	:FUNCtion:FOPerator SUBTract
:FUNCtion:VERSus	:FUNCtion:FOPerator VERSus
:FUNCtion:VERTical	:FUNCtion:VTRacking
:FUNCtion:VERTical:OFFSet	:FUNCtion:OFFSet
:FUNCtion:VERTical:RANGe	:FUNCtion:YSCale

Table 32 :HARDcopy Commands

86100D Legacy Command	Similar FlexDCA Command
FlexDCA does not have a :HARDcopy subsystem. Instead, use the :DISK commands shown in this table.	
:HARDcopy:AREA	:DISK:SIMage:WINDow :DISK:SIMage:SAVE (Saves screen image to graphics file.) :DISK:DWIZard:SAVE (Saves screen image and other information to a zip file. Refer to the Documentation Wizard in FlexDCA's help system.)
:HARDcopy:DPRinter	–
:HARDcopy:FACTors	:DISK:DWIZard:SSETup Saves instrument state file to a zip file. Refer to the Documentation Wizard in FlexDCA's help system.
:HARDcopy:IMAGe	:DISK:DWIZard:INVert :DISK:DWIZard:MONochrome :DISK:SIMage:SAVE (Saves screen image to graphcs file.)
:HARDcopy:PRINTers?	–

Table 33 :HISTogram Commands

86100D Legacy Command	Similar FlexDCA Command
:HISTogram:AXIS	:HISTogram:AXIS
:HISTogram:MODE	:HISTogram:DISPlay
:HISTogram:SCALe:SIZE	:HISTogram:HSIZe :HISTogram:VSIZe
:HISTogram:SOURce	:HISTogram:SOURce
:HISTogram:WINDow:BORDer	:HISTogram:WINDow:BORDer
:HISTogram:WINDow:DEFault	:HISTogram:WINDow:DEFault
:HISTogram:WINDow:SOURce	:HISTogram:SOURce
:HISTogram:WINDow:X1Position	:HISTogram:WINDow:X1
:HISTogram:WINDow:X2Position	:HISTogram:WINDow:X2
:HISTogram:WINDow:Y1Position	:HISTogram:WINDow:Y1
:HISTogram:WINDow:Y2Position	:HISTogram:WINDow:Y2

Table 34 :LTESt Commands (Sheet 1 of 2)

86100D Legacy Command	Similar FlexDCA Command
:LTESt:FAIL	:LTESt:MEASure:AMPLitude:LOCation:FREGion :LTESt:MEASure:EYE:LOCation:FREGion :LTESt:MEASure:JITTer:LOCation:FREGion :LTESt:MEASure:MTESt:LOCation:FREGion :LTESt:MEASure:OSCilloscope:LOCation:FREGion
:LTESt:JITTer:SELect	:LTESt:MEASure:JITTer:LOCation:STATe
:LTESt:LLIMit	:LTESt:MEASure:AMPLitude:LOCation:LLIMit :LTESt:MEASure:EYE:LOCation:LLIMit :LTESt:MEASure:JITTer:LOCation:LLIMit :LTESt:MEASure:MTESt:LOCation:LLIMit :LTESt:MEASure:OSCilloscope:LOCation:LLIMit
:LTESt:MNFound	:LTESt:MEASure:AMPLitude:LOCation:UPACtion :LTESt:MEASure:EYE:LOCation:UPACtion :LTESt:MEASure:JITTer:LOCation:UPACtion :LTESt:MEASure:MTESt:LOCation:UPACtion :LTESt:MEASure:OSCilloscope:LOCation:UPACtion
:LTESt:RUNTil	:LTESt:MEASure:AMPLitude:LOCation:FAILures :LTESt:MEASure:EYE:LOCation:FAILures :LTESt:MEASure:JITTer:LOCation:FAILures :LTESt:MEASure:MTESt:LOCation:FAILures :LTESt:MEASure:OSCilloscope:LOCation:FAILures
:LTESt:SINTegrity:SELect	:LTESt:MEASure:JITTer:LOCation:STATe :LTESt:MEASure:AMPLitude:LOCation:STATe
:LTESt:SOURce	Read the :LTESt introduction.
:LTESt:SSCReen	:LTESt:MEASure:SIMage:STATe
:LTESt:SSCReen:AREA	:LTESt:MEASure:SIMage:GONLy
:LTESt:SSCReen:IMAGe	:LTESt:MEASure:SIMage:INVert
:LTESt:SSUMmary	:LTESt:MEASure:SSUMmary:STATe
:LTESt:SWAVeform	:LTESt:MEASure:SWAVeform:CHAN:STATe :LTESt:MEASure:SWAVeform:DIFF:STATe :LTESt:MEASure:SWAVeform:FUNCtion:STATe
:LTESt:SWAVeform:RESet	:LTESt:MEASure:SWAVeform:RESet
:LTESt:TEST	:LTESt:MEASure:AMPLitude:LOCation:STATe :LTESt:MEASure:EYE:LOCation:STATe :LTESt:MEASure:JITTer:LOCation:STATe :LTESt:MEASure:MTESt:LOCation:STATe :LTESt:MEASure:OSCilloscope:LOCation:STATe

Table 34 :LTESt Commands (Sheet 2 of 2)

86100D Legacy Command	Similar FlexDCA Command
:LTESt:ULIMit	:LTESt:MEASure:AMPLitude:LOCation:ULIMit :LTESt:MEASure:EYE:LOCation:ULIMit :LTESt:MEASure:JITTer:LOCation:ULIMit :LTESt:MEASure:MTESt:LOCation:ULIMit :LTESt:MEASure:OSCilloscope:LOCation:ULIMit

Table 35 :MARKer Commands

86100D Legacy Command	Similar FlexDCA Command
:MARKer:PROPagation	:GRAPh:T:OHMS:X:DCONstant :GRAPh:T:OHMS:X:VFACtor :GRAPh:T:PERCent:X:DCONstant :GRAPh:T:PERCent:X:VFACtor :GRAPh:T:VOLTs:X:DCONstant :GRAPh:T:VOLTs:X:VFACtor
:MARKer:REACtance?	:MEASure:TDR:ECAPacitance :MEASure:TDR:EINDuctance
:MARKer:REFerence	–
:MARKer:RPANnotation	–
:MARKer:STATe	:MARKer:X1:STATe :MARKer:X2:STATe :MARKer:Y1:STATe :MARKer:Y2:STATe
:MARKer:X1Position	:MARKer:X1:POSition
:MARKer:X1Y1source	:MARKer:X1:SOURce :MARKer:Y1:SOURce
:MARKer:X2Position	:MARKer:X2:POSition
:MARKer:X2Y2source	:MARKer:X2:SOURce :MARKer:Y2:SOURce
:MARKer:XDELta?	:MEASure:MARKer:XDELta? :MEASure:MARKer:IXDelta?
:MARKer:XUNits	:GRAPh:T:OHMS:X:UNITs :GRAPh:T:PERCent:X:UNITs :GRAPh:T:VOLTs:X:UNITs
:MARKer:Y1Position	:MARKer:Y1:POSition
:MARKer:Y2Position	:MARKer:Y2:POSition
:MARKer:YDELta?	:MEASure:MARKer:YDELta?
:MARKer:YUNits	–

Table 36 :MTESt Commands (Sheet 1 of 2)

86100D Legacy Command	Similar FlexDCA Command
:MTESt:ALIGn	Masks now automatically align.
:MTESt:AMARgin:BER	:MTESt{1:16}:MARGin:AUTo:HRATio
:MTESt:AMARgin:CALCulate	:MTESt{1:16}:MARGin:METHod AUTO
:MTESt:AMEThod	:MTESt{1:16}:ALIGnment:X:METHod
:MTESt:AOPTimize	:MTESt{1:16}:AOPTimize
:MTESt:ARUN	–
:MTESt:COUNt:FAILures?	:MEASure:MTESt{1:16}:THRegion{1:16}?
:MTESt:COUNt:FSAMples?	:MEASure:MTESt{1:16}:HITs?
:MTESt:COUNt:HITS?	:MEASure:MTESt{1:16}:HITS? :MEASure:MTESt{1:16}:THITs? :MEASure:MTESt{1:16}:HREGion? :MEASure:MTESt{1:16}:MHITs? :MEASure:MTESt{1:16}:MHRegion?
:MTESt:COUNt:SAMPles?	:MEASure:MTESt{1:16}:NSAMples?
:MTESt:COUNt:WAVeforms?	:MEASure:MTESt{1:16}:NWAVeforms?
:MTESt:DELete	–
:MTESt:EXIT	:MTESt{1:16}:DISPlay
:MTESt:LOAD	:MTESt{1:16}:FNAMe :MTESt{1:16}:LOAD
:MTESt:MASK:DELete	–
:MTESt:MMARgin:PERCent	:MTESt{1:16}:MARGin:METHod :MTESt{1:16}:MARGin:PERCent
:MTESt:MMARgin:STATe	:MTESt{1:16}:MARGin:STATe
:MTESt:RUNTil	:LTESt:MTESt:STATe :LTESt:MEASure:MTESt:LOCation:STATe
:MTESt:SAVE	–
:MTESt:SCALe:DEFault	–
:MTESt:SCALe:MODE	–
:MTESt:SCALe:SOURce?	–
:MTESt:SCALe:X1	:MTESt{1:16}:X1
:MTESt:SCALe:XDELta	:MTESt{1:16}:XDELta
:MTESt:SCALe:Y1	:MTESt{1:16}:Y1
:MTESt:SCALe:Y2	:MTESt{1:16}:Y2

Table 36 :MTESt Commands (Sheet 2 of 2)

86100D Legacy Command	Similar FlexDCA Command
:MTESt:SOURce	:MTESt{1:16}:SOURce
:MTESt:SCALe:YTRack	:MTESt{1:16}:YTRack
:MTESt:SSCReen	:LTESt:MTESt:SIMage:STATe
:MTESt:SSCReen:AREA	:LTESt:MTESt:SIMage:GONLy
:MTESt:SSCReen:IMAGe	:LTESt:MTESt:SIMage:INVert :LTESt:MTESt:SIMage:MONochrome
:MTESt:SSUMmary	:LTESt:MTESt:SSUMmary:STATe
:MTESt:STARt	:MTESt{1:16}:LOAD
:MTESt:SWAVeform	:LTESt:MTESt:SWAVeform:CHAN:STATe :LTESt:MTESt:SWAVeform:DIFF:STATe :LTESt:MTESt:SWAVeform:FUNCtion:STATe
:MTESt:SWAVeform:RESet	:LTESt:MTESt:SWAVeform:RESet
:MTESt:TEST	:MTESt{1:16}:DISPlay
:MTESt:TITLe?	:MTESt{1:16}:TITLe?
:MTESt:YALign	:MTESt{1:16}:YALign

Table 37 :MEASure Commands (Sheet 1 of 8)

86100D Legacy Command	Similar FlexDCA Command
:MEASure:AMPLitude:ANALysis	:MEASure:AMPLitude:DEFine:ANALysis
:MEASure:AMPLitude:DI?	:MEASure:AMPLitude:DIONes:SOURce :MEASure:AMPLitude:DIONes? :MEASure:AMPLitude:DIZeros:SOURce :MEASure:AMPLitude:DIZeros?
:MEASure:AMPLitude:EOPening?	:MEASure:AMPLitude:EOPening:SOURce :MEASure:AMPLitude:EOPening?
:MEASure:AMPLitude:ISI?	:MEASure:AMPLitude:ISIOnes:SOURce :MEASure:AMPLitude:ISIOnes? :MEASure:AMPLitude:ISIZeros:SOURce :MEASure:AMPLitude:ISIZeros?
:MEASure:AMPLitude:ISIVsbit?	:MEASure:AMPLitude:ISIVsbit?
:MEASure:AMPLitude:ISIVsbit:BITS?	:MEASure:AMPLitude:ISIVsbit:BITS?
:MEASure:AMPLitude:ISIVsbit:HIGHest?	:MEASure:AMPLitude:ISIVsbit:HIGHest?
:MEASure:AMPLitude:ISIVsbit:LOWest?	:MEASure:AMPLitude:ISIVsbit:LOWest?
:MEASure:AMPLitude:LEVel:CIDigits:LAGGing	:MEASure:AMPLitude:DEFine:LEVel:CIDigits:LAGGing
:MEASure:AMPLitude:LEVel:CIDigits:LEADing	:MEASure:AMPLitude:DEFine:LEVel:CIDigits:LEADing
:MEASure:AMPLitude:LEVel:DEFine	:MEASure:AMPLitude:DEFine:LEVel
:MEASure:AMPLitude:LOCation	:MEASure:AMPLitude:DEFine:LOCation
:MEASure:AMPLitude:OLEVel?	:MEASure:AMPLitude:OLEVel:SOURce :MEASure:AMPLitude:OLEVel?
:MEASure:AMPLitude:PI?	:MEASure:AMPLitude:PIONes:SOURce :MEASure:AMPLitude:PIONes? :MEASure:AMPLitude:PIZeros:SOURce :MEASure:AMPLitude:PIZeros?
:MEASure:AMPLitude:PIRMs?	:MEASure:AMPLitude:PIROnes:SOURce :MEASure:AMPLitude:PIROnes? :MEASure:AMPLitude:PIRZeros:SOURce :MEASure:AMPLitude:PIRZeros?
:MEASure:AMPLitude:Q?	:MEASure:AMPLitude:Q:SOURce :MEASure:AMPLitude:Q?
:MEASure:AMPLitude:RINoise?	:MEASure:AMPLitude:RINoise:SOURce :MEASure:AMPLitude:RINoise?
:MEASure:AMPLitude:RINoise:DEF	:MEASure:AMPLitude:DEFine:RINoise:TYPe

Table 37 :MEASure Commands (Sheet 2 of 8)

86100D Legacy Command	Similar FlexDCA Command
:MEASure:AMPLitude:RINoise:UNITs	:MEASure:AMPLitude:DEFine:RINoise:UNITs
:MEASure:AMPLitude:RN?	:MEASure:AMPLitude:RNONes:SOURce :MEASure:AMPLitude:RNONes? :MEASure:AMPLitude:RNZeros:SOURce :MEASure:AMPLitude:RNZeros?
:MEASure:AMPLitude:RNSTabilize	:MEASure:AMPLitude:DEFine:RNSTabilize
:MEASure:AMPLitude:RNSValue	:MEASure:AMPLitude:DEFine:RNSValue:ONE :MEASure:AMPLitude:DEFine:RNSValue:ZERO :MEASure:AMPLitude:DEFine:RNSValue:GMRN
:MEASure:AMPLitude:SAMPlitude?	:MEASure:AMPLitude:SAMPlitude:SOURce :MEASure:AMPLitude:SAMPlitude?
:MEASure:AMPLitude:TI?	:MEASure:AMPLitude:TIONes:SOURce :MEASure:AMPLitude:TIONes? :MEASure:AMPLitude:TIZeros:SOURce :MEASure:AMPLitude:TIZeros?
:MEASure:AMPLitude:TI:DEFine	:DISPlay:AMPLitude:LEVel
:MEASure:AMPLitude:UNITs	:MEASure:AMPLitude:DEFine:UNITs
:MEASure:AMPLitude:ZLEVel?	:MEASure:AMPLitude:ZLEVel
:MEASure:ANNotation	–
:MEASure:APOWer	:MEASure:EYE:APOWer:SOURce :MEASure:EYE:APOWer:UNITs :MEASure:EYE:APOWer? :MEASure:OSCilloscope:APOWer:SOURce :MEASure:OSCilloscope:APOWer:UNITs :MEASure:OSCilloscope:APOWer?
:MEASure:APOWer:CORRection	:CALibrate:DARK:CHANnel{1A:8D}:STARt
:MEASure:CGRade:AMPLitude?	:MEASure:EYE:AMPLitude:SOURce :MEASure:EYE:AMPLitude?
:MEASure:CGRade:BITRate?	:MEASure:EYE:BITRate:SOURce :MEASure:EYE:BITRate?
:MEASure:CGRade:COMPlete	:MEASure:EYE:COMPlete
:MEASure:CGRade:CRATio	–
:MEASure:CGRade:CROSsing?	:MEASure:EYE:CROSsing:SOURce :MEASure:EYE:CROSsing?
:MEASure:CGRade:DCDistortion	:MEASure:EYE:DCDistortion:SOURce :MEASure:EYE:DCDistortion?
:MEASure:CGRade:DCYCle	–
:MEASure:CGRade:EHEight	:MEASure:EYE:EHEight:SOURce :MEASure:EYE:EHEight?

Table 37 :MEASure Commands (Sheet 3 of 8)

86100D Legacy Command	Similar FlexDCA Command
:MEASure:CGRade:ERATio	:MEASure:EYE:ERATio:SOURce :MEASure:EYE:ERATio:UNITs :MEASure:EYE:ERATio?
:MEASure:CGRade:ERFactor	:MEASure:ERATio:CHANnel{1A:8D}:ACFactor :MEASure:ERATio:CHANnel{1A:8D}:CFACtor
:MEASure:CGRade:ESN	:MEASure:EYE:ESN:SOURce :MEASure:EYE:ESN?
:MEASure:CGRade:EWIDth	:MEASure:EYE:EWIDth:SOURce :MEASure:EYE:EWIDth? :MEASure:EYE:EWIDth:FORMat
:MEASure:CGRade:JITTer	:MEASure:EYE:JITTer:SOURce :MEASure:EYE:JITTer:FORMat :MEASure:EYE:JITTer?
:MEASure:CGRade:OFACtor	–
:MEASure:CGRade:OLEVel	:MEASure:EYE:OLEVel:SOURce :MEASure:EYE:OLEVel?
:MEASure:CGRade:PEAK?	:MEASure:EYE:PEAK:SOURce :MEASure:EYE:PEAK?
:MEASure:CGRade:PWIDth	–
:MEASure:CGRade:SMOothing	–
:MEASure:CGRade:SOURce	Use the :SOURce child command for the selected measurement. For example, :MEASURE:EYE:AMPLitude:SOURce
:MEASure:CGRade:ZLEVel	:MEASure:EYE:ZLEVel:SOURce :MEASure:EYE:ZLEVel?
:MEASure:CLEar	:MEASure:AMPLitude:LIST:CLEar :MEASure:EYE:LIST:CLEar :MEASure:JITTer:LIST:CLEar :MEASure:OSCilloscope:LIST:CLEar :MEASure:TDR:LIST:CLEar

Table 37 :MEASure Commands (Sheet 4 of 8)

86100D Legacy Command	Similar FlexDCA Command
:MEASure:DEFine THResholds	:MEASure:THReshold:METHod :MEASure:THReshold:UNITs Based on Percent :MEASure:THReshold:DISTal :MEASure:THReshold:MESial :MEASure:THReshold:PROXimal Based on Units :CHANnel{1A:8D}:UDEFined:DISTal :EMEMory{1:8}:UDEFined:DISTal :WMEMory{1:8}:UDEFined:DISTal :FUNCtion{1:16}:UDEFined:DISTal :CHANnel{1A:8D}:UDEFined:MESial :EMEMory{1:8}:UDEFined:MESial :WMEMory{1:8}:UDEFined:MESial :FUNCtion{1:16}:UDEFined:MESial :CHANnel{1A:8D}:UDEFined:PROXimal :EMEMory{1:8}:UDEFined:PROXimal :WMEMory{1:8}:UDEFined:PROXimal :FUNCtion{1:16}:UDEFined:PROXimal
:MEASure:DEFine TREFerence	:MEASure:THReshold:EREFerence
:MEASure:DEFine TOPBase	:MEASure:TBASe:METHod :CHANnel{1A:8D}:UDEFined:BASe :CHANnel{1A:8D}:UDEFined:TOP :EMEMory{1:8}:UDEFined:BASe :EMEMory{1:8}:UDEFined:TOP :WMEMory{1:8}:UDEFined:BASe :WMEMory{1:8}:UDEFined:TOP :FUNCtion{1:16}:UDEFined:BASe :FUNCtion{1:16}:UDEFined:TOP
:MEASure:DEFine EWINdow	:MEASure:EBOundary:DEFault :MEASure:EBOundary:LEFT :MEASure:EBOundary:RIGHt
:MEASure:DEFine DELTatime	:MEASure:DELTatime:EDIRection :MEASure:DELTatime:ENUMber :MEASure:DELTatime:ETHReshold
:MEASure:DEFine CGRade	–
:MEASure:DEFine RZEWindow	–
:MEASure:DELTatime	:MEASure:EYE:DELTatime:SOURce :MEASure:EYE:DELTatime? :MEASure:OSCilloscope:DELTatime:SOURce :MEASure:OSCilloscope:DELTatime?
:MEASure:DUTYcycle	:MEASure:OSCilloscope:DUTYcycle:SOURce :MEASure:OSCilloscope:DUTYcycle?

Table 37 :MEASure Commands (Sheet 5 of 8)

86100D Legacy Command	Similar FlexDCA Command
:MEASure:FALLtime?	:MEASure:EYE:FALLtime:SOURce :MEASure:EYE:FALLtime? :MEASure:OSCilloscope:FALLtime:SOURce :MEASure:OSCilloscope:FALLtime?
:MEASure:FREQuency	:MEASure:OSCilloscope:FREQuency:SOURce :MEASure:OSCilloscope:FREQuency?
:MEASure:HISTogram:HITS?	:MEASure:HISTogram{1:4}:HITS?
:MEASure:HISTogram:M1S?	:MEASure:HISTogram{1:4}:M1S?
:MEASure:HISTogram:M2S?	:MEASure:HISTogram{1:4}:M2S?
:MEASure:HISTogram:M3S?	:MEASure:HISTogram{1:4}:M3S?
:MEASure:HISTogram:MEAN?	:MEASure:HISTogram{1:4}:MEAN?
:MEASure:HISTogram:MEDian?	:MEASure:HISTogram{1:4}:MEDian?
:MEASure:HISTogram:PEAK?	:MEASure:HISTogram{1:4}:PEAK?
:MEASure:HISTogram:PP?	:MEASure:HISTogram{1:4}:PP?
:MEASure:HISTogram:PPOSition?	:MEASure:HISTogram{1:4}:PPOSition?
:MEASure:HISTogram:SCALe?	:MEASure:HISTogram{1:4}:SCALe?
:MEASure:HISTogram:STDDev?	:MEASure:HISTogram{1:4}:STDDev?
:MEASure:JITTer:DCD?	:MEASure:JITTer:DCD?
:MEASure:JITTer:DDJ?	:MEASure:JITTer:DDJ?
:MEASure:JITTer:DDJVsbit?	:MEASure:JITTer:DDJVsbit?
:MEASure:JITTer:DDJVsbit:BITS?	:MEASure:JITTer:DDJVsbit:BITS?
:MEASure:JITTer:DDJVsbit:EARLiest?	:MEASure:JITTer:DDJVsbit:EARLiest?
:MEASure:JITTer:DDJVsbit:LATest?	:MEASure:JITTer:DDJVsbit:LATest?
:MEASure:JITTer:DJ?	:MEASure:JITTer:DJ?
:MEASure:JITTer:EBITs?	:MEASure:JITTer:EBITs?
:MEASure:JITTer:EDGE	:MEASure:DEFine:EDGe
:MEASure:JITTer:FREQuency:ANALysis	–
:MEASure:JITTer:FREQuency:COMPonents?	–
:MEASure:JITTer:FREQuency:MAXNumber	–
:MEASure:JITTer:FREQuency:SCAN	–
:MEASure:JITTer:ISI?	:MEASure:JITTer:ISI?

Table 37 :MEASure Commands (Sheet 6 of 8)

86100D Legacy Command	Similar FlexDCA Command
:MEASure:JITTer:LEVel?	:MEASure:JITTer:LEVel?
:MEASure:JITTer:LEVel:DEFine	:MEASure:JITTer:LEVel
:MEASure:JITTer:PATTern?	:MEASure:JITTer:PATTern?
:MEASure:JITTer:PJ?	:MEASure:JITTer:PJ?
:MEASure:JITTer:PJRMs?	:MEASure:JITTer:PJRMs?
:MEASure:JITTer:RJ?	:MEASure:JITTer:RJ?
:MEASure:JITTer:RJSTablize	:MEASure:DEFine:RJSTablize
:MEASure:JITTer:RJSValue	:MEASure:DEFine:RJSValue :MEASure:DEFine:RJSValue:GMRJ
:MEASure:JITTer:SIGNal	–
:MEASure:JITTer:SIGNal:AUTodetect	–
:MEASure:JITTer:TJ?	:MEASure:JITTer:TJ?
:MEASure:JITTer:TJ:DEFine	:MEASure:DEFine:TJBer
:MEASure:JITTer:UNITs	:MEASure:DEFine:UNITs
:MEASure:MATLab	–
:MEASure:MATLab<N>:SCRipt	–
:MEASure:MATLab<N>:ETENable	–
:MEASure:MATLab<N>:ETEXt?	–
:MEASure:NWIDth	:MEASure:OSCilloscope:NWIDth:SOURce :MEASure:OSCilloscope:NWIDth?
:MEASure:OMAMplitude	:MEASure:OSCilloscope:OMAmplitude:SOURce :MEASure:OSCilloscope:OMAmplitude?
:MEASure:OVERshoot	:MEASure:OSCilloscope:OVERshoot:SOURce :MEASure:OSCilloscope:OVERshoot?
:MEASure:PERiod	:MEASure:OSCilloscope:PERiod:SOURce :MEASure:OSCilloscope:PERiod?
:MEASure:PWIDth	:MEASure:OSCilloscope:PWIDth:SOURce :MEASure:OSCilloscope:PWIDth?
:MEASure:RESults?	:MEASure:RESults? The format of the returned output is significantly different.
:MEASure:RISetime?	:MEASure:EYE:RISetime:SOURce :MEASure:EYE:RISetime :MEASure:OSCilloscope:RISetime:SOURce :MEASure:OSCilloscope:RISetime? :MEASure:TDR:RISetime:SOURce :MEASure:TDR:RISetime

Table 37 :MEASure Commands (Sheet 7 of 8)

86100D Legacy Command	Similar FlexDCA Command
:MEASure:SCRatch	:MEASure:AMPLitude:LIST:CLEar :MEASure:EYE:LIST:CLEar :MEASure:JITTer:LIST:CLEar :MEASure:OSCilloscope:LIST:CLEar :MEASure:TDR:LIST:CLEar
:MEASure:SENDvalid	–
:MEASure:SINTegrity:BERFloor?	:MEASure:AMPLitude:BERFloor
:MEASure:SINTegrity:BERLimit?	:MEASure:AMPLitude:BERLimit
:MEASure:SINTegrity:PATTern?	:MEASure:JITTer:PATTern
:MEASure:SINTegrity:SIGNal	:MEASure:JITTer:DEFine:SIGNal
:MEASure:SINTegrity:SIGNal:AUTodetect	:MEASure:JITTer:DEFine:SIGNal:AUTodetect
:MEASure:SOURce	Use the :SOURce child command for the selected Oscilloscope mode measurement. For example, :MEASURE:EYE:AMPLitude:SOURce.
:MEASure:TEDGe?	:MEASure:OSCilloscope:TEDGe:SOURce :MEASure:OSCilloscope:TEDGe? :MEASure:TDR:TEDGe:SOURce :MEASure:TDR:TEDGe?
:MEASure:TDR:AVERage	:MEASure:TDR:VAVerage:SOURce :MEASure:TDR:VAVerage?
:MEASure:TDR:MAX	:MEASure:TDR:VMAXimum:SOURce :MEASure:TDR:VMAXimum?
:MEASure:TDR:MIN	:MEASure:TDR:VMINimum:SOURce :MEASure:TDR:VMINimum?
:MEASure:TMAX	:MEASure:OSCilloscope:TMAXimum:SOURce :MEASure:OSCilloscope:TMAXimum?
:MEASure:TMIN	:MEASure:OSCilloscope:TMINimum:SOURce :MEASure:OSCilloscope:TMINimum?
:MEASure:TVOLt?	:MEASure:OSCilloscope:TVOLt:EDIRection :MEASure:OSCilloscope:TVOLt:ENUMber :MEASure:OSCilloscope:TVOLt:YVALue :MEASure:OSCilloscope:TVOLt:SOURce :MEASure:OSCilloscope:TVOLt?
:MEASure:VAMPlitude?	:MEASure:OSCilloscope:VAMPlitude:SOURce :MEASure:OSCilloscope:VAMPlitude?
:MEASure:VAVerage	:MEASure:OSCilloscope:VAVerage:AREa :MEASure:OSCilloscope:VAVerage:SOURce :MEASure:OSCilloscope:VAVerage?
:MEASure:VBASe	:MEASure:OSCilloscope:VBASe:SOURce :MEASure:OSCilloscope:VBASe?

Table 37 :MEASure Commands (Sheet 8 of 8)

86100D Legacy Command	Similar FlexDCA Command
:MEASure:VMAX	:MEASure:OSCilloscope:VMAXimum:SOURce :MEASure:OSCilloscope:VMAXimum?
:MEASure:VMIN	:MEASure:OSCilloscope:VMINimum:SOURce :MEASure:OSCilloscope:VMINimum?
:MEASure:VPP	:MEASure:OSCilloscope:VPP:SOURce :MEASure:OSCilloscope:VPP?
:MEASure:VRMS	:MEASure:OSCilloscope:VRMS:AREa :MEASure:OSCilloscope:VRMS:TYPe :MEASure:OSCilloscope:VRMS:SOURce :MEASure:OSCilloscope:VRMS?
:MEASure:VTIMe?	:MEASure:OSCilloscope:VTIMe:TIMe :MEASure:OSCilloscope:VTIMe:SOURce :MEASure:OSCilloscope:VTIMe?
:MEASure:VTOP	:MEASure:OSCilloscope:VTOP:SOURce :MEASure:OSCilloscope:VTOP?

Table 38 :SPRocessing Commands

86100D Legacy Command	Similar FlexDCA Command
:SPRocessing:LFEQualizer	:FUNCtion:FOPerator FFEQualizer
:SPRocessing:LFEQualizer:BANDwidth	:SPRocess:FFEQualizer:BANDwidth
:SPRocessing:LFEQualizer:BWMode	:SPRocess:FFEQualizer:BANDwidth:AUTo
:SPRocessing:LFEQualizer:FDELay	:SPRocess:FFEQualizer:NPRecursors
:SPRocessing:LFEQualizer:NTAPs	:SPRocess:FFEQualizer:TAPS:COUNt
:SPRocessing:LFEQualizer:TAP	:SPRocess:FFEQualizer:TAPS
:SPRocessing:LFEQualizer:TAP:AUTomatic	:SPRocess:FFEQualizer:TAPS:AUTo
:SPRocessing:LFEQualizer:TAP:NORMalize	:SPRocess:FFEQualizer:TAPS:NORMalize
:SPRocessing:LFEQualizer:TDELay	:SPRocess:FFEQualizer:TSPacing:TIMe :SPRocess:FFEQualizer:TSPacing:TPBit
:SPRocessing:LFEQualizer:TDMode	:SPRocess:FFEQualizer:TSPacing
:SPRocessing:MATLab	–
:SPRocessing:MATLab:ETENable	–
:SPRocessing:MATLab:ETEXt	–
:SPRocessing:MATLab:SCRipt	–
:SPRocessing:OUTPut	:FUNCtion:FOPerator
:SPRocessing:SOURce	:FUNCtion:OPERand
:SPRocessing:SOURce:DISPlay	:FUNCtion:DISPlay :CHANnel:DISPlay :DIFF:DISPlay :WMEMory:DISPlay

Table 39 :SYSTem Commands

86100D Legacy Command	Similar FlexDCA Command
:SYSTem:DATE	:SYSTem:DATE
:SYSTem:DSP	:SYSTem:DSP
:SYSTem:ERRor?	:SYSTem:ERRor?
:SYSTem:FCONfig	–
:SYSTem:HEADer	FlexDCA does not output a header with a query response.
:SYSTem:LONGform	–
:SYSTem:MODE	:SYSTem:MODE
:SYSTem:SETup	–
:SYSTem:TIME	:SYSTem:TIME

Table 40 :TDR Commands

86100D Legacy Command	Similar FlexDCA Command
TDR/TDT configuration and measurements is very different between the legacy and FlexDCA environments. Consult FlexDCA's help system and use FlexDCA's SCPI Recorder. Use FlexDCA's :TRACe subsystem to display TDR and TDT waveforms. Search in FlexDCA's help for information on "waveform content windows".	
:CONNect	:TDR:CHANnel{1A:8D}:PORT
:DUT:DIRection	–
:DUT:TYPE	:TDR:AMODe :TDR:DUT{1:16}:DTYPe
:RESPonse:CALibrate	:TDR:CALibration:DUT{1:16}:STARt
:RESPonse:DISPlay	:TRACe{1:128}:DISPlay
:RESPonse:RISetime	:TDR:DUT{1:16}:RISetime
:RESPonse:RPLane?	–
:RESPonse:TYPE	–
:RESPonse:VAMPlitude?	–
:RESPonse:VERTical	–
:RESPonse:VERTical:OFFSet	–
:RESPonse:VERTical:RANGe	–
:RESPonse:VLOad?	–
:STIMulus:EXTernal	–
:STIMulus:EXTernal:POLarity	–
:STIMulus:MODE	:TDR:STIMulus:CHANnel{1A:8D}:TYPe
:STIMulus:RATE	:TDR:STIMulus:SLOT{1:8}:RATE :TDR:STIMulus:SLOT{1:8}:RATE:AUTomatic
:STIMulus:STATe	:TDR:STIMulus:CHANnel{1A:8D}:STEP

Table 41 :TIMebase Commands

86100D Legacy Command	Similar FlexDCA Command
:TIMebase:BRATe	:TIMebase:BRATe
:TIMebase:MPOSition	:TIMebase:MPOSition
:TIMebase:POSition	:TIMebase:POSition
:TIMebase:PRECision	:PTIMebase:STATe
:TIMebase:PRECision:REFSource	:PTIMebase:RSOurce
:TIMebase:PRECision:RFRequency	:PTIMebase:RFRequency
:TIMebase:PRECision:RFRequency:AUTodetect	:PTIMebase:RFRequency:AUTo
:TIMebase:PRECision:TREFerence	:PTIMebase:RTReference
:TIMebase:RANGe	:TIMebase:XRANge
:TIMebase:REFerence	:TIMebase:REFerence
:TIMebase:SCALe	:TIMebase:SCALe
:TIMebase:UNITs	:TIMebase:UNITs

Table 42 :TRIGger Commands

86100D Legacy Command	Similar FlexDCA Command
:TRIGger:ATTenuation	–
:TRIGger:BRATe	:TRIGger:BRATe
:TRIGger:BRATe:AUTodetect	:TRIGger:BRATe:AUTodetect
:TRIGger:BWLimit	:TRIGger:BWLimit
:TRIGger:DCDRatio	:TRIGger:DCDRatio
:TRIGger:DCDRatio:AUTodetect	:TRIGger:DCDRatio:AUTodetect
:TRIGger:GATed	–
:TRIGger:HYSTeresis	:TRIGger:HYSTeresis
:TRIGger:LEVel	:TRIGger:LEVel
:TRIGger:PLENgth	:TRIGger:PLENgth
:TRIGger:PLENgth:AUTodetect	:TRIGger:PLENgth:AUTodetect
:TRIGger:PLOCk	:TRIGger:PLOCk
:TRIGger:PLOCk:AUTodetect	–
:TRIGger:RBIT	–
:TRIGger:SLOPe	:TRIGger:SLOPe
:TRIGger:SOURce	:TRIGger:SOURce

Table 43 :WAVeform Commands (Sheet 1 of 2)

86100D Legacy Command	Similar FlexDCA Command
:WAVeform:BANDpass?	–
:WAVeform:BYTeorder	–
:WAVeform:COUNt?	–
:WAVeform:DATA	–
:WAVeform:DATA?	EYE Format :WAVeform:EYE:ASCii:DATa? :WAVeform:EYE:INTeger:DATa? XYFormat :WAVeform:XYFormat:ASCii:XDATa? :WAVeform:XYFormat:ASCii:YDATa? :WAVeform:XYFormat:DOUBle:XDATa? :WAVeform:XYFormat:DOUBle:YDATa? :WAVeform:XYFormat:FLOat:XDATa? :WAVeform:XYFormat:FLOat:YDATa? YFORmat :WAVeform:YFORmat:ASCii:YDATa? :WAVeform:YFORmat:DOUBle:YDATa? :WAVeform:YFORmat:FLOat:YDATa? :WAVeform:YFORmat:WORD:YDATa?
:WAVeform:FORMat	–
:WAVeform:POINts?	:WAVeform:EYE:POINts? :WAVeform:XYFormat:POINts? :WAVeform:YFORmat:POINts?
:WAVeform:PREamble	–
:WAVeform:SOURce	:WAVeform:SOURce
:WAVeform:SOURce:CGRade	:WAVeform:SOURce
:WAVeform:TYPE?	–
:WAVeform:XDISplay?	:TIMebae:XLEFt?
:WAVeform:XINCrement?	:WAVeform:EYE:XINCrement? :WAVeform:YFORmat:XINCrement?
:WAVeform:XORigin?	:WAVeform:EYE:XORigin? :WAVeform:YFORmat:XORigin?
:WAVeform:XRANge?	:TIMebase:RANge?
:WAVeform:XREFerence?	:EMEMory:XREFerence? :FUNCtion:XREFerence? :WMEMory:XREFerence?
:WAVeform:XUNits?	:TIMebase:UNITs?

Table 43 :WAVeform Commands (Sheet 2 of 2)

86100D Legacy Command	Similar FlexDCA Command
:WAVeform:YDISplay?	–
:WAVeform:YINCrement?	:WAVeform:EYE:YINCrement?
:WAVeform:YORigin?	:WAVeform:EYE:YORigin?
:WAVeform:YRANge?	:CHANnel:YTOP? and YBOTtom? :DIFF:YTOP? and YBOTtom? :EMEMory:YTOP? and YBOTtom? :FUNCtion:YTOP? and YBOTtom? :WMEMory:YTOP? and YBOTtom?
:WAVeform:YREFerence?	–
:WAVeform:YUNits?	:CHANnel:YUNits? :DIFF:YUNits? :EMEMory:YUNits? :FUNCtion:YUNits? :WMEMory:YUNits?

Table 44 :WMEMory Commands

86100D Legacy Command	Similar FlexDCA Command
:WMEMory:DISPlay	:WMEMory:DISPlay
:WMEMory:LOAD	:DISK:WAVeform:RECall
:WMEMory:SAVE	:WMEMory:LOAD
:WMEMory:XOFFset	:WMEMory:XPOSition
:WMEMory:XRANge	:WMEMory:XSCale
:WMEMory:YOFFset	:WMEMory:YOFFset
:WMEMory:YRANge	:WMEMory:YSCale

Index

R

S

T

U

V

W