Keysight M8000 Series of BER Test Solutions

J-BERT M8020A High-Performance BERT
M8030A Multi-Channel BERT
M8040A High-Performance BERT
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.

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, 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 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 WILL CONTROL.

Safety Notices

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.
Safety Summary

The following general safety precautions must be observed during all phases of operation of this instrument. Failure to comply with these precautions or with specific warnings or operating instructions in the product manuals violates safety standards of design, manufacture and intended use of the instrument. Keysight Technologies assumes no liability for the customer’s failure to comply with these requirements. Product manuals are provided with your instrument on CD-ROM and/or in printed form. Printed manuals are an option for many products. Manuals may also be available on the Web. Go to www.keysight.com and type in your product number in the Search field at the top of the page.

General

This product is a Safety Class 1 instrument (provided with a protective earth terminal). The protective features of this product may be impaired if it is used in a manner not specified in the operation instructions.

All Light Emitting Diodes (LEDs) used in this product are Class 1 LEDs as per IEC 60825-1.

Environment Conditions

This instrument is intended for indoor use in an installation category II, pollution degree 2 environment. It is designed to operate at a maximum relative humidity of 95% and at altitudes of up to 2000 meters. Refer to the specifications tables for the ac mains voltage requirements and ambient operating temperature range.

Before Applying Power

Verify that all safety precautions are taken. The power cable inlet of the instrument serves as a device to disconnect from the mains in case of hazard. The instrument must be positioned so that the operator can easily access the power cable inlet. When the instrument is rack mounted the rack must be provided with an easily accessible mains switch.

Ground the Instrument

To minimize shock hazard, the instrument chassis and cover must be connected to an electrical protective earth ground. The instrument must be connected to the ac power mains through a grounded power cable, with the ground wire firmly connected to an electrical ground (safety ground) at the power outlet. Any interruption of the protective (grounding) conductor or disconnection of the protective earth terminal will cause a potential shock hazard that could result in personal injury.

Do Not Operate in an Explosive Atmosphere

Do not operate the instrument in the presence of flammable gases or fumes.

Do Not Remove the Instrument Cover

Operating personnel must not remove instrument covers. Component replacement and internal adjustments must be made only by qualified personnel. Instruments that appear damaged or defective should be made inoperative and secured against unintended operation until they can be repaired by qualified service personnel.
### Safety Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Symbol" /></td>
<td>Indicates warning or caution. If you see this symbol on a product, you must refer to the manuals for specific Warning or Caution information to avoid personal injury or damage to the product.</td>
</tr>
<tr>
<td><img src="image" alt="Symbol" /></td>
<td>Frame or chassis ground terminal. Typically connects to the equipment's metal frame.</td>
</tr>
<tr>
<td><img src="image" alt="Symbol" /></td>
<td>KC is the Korean certification mark to demonstrate that the equipment is Class A suitable for professional use and is for use in electromagnetic environments outside of the home.</td>
</tr>
<tr>
<td><img src="image" alt="Symbol" /></td>
<td>Indicates that antistatic precautions should be taken.</td>
</tr>
<tr>
<td><img src="image" alt="Symbol" /></td>
<td>Indicates the time period during which no hazardous or toxic substance elements are expected to leak or deteriorate during normal use. Forty years is the expected useful life of the product.</td>
</tr>
<tr>
<td><img src="image" alt="Symbol" /></td>
<td>The RCM Mark is a compliance mark to the ACMA (Australian Spectrum Management Agency). This indicates compliance with all Australian EMC regulatory information.</td>
</tr>
</tbody>
</table>
## Compliance and Environmental Information

<table>
<thead>
<tr>
<th>Safety Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Safety Symbol" /></td>
<td>This product complies with WEEE Directive (2002/96/EC) marking requirements. The affixed label indicates that you must not discard this electrical/electronic product in domestic household waste.</td>
</tr>
<tr>
<td></td>
<td>Product Category: With reference to the equipment types in WEEE Directive Annex I, this product is classed as a “Monitoring and Control instrumentation” product. Do not dispose in domestic household waste.</td>
</tr>
</tbody>
</table>
## Contents

Safety Summary .................................................. 3  
Safety Symbols ................................................... 4  
Compliance and Environmental Information ................. 5  

1 Programming Basics

**Modular Configuration** .................................... 16  
- M8020A Modular Configuration .................................. 16  
- M8030A Modular Configuration .................................. 19  
- M8040A Modular Configuration .................................. 21  

**Using Identifiers** ........................................... 23  
- Location Identifiers .............................................. 23  
- Group Name Identifiers .......................................... 24  
- Measurement Name Identifiers .................................. 24  

**PLUGin Subsystem** ......................................... 25  

2 Recommended Programming Techniques

**Output Protection** .......................................... 28  
- Pattern Generator Output Termination ......................... 28  
- Output Protection Algorithm ................................... 28  

**Controlling the Output Levels** .......................... 29  
- Controlling the Output Levels - Concepts .................... 29  
- Controlling the Output Levels - Procedures ................. 29
Contents

Reading the M8020A/M8030A/M8040A's Status 30
   Reading the M8020A/M8030A/M8040A's Status - Concepts 30
   M8020A/M8030A/M8040A Status Reporting Structure 33

Program Snippet Extending the *opc? Functionality 37

Working with Patterns 38
   Symbol Coding Descriptions 39
   Symbol Sequences 44

Creating Pattern Sequences 45
   Sequence Editor 45
   Pattern Sequence Building Blocks 47
   Sequencer Triggers 51
   Block Branching 52
   Link Training Events 53
   Block Controls 53
   Coding Configuration 54
   Link Training Configuration 58
   Downloading Pattern Sequence to Hardware 60

3  Programming Examples

   Introduction 62

   Initializing the Connection 62
   N4903B Initialization 62
   M8020A Initialization 63

   SJ Example 64
   N4903B SJ Example 64
   M8020A SJ Example 66

   PJ Example 70
   N4903B PJ Example 70
   M8020A PJ Example 72
Sampling Point Alignment Example 76
  N4903B Sampling Point Alignment Example 76
  M8020A Sampling Point Alignment Example 78

BER Example 80
  N4903B BER Example 80
  M8020A BER Example 81

M8070A Jitter Tolerance Measurement CSV Output 85

4 SCPI Command Language

SCPI Command Language - Introduction 88
  SCPI Common Commands 88
  SCPI Instrument Control Commands 88
  IEEE 488.2 Mandatory Commands 89
  Overlapped and Sequential Commands 89
  Data Types 90

Important Points about SCPI 91
  Important Points about SCPI - Concepts 91

Sending Commands to the M8020A/M8030A/M8040A 95
  Sending Commands to the M8020A/M8030A/M8040A - Overview 95
  Sending Commands using VISA 95

Command Line Arguments 96

Communication 97
5 SCPI Command Reference

Subsystems 100

IEEE Commands - Reference 102
   Mandatory Commands 102

SCPI Standard Commands 108

Miscellaneous Commands 109

Command Syntax to Find Min/Max Values 110

Location and Module Mapping - Reference 111

STATus Subsystem 112
   :STATus:QUEStionable Subnode 113
   :STATus:OPERation Subnode 122
   :STATus:INSTrument Subnode 126
   :STATus:INSTrument:ALIGnment Subnode 131

TRIGger Subsystem 133

Source Subsystem 140
   [:SOURce]:PULSe Subnode 144
   [:SOURce]:VOLTage Subnode 149
   [:SOURce]:JITTer Subnode 154
      [:SOURce]:JITTer:HFRequency Subnode 157
      [:SOURce]:JITTer:HFRequency:PERiodic Subnode 160
      [:SOURce]:JITTer:HFRequency:BUNCorrelate Subnode 163
      [:SOURce]:JITTer:HFRequency:RANDom Subnode 167
      [:SOURce]:JITTer:HFRequency:SPECtrally Subnode 171
      [:SOURce]:JITTer:LFRequency Subnode 174
      [:SOURce]:JITTer:LFRequency:PERiodic Subnode 176
      [:SOURce]:JITTer:LFRequency:RSSClocking Subnode 178
      [:SOURce]:JITTer:SWEep Subnode 180
Contents

[:SOURce]:SSCLocking Subnode 187
[:SOURce]:INTerference Subnode 193
  [:SOURce]:INTerference:RANDom Subnode 203
  [:SOURce]:INTerference:SINusoidal Subnode 206
[:SOURce]:CONFigure Subnode 210

OUTPut Subsystem 211
  :OUTPut:DEEmphasis Subnode 220
  :OUTPut:EINSertion Subnode 228
  :OUTPut:TRIGger Subnode 232
  :OUTPut:DEMBedding Subnode 234
  :OUTPut:EMBedding Subnode 237

EVENt Subsystem 240

INPut Subsystem 242
  :INPut:CDR Subnode 252
    :INPut:CDR:OJTF Subnode 260
  :INPut:ALIGNment Subnode 262
    :INPut:ALIGNment:STATus Subnode 268
  :INPut:EQUalization Subnode 270
    :INPut:EQUalization:FFEQualizer Subnode 273

SENSe Subsystem 276
CLOCk Subsystem 277
N1000 Subsystem 281
FETCH Subsystem 290
MMEMory Subsystem 295
ACQuisition subsystem 301
Contents

**PLUGin Subsystem** 304
- :PLUGin:ERATiо Subnode 306
- :PLUGin:OTIMing Subnode 328
  - :PLUGin:OTIMing:FETCh Subnode 348
  - :PLUGin:OLEVeI Subnode 359
  - :PLUGin:OLEVeI:ACQuisition Subnode 365
  - :PLUGin:OLEVeI:EVAluation Subnode 370
  - :PLUGin:OLEVeI:FETCh Subnode 372
  - :PLUGin:OLEVeI:FETCh:LEVel Subnode 374
  - :PLUGin:OLEVeI:FETCh:NOISе Subnode 377
  - :PLUGin:OLEVeI:FETCh:QLEVel Subnode 380
  - :PLUGin:OLEVeI:SHOW Subnode 385
  - :PLUGin:OLEVeI:RUN Subnode 388
- :PLUGin:JTOLerances Subnode 392
- :PLUGin:EDiagram Subnode 423
  - :PLUGin:EDiagram:ACQuisition Subnode 429
  - :PLUGin:EDiagram:EVAluation Subnode 433
  - :PLUGin:EDiagram:FETCh[:RESult] Subnode 435
  - :PLUGin:EDiagram:RUN Subnode 441
  - :PLUGin:EDiagram:SHOW Subnode 444
- :PLUGin:PSWEep Subnode 449
  - :PLUGin:PSWEep:ACQuisition Subnode 456
  - :PLUGin:PSWEep:RUN Subnode 466
  - :PLUGin:PSWEep:SHOW Subnode 470
- :PLUGin:CCAPture Subnode 473
  - :PLUGin:CCAPture:FETCh Subnode 480
  - :PLUGin:CCAPture:ACQuisition Subnode 485
  - :PLUGin:CCAPture:RUN Subnode 490

**SYSTem Subsystem** 493
Contents

**DATA Subsystem**  511
  :DATA:PATTerr Subnode  512
  :DATA:SEQunce Subnode  520
  :DATA:SYNC Subnode  529
  :DATA:LINecoding Subnode  531
  :DATA:FEC Subnode  536

Index
1 Programming Basics

Modular Configuration / 16
Using Identifiers / 23
PLUGin Subsystem / 25
Modular Configuration

In order to describe remote programming for the M8020A, M8030A and M8040A, the modular configuration must be discussed.

The M8020A, M8030A and M8040A are modular instruments. They can consist of several numbers of components. A component can be an AXIe chassis, modules, channels, etc. All these components 'form' an instrument. These instruments can be controlled by M8070A system software.

The M8020A instrument supports the following modules.
- M8041A high-performance BERT generator-analyzer-clock 8/16 Gb/s
- M8051A high-performance BERT generator-analyzer 8/16 Gb/s
- M8061A multiplexer 2:1 with de-emphasis 32 Gb/s
- M8062A 32Gb/s Front-end for J-BERT M8020A High-Performance BERT

The M8020A modules must be installed in the M9505A 5-slot chassis.

The M8030A instrument supports the following modules.
- M8041A high-performance BERT generator-analyzer-clock 8/16 Gb/s
- M8051A high-performance BERT generator-analyzer 8/16 Gb/s
- M8192A Multi-channel synchronization module

The M8030A modules must be installed in the M9514A AXIe 14-slot chassis.

For complete details on the features and hardware components of each of the above mentioned modules, refer to M8020A and M8030A Getting Started Guide.

M8020A Modular Configuration

The following section describes and illustrates various setup combinations in which you can install the M8020A modules.

The following configurations are possible in an M9505A 5-slot chassis:
- 1 or 2-channel, 16 Gb/s - (1) M8041A
- 3 or 4-channel, 16 Gb/s - (1) M8041A + (1) M8051A
- 1-channel, 32 Gb/s (Pattern Generator only) - (1) M8041A + (1) M8061A
- 1-channel, 32 Gb/s (Pattern Generator only or full BERT) - (1) M8041A + (1) M8062A
1 or 2-channel System Configuration

The 1 or 2-channel configuration is a single channel system (a second channel can be added with license) consisting of the 5-slot M9505A AXIe Chassis and an M8041A module. The M8041A occupies three slots. A maximum of two M8020A modules can be installed in a 5-slot chassis.

3 or 4-channel System Configuration

The four channel configuration consists of the 5-slot M9505A AXIe Chassis, an M8041A module, and an M8051A module. The M8041A occupies three slots and the M8051A occupies two slots.

NOTE

This configuration requires a cable (provided with the M8051A) that connects the M8041A SYNC OUT to the M8051A SYNC IN to synchronize the two modules to a common system clock.
1-channel, 32 Gb/s (Pattern Generator Only)

A typical configuration using the M8061A 32 Gb/s multiplexer with de-emphasis consists of the 5-slot M9505A AXIe Chassis, an M8041A module, and an M8061A module. The M8041A occupies three slots and the M8061A occupies two slots.
1-channel, 32 Gb/s (Pattern Generator only or full BERT)

A typical configuration for an M8020A 32 Gb/s full BERT consists of the 5-slot M9505A AXIe Chassis, an M8041A module, and an M8062A module. The M8041A occupies three slots and the M8062A occupies two slots.

![M8020A configuration for 32 Gb/s BERT](image)

M8030A Modular Configuration

The M8030A is a modular test solution which can be tailored to specific needs from two channels with one M8041A to up to 10 channels. The modules must be installed in the M9514A AXIe 14-slot chassis as shown in Table 3 on page -19:

<table>
<thead>
<tr>
<th>Slot Number</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td># 1</td>
<td>For M8030A-BU1, M9537A AXIe embedded controller. For M8030A-BU2, this slot is empty and covered with filler front-panel</td>
</tr>
<tr>
<td># 2-4</td>
<td>M8041A module</td>
</tr>
<tr>
<td># 5-6</td>
<td>M8051A module</td>
</tr>
<tr>
<td># 7</td>
<td>M9521A AXIe system module, always included in M8030A-BU1 or M8030A-BU2, mandatory</td>
</tr>
<tr>
<td># 8-9</td>
<td>M8051A module</td>
</tr>
</tbody>
</table>
Figure 5 on page -20 shows an example of modules arrangement in the M9514A AXIe 14-slot chassis.

<table>
<thead>
<tr>
<th>Slot Number</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td># 10-11</td>
<td>M8051A module</td>
</tr>
<tr>
<td># 12-13</td>
<td>M8051A module</td>
</tr>
<tr>
<td># 14</td>
<td>M8192A multi-channel synchronization module, mandatory</td>
</tr>
</tbody>
</table>
M8040A Modular Configuration

The M8040A is a modular test solution which simplifies accurate receiver characterization of devices operating up to 32 and 64 GBaud with NRZ and PAM4. It supports the following modules.

- M8045A pattern generator
- M8046A analyzer
- M8057A remote head

The following section describes and illustrates various setup combinations in which you can install the M8040A modules.

M8045A Pattern Generator Module

The M8045A module can be a one or two data channel system (a second channel can be added with license). A one channel instrument has to be returned to the factory for installing the second channel (hardware) and license. It occupies three slots of the 5-slot M9505A AXIe chassis. The following figure illustrates an M8045A module two data channel system installed in an M9595A AXIe chassis.

NOTE

The M8045A module must be installed in slots 1 through 3 of the M9505A AXIe chassis for proper local bus communication unless the M9537A AXIe Embedded Controller is installed (must be in slot 1).
M8046A Analyzer Module

The M8046A module occupies single slot of the 5-slot M9505A AXIe chassis. The following figure illustrates an M8045A module with M8046A module installed in an M9505A AXIe chassis.

M8057A Remote Head

The M8057A remote head is an external box which can be connected to each channel of M8045A module, using the matched pair of cables. It helps in minimizing signal degradations caused by lossy channels. The following figure illustrates an M8057A remote head connected with one channel of M8045A module.
Using Identifiers

There are three different types of Identifiers:

- Location
- Group Name
- Measurement Name

Location Identifiers

When an instrument has several output channels with identical capabilities or a subset of capabilities and has more than one module, the selection of which module/channel to use is done using its location identifier.

The location identifier is a predefined notation that can be used to address certain properties, like high level, low level etc., of channels. The location identifier is surrounded by single quotes ('identifier') or double quotes ("identifier"). A location identifier will be appended to a SCPI command followed by its argument(s). The location identifier addresses the corresponding components. The location identifier consists of two parts: the first part specifies the module, the second part addresses, in most cases, a visible ‘component’ of a module, like an input or output connector.

For example, SCPI command :OUTP 'M1.DataOut1', OFF turns channel 1 (DataOut1) of module 1 (M1) output off.

If a SCPI command is to be sent to a specific module only and not a specific channel, simply use "M1" or "M2" as the location identifier.

If you want to set a value that acts on all components of a modular instrument (for example, set all jitter sources to off), omit the location identifier.

Multiple Source Per Channel Identifier

Some features have multiple sources per channel. To identify these sources, the "(*)" suffix is used.

For example, there are two periodic jitter sources per channel in a module. The following SCPI command specifies periodic jitter source 2 as the source for module 1/data out channel 1:

[:SOURce]:JITTer:PERiodic2[:STATe] 'm1.dataout1',ON
Group Name Identifiers

Group name identifiers are used to affect a group of inputs or outputs using a single SCPI command. That is, all properties belonging to these identifiers can be addressed simultaneously using the group name.

The following SCPI command shows the syntax for creating a group name then adding the desired location identifiers.

```
SYSTem:INSTRument:GROup:DEFine
  'GroupName','Identifier','Identifier','Identifier','...' 
```

The following is an example showing 'Outputs' as the group name followed by 'M1.DataOut1' and 'M1.DataOut2' location identifiers assigned to the 'Outputs' group name identifier.

```
```

The following SCPI command will set all voltage amplitudes in the 'Outputs' group to 0.05.

```
[:SOURce]:VOLTage[:AMPLitude]VOLT:AMPL 'Outputs',0.05
```

Measurement Name Identifiers

Measurement name identifiers are used to set up parameters associated with a specific measurement name. For example, the following shows the bit error ratio plugin SCPI command used to define a measurement name called 'MyMeasurement':

```
:PLUGin:ERATio:NEW 'MyMeasurement'
```

You can now set up parameters within the :PLUGin:ERATio subsystem to be associated with the 'MyMeasurement' identifier.

The following shows the accumulation duration mode (set to fixed time) and duration (set to 120 seconds) parameters associated with the 'MyMeasurement' identifier.

```
PLUGin:ERATio:ACQuisition:DURation 'MyMeasurement', FTIM
PLUGin:ERATio:ACQuisition:TIME 'MyMeasurement',120
```
The M8020A/M8030A/M8040A platform supports a plugin interface. C# assemblies which implement certain interfaces are recognized by the software and integrated into the M8070A GUI and instrument software. Error ratio, Output Timing and Jitter Tolerance measurements are examples of the plugin concept. These C# assemblies are placed in a predefined location and are integrated into the M8070A software at start up time automatically.

These measurements can be controlled by the GUI and by SCPI commands. This means that the SCPI tree has to be defined by the plugin itself and is integrated into the existing SCPI tree of the M8020A/M8030A/M8040A platform. SCPI commands for a plugin are placed below a predefined node named ".:PLUGin" followed by the type of the plug in.

There is a set of "general" SCPI commands for creating, deleting and controlling a plugin. Also, there is a flexible set of SCPI commands provided by the plugin itself for setting and reading measurement parameters.

Additionally, a user can create multiple measurements of the same type. A measurement name identifier is used to address a created measurement.
Recommended Programming Techniques

- Output Protection / 28
- Controlling the Output Levels / 29
- Reading the M8020A/M8030A/M8040A’s Status / 30
- Reading the M8020A/M8030A/M8040A’s Status / 30
- Working with Patterns / 38
- Creating Pattern Sequences / 45
Output Protection

Pattern Generator Output Termination

The pattern generator's output ports must be terminated with 50 Ω if they are not connected. Termination of output ports improves the test performance.

Refer also to the M8020A/M8030A/M8040A Getting Started Guide for information on terminating the M8061A outputs to ensure proper performance.

Output Protection Algorithm

The instrument has an internal protection algorithm that protects the instrument from improper termination of the pattern generator's output ports.

The algorithm checks whether the termination voltage and termination resistance are correct before the output is enabled. This check happens once only.

In non-balanced dc coupled mode, the termination voltage must be within ± 0.1 V of the specified value. The resistance must be between 40 and 60 Ω.

In balanced mode, the resistance must be between 80 and 120 Ω (differential termination).

In ac coupled mode, which is intended for balanced only (for example, 8b/10b coded patterns), the output is enabled if no termination voltage and no termination resistance are detected. This is the case if ac capacitors are in front of the outputs. The check distinguishes between a terminated or an open output.

Do not operate the output driver into an open.
Controlling the Output Levels

Controlling the Output Levels – Concepts

When the output levels are changed at the M8020A/M8030A/M8040A data and clock output ports, four parameters are changed:

- \( V_{\text{high}} \)
- \( V_{\text{low}} \)
- \( V_{\text{ampl}} \)
- \( V_{\text{offs}} \)

The M8020A/M8030A groups these parameters into "pairs" (\( V_{\text{ampl}}/V_{\text{offs}} \), \( V_{\text{high}}/V_{\text{low}} \)). If one of these values is modified, its "partner" remains constant and the values in the other pair are modified accordingly. For example, if \( V_{\text{ampl}} \) is changed, \( V_{\text{offs}} \) stays constant and \( V_{\text{high}} \) and \( V_{\text{low}} \) are modified accordingly.

Controlling the Output Levels – Procedures

Changing the Voltages with SCPI

The following commands show how you would set the data output so that it has an amplitude of 1 V and an offset of 0.5 V:

```
:SOUR:VOLT:AMPL 'M1.DataOut1',1; OFFS 'M1.DataOut1',0.5
```

This sets the output accordingly (\( V_{\text{high}} = 1 \text{ V}, V_{\text{low}} = 0 \text{ V} \)). Conversely, you could set \( V_{\text{high}} \) and \( V_{\text{low}} \) directly:

```
:SOUR:VOLT:HIGH 'M1.DataOut1',1;
LOW 'M1.DataOut1',0
```
Reading the M8020A/M8030A/M8040A’s Status

Overview of Registers

Specifically, it has the following registers:

- Status Byte
  The Status Byte is a single register that stores the events occurring in the other registers.

- Standard Event Status Register
  The Standard Event Status Register monitors some non-critical errors and basic operations.

- Questionable Data Status Register
  The bits in the Questionable Data Status Register are set when certain events occur in the M8020A/M8030A/M8040A that can lead to questionable results.

- Operation Status Register
  The Operation Status Register indicates when certain operations have been completed.

How the M8020A/M8030A/M8040A Uses Status Registers

You can determine the state of certain instrument hardware and firmware events and conditions by programming the status register system.

The following subsections provide you with details about the M8020A/M8030A/M8040A's status system.

Overview of the M8020A/M8030A/M8040A’s Status System

The M8020A/M8030A/M8040A has status reporting features that give important information about events and conditions within the instrument. For example, a flag may be set to indicate the end of a measurement or perhaps a command error. To access this information, it is necessary to query a set of registers using SCPI.
M8020A/M8030A/M8040A Status System Structure

The M8020A/M8030A/M8040A's status system is comprised of multiple registers that are arranged in a hierarchical order. The lower-level status registers propagate their data to the higher-level registers in the data structures by means of summary bits. The Status Byte register is at the top of the hierarchy and contains general status information for the M8020A/M8030A/M8040A's events and conditions. All other individual registers are used to determine the specific events or conditions.

Status Register Group Model

Figure 9 on page -31 illustrates the typical structure of a status register. The M8020A/M8030A/M8040A instrument status register model follows the structure described in IEEE 488.2, section 11.4.2.

A condition is a device state which is either TRUE or FALSE. A condition register reflects these states in its condition bits.

A Condition Register is embedded in a register structure consisting of Transition Filters, an Event Register and an Event Enable Register.
This register is read by the CONDition? SCPI commands.

**Transition Filters:** A positive transition filter allows an event to be reported when a condition changes from false to true. A negative transition filter allows an event to be reported when a condition changes from true to false. Setting both positive and negative filters true allows an event to be reported anytime the condition changes. Clearing both filters disables event reporting.

This register is set and read by the NTRansition[?] and PTRansition[?] SCPI commands.

**Event Register:** An event register latches transition events from the condition register as specified by the positive and negative transition filters. Bits in the event register are latched and once set, they remain set until cleared by either querying the register contents or sending the *CLS command.

This register is read by the EVENt? SCPI commands.

**Event Enable Register:** An enable register specifies the bits in the event register that can generate a summary bit. Summary bits are, in turn, used by the next higher register.

This register is set and read by the ENABle[?] SCPI commands.

The registers work together as follows:

1. The **Condition Register** corresponds to a condition on the hardware or in the software. If the monitored condition is present, the corresponding bit is high.

2. The **Transition Registers** monitor changes in the Condition Register. If the Positive Transition Register is configured to watch for a condition, when this condition occurs and the bit in the Condition Register goes high, the Positive Transition Register passes this event to the **Event Register**.

3. If this bit is enabled in the **Event Enable Register**, a summary bit is generated in the next higher register. For the higher register, this is the Condition Register and the event is handled the same way as described here.
M8020A/M8030A/M8040A Status Reporting Structure

Figure 10 on page 33 shows the M8020A/M8030A/M8040A’s status reporting structure.

Figure 10 Status reporting structure
Status Byte

The Status Byte is the summary register to which the other registers report. Each reporting register is assigned a bit in the Status Byte Register.

The bits in the Status Byte byte have the following meaning:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>EAV</td>
<td>Error available: the error queue contains at least one message.</td>
</tr>
<tr>
<td>3</td>
<td>QUES</td>
<td>A bit has been set in the Questionable Data Status register (indicates that a signal is of questionable quality).</td>
</tr>
<tr>
<td>4</td>
<td>MAV</td>
<td>Message available: There is at least one message in the message queue.</td>
</tr>
<tr>
<td>5</td>
<td>ESB</td>
<td>A bit in the Standard Event Register has been set.</td>
</tr>
<tr>
<td>6</td>
<td>RQS</td>
<td>ReQuest Service.</td>
</tr>
<tr>
<td>7</td>
<td>OPER</td>
<td>A bit in the Operation Status Register has been set.</td>
</tr>
</tbody>
</table>
Standard Event Status Register

The Standard Event Status register provides general-purpose information about the instrument. It sets bit 5 in the Status Byte.

<table>
<thead>
<tr>
<th>Bit</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Operation Complete</td>
<td>Operation Complete bit. It is set in response to the *OPC command, but only if the instrument has completed all its pending operations.</td>
</tr>
<tr>
<td>1</td>
<td>Request Control</td>
<td>Request Control bit.</td>
</tr>
<tr>
<td>2</td>
<td>Query Error</td>
<td>Query error bit. It indicates that there is a problem with the output data queue. There has been an attempt to read the queue when it is empty, the output data has been lost, or the query command has been interrupted.</td>
</tr>
<tr>
<td>3</td>
<td>Device Dependent Error</td>
<td>Device-dependent error bit. It is set when an instrument-specific error has occurred.</td>
</tr>
<tr>
<td>4</td>
<td>Execution Error</td>
<td>Execution error bit. It is set when a command (GPIB instrument specific) cannot be executed due to an out of range parameter or some instrument condition that prevents execution.</td>
</tr>
<tr>
<td>5</td>
<td>Command Error</td>
<td>Command error bit. It is set whenever the instrument detects an error in the format or content of the program message (usually a bad header, missing argument, or wrong data type etc.).</td>
</tr>
<tr>
<td>6</td>
<td>User Request</td>
<td>User Request bit.</td>
</tr>
<tr>
<td>7</td>
<td>Power On</td>
<td>Power-on bit. It is set each time the instrument is powered from off to on.</td>
</tr>
</tbody>
</table>
Questionable Status Register

The Questionable Status Register indicates that a currently running process is of questionable quality. The output of this register sets bit 3 of the Status Byte.

<table>
<thead>
<tr>
<th>Bit</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 8</td>
<td>Not used</td>
<td>This bit indicates the status of certain hardware components.</td>
</tr>
<tr>
<td>9</td>
<td>INSTRument</td>
<td>This bit indicates the status of certain hardware components.</td>
</tr>
<tr>
<td>10</td>
<td>Not used</td>
<td>This bit indicates whether one or more of the instrument outputs detect an overload condition.</td>
</tr>
<tr>
<td>11</td>
<td>PROTection</td>
<td>This bit indicates whether the error detector(s) has experienced a symbol alignment loss.</td>
</tr>
<tr>
<td>12</td>
<td>SYMBol</td>
<td>This bit indicates whether the error detector(s) has experienced a symbol alignment loss.</td>
</tr>
<tr>
<td>13 - 15</td>
<td>Not used</td>
<td></td>
</tr>
</tbody>
</table>

Operation Status Register

The output of this register gives information about the current operation the instrument is performing. It sets bit 7 of the Status Byte.

<table>
<thead>
<tr>
<th>Bit</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 7</td>
<td>Not used</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>RUN</td>
<td></td>
</tr>
<tr>
<td>9 - 15</td>
<td>Not used</td>
<td></td>
</tr>
</tbody>
</table>
Program Snippet Extending the *opc? Functionality

Whenever changes are done that require the hardware to stop clock generation, a "*opc?" command does not wait long enough. It only waits until the clock generation is stopped and not when the clock generation is again started. You have to wait until the hardware is in run condition again (clocks are again running). In general, once you perform these kind of operation, you should execute "*opc?" command to make sure whether the operation is completed or not.

This can be polled in the status system, e.g. with a program snippet like this:

```csharp
var timeStart = DateTime.Now;
while ((response != "1") && ((DateTime.Now - timeStart).TotalSeconds < 60))
Assert.AreEqual("1", response);
```

This program snippet can be used after the following operations:

- Frequency change or anything related. Everything that is in the M8070A GUI under "ClkGen" and "Synthesizer" functional blocks.
- SSC parameter change (everything that is in the M8070A GUI under "ClkGen", "SSC" functional blocks or the global SSC on/off)
- Changes in the sequence symbol width
- *rst or any other reset command or setting load, as they change all parameters.
Working with Patterns

Patterns can be accessed with a program and are stored in three different areas:
1. Local to current setting ("current:")
2. Shared between settings ("shared:")
3. Factory supplied standard patterns ("factory:"). These patterns are read only and cannot be modified.

Below these root nodes is a folder structure using ‘/’ as a separation character. So a complete pattern name might be something like "factory:SATA/LTDP-short".

Patterns consist of a sequence of symbols. A symbol has a coding:
1. Binary ("Bit")
2. 8b10b ("B8B10")
3. 128/130 ("B128B130")
4. 128/132 ("B128B132")

A symbol can have additional attributes to modify behavior. All symbols support:
1. Mask: this attribute affects DataIn locations only. It specifies if the symbol is actually compared or masked (excluded from compare).
2. Squelch: this attribute affects DataOut locations only. If this attribute is 1, a squelch (out of band) level is used.

In addition, there are coding specific attributes, to control aspects of the coding such as bypassing or using a scrambler. Refer to :DATA:PATTern:IDATa[?] on page 514 for more information.

The :DATA:PATTern:USE SCPI command is used to specify the patterns and attributes. Refer to :DATA:PATTern:USE[?] on page 513 for details on how to specify a pattern and set up its parameters. The parameters include the pattern name, number of symbols, symbol coding (optional), mask (optional) and squelch (optional).
Symbol Coding Descriptions

Bit Coding

To encode a bit coded (binary) symbol 1, 2, or 3 bits are needed. This depends on the use of mask and squelch.

Table 8  Plain bit coding without using mask or squelch

<table>
<thead>
<tr>
<th>Bit Offset/Range</th>
<th>Bit (range) Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Data</td>
<td>Data bit</td>
</tr>
</tbody>
</table>

Table 9  Mask is used

<table>
<thead>
<tr>
<th>Bit Offset/Range</th>
<th>Bit (range) Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Data</td>
<td>Data bit</td>
</tr>
<tr>
<td>1</td>
<td>Mask</td>
<td>Mask (ignored on DataOut)</td>
</tr>
</tbody>
</table>

Table 10  Squelch is used

<table>
<thead>
<tr>
<th>Bit Offset/Range</th>
<th>Bit (range) Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Data</td>
<td>Data bit</td>
</tr>
<tr>
<td>1</td>
<td>Squelch</td>
<td>Squelch (ignored on DataIn)</td>
</tr>
</tbody>
</table>

Table 11  Mask and squelch are used

<table>
<thead>
<tr>
<th>Bit Offset/Range</th>
<th>Bit (range) Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Data</td>
<td>Data bit</td>
</tr>
<tr>
<td>1</td>
<td>Mask</td>
<td>Mask (ignored on DataOut)</td>
</tr>
<tr>
<td>2</td>
<td>Squelch</td>
<td>Squelch (ignored on DataIn)</td>
</tr>
</tbody>
</table>
### Table 12  Error is used

<table>
<thead>
<tr>
<th>Bit Offset/Range</th>
<th>Bit (range) Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Data</td>
<td>Data bit</td>
</tr>
<tr>
<td>1</td>
<td>Error</td>
<td>Error (ignored on DataOut, ignored on DataIn)</td>
</tr>
</tbody>
</table>
**8b10b Coding**

To encode an 8b10b symbol 16 bits (2 bytes) are used.

### Table 13  8b10b coding

<table>
<thead>
<tr>
<th>Bit Offset/Range</th>
<th>Bit (range) Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:0</td>
<td>Data</td>
<td>Symbol data</td>
</tr>
<tr>
<td>8</td>
<td>K/D</td>
<td>0 = D-character, 1 = K-character</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>Reserved for future use. Must be set to 0</td>
</tr>
<tr>
<td>10</td>
<td>Mask</td>
<td>Mask (if present, ignored on DataOut)</td>
</tr>
<tr>
<td>11</td>
<td>Squelch</td>
<td>Squelch (if present, ignored on DataIn)</td>
</tr>
<tr>
<td>12</td>
<td>Enable scrambler</td>
<td>Enable scrambler (ignored on DataIn)</td>
</tr>
<tr>
<td>13</td>
<td>Pause scrambler</td>
<td>Pause scrambler (ignored on DataIn)</td>
</tr>
<tr>
<td>14</td>
<td>Reset scrambler</td>
<td>Reset scrambler (ignored on DataIn)</td>
</tr>
<tr>
<td>15</td>
<td>Start of Frame</td>
<td>Start of Frame (ignored on DataOut)</td>
</tr>
</tbody>
</table>

### Table 14  Variant for captured pattern

<table>
<thead>
<tr>
<th>Bit Offset/Range</th>
<th>Bit (range) Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7:0</td>
<td>Data</td>
<td>Symbol data</td>
</tr>
<tr>
<td>8</td>
<td>K/D</td>
<td>0 = D-character, 1 = K-character</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>Reserved for future use. Must be set to 0</td>
</tr>
<tr>
<td>10</td>
<td>Error</td>
<td>Error (ignored on DataOut, ignored on DataIn)</td>
</tr>
<tr>
<td>15:11</td>
<td>Squelch</td>
<td>Reserved for future use. Must be set to 0</td>
</tr>
</tbody>
</table>
### 128/130 Coding

To encode 128/130 symbol 144 bits (18 bytes) are used.

<table>
<thead>
<tr>
<th>Bit Offset/Range</th>
<th>Bit (range) Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:0</td>
<td>Framing</td>
<td>Framing bits</td>
</tr>
<tr>
<td>129:2</td>
<td>Data</td>
<td>Data</td>
</tr>
<tr>
<td>130</td>
<td>Mask</td>
<td>Mask (if present, ignored on DataOut)</td>
</tr>
<tr>
<td>131</td>
<td>Squelch</td>
<td>Squelch (if present, ignored on DataIn)</td>
</tr>
<tr>
<td>132</td>
<td>Reset Scrambler</td>
<td>Reset scrambler (ignored on DataIn)</td>
</tr>
<tr>
<td>133</td>
<td>Pause scrambler</td>
<td>Pause scrambler (ignored on DataIn)</td>
</tr>
<tr>
<td>134</td>
<td>Bypass scrambler</td>
<td>Scrambler bypass</td>
</tr>
<tr>
<td>135</td>
<td>Bypass byte 0 scrambler</td>
<td>Scrambler bypass byte 0</td>
</tr>
<tr>
<td>136</td>
<td>Do DC balancing</td>
<td>Do DC balancing (ignored on DataIn)</td>
</tr>
<tr>
<td>137</td>
<td>Reset DC balancing</td>
<td>Reset DC balancing (ignored on DataIn)</td>
</tr>
<tr>
<td>138</td>
<td>Send scrambler state</td>
<td>Send scrambler state</td>
</tr>
<tr>
<td>139</td>
<td>Reset Parity</td>
<td>Reset Parity (ignored on DataIn)</td>
</tr>
<tr>
<td>140</td>
<td>Pause Parity</td>
<td>Pause Parity (ignored on DataIn)</td>
</tr>
<tr>
<td>141</td>
<td>Start of Frame</td>
<td>Start of Frame (ignored on DataOut)</td>
</tr>
<tr>
<td>143:142</td>
<td>Reserved for future use</td>
<td>Reserved for future use. Must be set to 0</td>
</tr>
</tbody>
</table>
128/132 Coding

To encode a 128/132 symbol 144 bits (18 bytes) are used.

Table 16  128/132 coding

<table>
<thead>
<tr>
<th>Bit Offset/Range</th>
<th>Bit (range) Name</th>
<th>Coding-128/132 content</th>
</tr>
</thead>
<tbody>
<tr>
<td>3:0</td>
<td>Framing</td>
<td>Framing bits</td>
</tr>
<tr>
<td>131:4</td>
<td>Data</td>
<td>Data</td>
</tr>
<tr>
<td>132</td>
<td>Mask</td>
<td>Mask (if present, ignored on DataOut)</td>
</tr>
<tr>
<td>133</td>
<td>Squelch</td>
<td>Squelch (if present, ignored on DataIn)</td>
</tr>
<tr>
<td>134</td>
<td>Reset Scrambler</td>
<td>Scrambler reset (ignored on DataIn)</td>
</tr>
<tr>
<td>135</td>
<td>Pause Scrambler</td>
<td>Scrambler pause (ignored on DataIn)</td>
</tr>
<tr>
<td>136</td>
<td>Bypass Scrambler</td>
<td>Scrambler bypass</td>
</tr>
<tr>
<td>137</td>
<td>Bypass Byte 0 Scrambler</td>
<td>Scrambler bypass byte 0</td>
</tr>
<tr>
<td>138</td>
<td>Do DC Balancing</td>
<td>Do DC balancing (ignored on DataIn)</td>
</tr>
<tr>
<td>139</td>
<td>Reset DC Balancing</td>
<td>Rest DC balancing (ignored on DataIn)</td>
</tr>
<tr>
<td>140</td>
<td>Send Scrambler State</td>
<td>Send scrambler state (ignored on DataIn)</td>
</tr>
<tr>
<td>141</td>
<td>Reset Parity</td>
<td>Reset parity (ignored on DataIn)</td>
</tr>
<tr>
<td>142</td>
<td>Pause Parity</td>
<td>Pause parity (ignored on DataIn)</td>
</tr>
<tr>
<td>143</td>
<td>Start of Frame</td>
<td>Start of frame (ignored on DataOut)</td>
</tr>
</tbody>
</table>
Symbol Sequences

When combining symbols to a sequence of symbols they are filled in one after the other without any gaps. For the non-bit codings that use multiple bytes, the symbols naturally start again on a byte boundary. For the bit codings (binary) they start at the next available bit. Bits are used in a fashion that allows reading from "left to right". This means the symbol bit 0 is encoded as bit 7 in a byte.

<table>
<thead>
<tr>
<th>Byte</th>
<th>Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 11  Example 1: plain bit coding

<table>
<thead>
<tr>
<th>Byte</th>
<th>Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 12  Example 2: mask and squelch bit coding
Creating Pattern Sequences

A pattern sequence is configured using a language. The pattern sequence language is based on XML.

Sequence Editor

Pattern sequences are created and edited using the Sequence Editor in the M8070A software interface as shown in Figure 14 on page 45.

Figure 13 Example 3: 8b10b coding

Figure 14 Sequence Editor in M8070A software interface
A default “Generator” and “Analyzer” sequence are configured in the Sequence Editor as shown in Figure 14 on page -45. You can edit these sequences or create new ones.

As a pattern sequence is being configured, the corresponding XML elements are generated and can be viewed in the XML editor by clicking on the <Xml> button shown in Figure 15 on page -46.

Figure 15 <Xml> button

Figure 16 on page -46 shows the XML elements for the default Generator and Analyzer. Clicking the UI button will return to the graphical interface.

In UI view, click in a pattern sequence block to display its sequence settings as shown in Figure 17 on page -47.
Pattern Sequence Building Blocks

**Minimal Sequence**

A minimalistic sequence looks like the following:

```xml
<sequenceDefinition
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.agilent.com/schemas/M8000/DataSequence">

</sequenceDefinition>
```

This sequence generates a 0 signal to all assigned locations.

**Description**

To specify a description for the sequence, insert the following:

```xml
<description>Default 2^7-1 PRBS Sequence</description>
```

**Version Number**

To specify a version number for the sequence syntax, insert the following:

```xml
<version>1.0.0</version>
```
Symbol Width

To specify the symbol encoding this sequence works with, insert the following:

```xml
<symbolWidth>1</symbolWidth>
```

Available symbol widths are:

- 1: this is for bit coded symbols
- 10: this is for 8b/10b coded and bit coded symbols
- 130: for 128/130 coded and bit coded symbols
- 132: for 128/132 coded and bit coded symbols

Pattern Generation Granularity

Pattern generation has granularity. A single bit cannot be generated. Instead, multiple bits are generated at once. Compared to other BERTs like the J-BERT N4903B, different granularities are supported to allow for less restrictions in the pattern generation. Also, granularity is hidden as much as possible by unrolling patterns internally.

<table>
<thead>
<tr>
<th>Symbol Width</th>
<th>Granularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>64</td>
</tr>
<tr>
<td>10</td>
<td>80</td>
</tr>
<tr>
<td>130</td>
<td>130</td>
</tr>
<tr>
<td>132</td>
<td>132</td>
</tr>
</tbody>
</table>

The actual pattern sequence attributes are specified within:

```xml
<sequence replicate="Serialize"></sequence>
```

Replicate allows global selection of how to distribute PRBS or memory patterns to multiple locations.

- Serialize: first bit/symbol is assigned to first location, next bit/symbol to next location and so on. This means a pattern needs to have a minimum block length \* locations bits. In the case of a PRBS, a PRWS is issued, setting the phases of the PRBS generators in a way that they will form a PRBS after a MUX.
- Copy: each location gets a copy of the PRBS /pattern.
Blocks and Loops

The basic building blocks of a sequence are blocks and loops.

A block is the basic unit of a sequence. Here patterns are assigned to locations or branches and are specified to other blocks. A block transitions to the next as a default action. But it also might be enclosed in a loop. Nested loops are also possible. There are three counted loop levels plus an infinite loop level:

- Loop level 1 can be used to repeat a block.
- Loop level 2 and 3 can span multiple blocks, but it must be ensured that either the start or end of a loop is on a fresh block.
- Loop level 4 is for infinite loops.

The following illustrates valid loop nesting with the full 4 loop levels:

![Diagram of loop nesting]

Figure 18 Loop nesting

Count attribute specifies the number of loop iterations. The valid value range is 1 to $2^{32}$. If omitted, infinite is assumed. The enabled attribute allows the complete loop, including all loop content, to be temporarily disabled. If omitted, the default is enabled.

```xml
<loop count="1" enabled="1"></loop>
```
Length Attribute

The length attribute specifies the “length” of the block in number of bits. The enabled attribute allows the block to be temporarily disabled (ignored). If omitted, the default is enabled.

```xml
<block length="1" enabled="1"/>
```

Synchronization and Loop Block

A special combination of sync and loop is available for achieving synchronization:

```xml
<syncAndLoopBlock length="100" enabled="1"/></syncAndLoopBlock>
```

Other attributes for block or syncAndLoopBlock:
1. name: for naming a block.
2. cdr: for freezing CDR when CDR control is “sequence”.
3. errorInsertion: for suppressing error insertion.
4. disparity: for setting the disparity in 8b10b coding.
5. startWordAlign: for initiating a word alignment in 8b10b and 128/130 coding. In the syncAndLoopBlock this flag is used for the initial block only, not for the looped block.

Pattern Selection

Within a block or syncAndLoopBlock, select the type of patterns to use. The choices are clock, pulse, Prbs, static and memory pattern. The XML elements for pattern selection are as follows:

```xml
<prbs polynomial="2^7-1" seed="127" invert="0" replicate="Serialize" enabled="1"/></prbs>
```

```xml
<pattern source="factory:SATA/LTDP-short_b8b10" replicate="Serialize" enabled="1"/></pattern>
```

```xml
<static value="0" compare="1"/>
```

```xml
<pulse offset="0" width="10" compare="1" squelch="1"/>
```

```xml
<clock divider="2" compare="1"/>
```

```xml
<linkTraining direction="up"/>
```
Sequencer Triggers

Trigger output pulses can be generated and controlled from sequence blocks. These pulses are sent to the front panel TRIG OUT connectors.

This requires that the trigger mode be set to sequencer controlled. The following describes how to change the trigger mode.

1. In the M8070A software interface, select System > Module View.
2. Click on Trig Out.
3. Under Configuration parameters, change Operating Mode to Sequencer Controlled.

Refer to :OUTPut:TRIGger:MODE[?] on page 232 for information about setting the trigger mode using this SCPI command.

Two variants are available:

1. A trigger pulse can be generated relative to a sequence block.
2. A trigger pulse can be generated whenever a specific bit sequence is generated by a PRBS generator.

This is controlled via nodes within a block node.

NOTE

If multiple sequences are used, only one sequence can actually have control of the trigger.

Triggering Relative to a Sequence Block

The following is an example of a trigger pulse relative to a sequencer block:

```xml
<trigger>
  <pulse offset="0" width="32"/>
</trigger>
```

The offset can be omitted and defaults to 0. The width can also be omitted and defaults to 32.
Triggering Based on Specific Bit Sequence

The following is an example of a trigger pulse generated based on a specific PRBS bit sequence:

```
<trigger>
  <pulseOnPrbsMatch width="32" matchPattern="0000000" polynomial="2^7-1" seed="127" invert="0"/>
</trigger>
```

All attributes can be omitted. The width defaults to 32. The match pattern defaults to all 0's or all 1's depending on the invert attribute. The polynomial defaults to the polynomial used by the generators or analyzers. Invert also defaults to the PRBS invert setting of the generators or analyzers.

Block Branching

Within a block a conditional branch to another block can be done. The most simple form is as follows:

```
<if source="Break"/>
```

This example simply jumps to the next block when a break command is done. The following describes how to set up a break command.

1. In the M8070A software interface, select Patterns > Sequence Editor.
2. Under the Block Branches settings, change Condition to Break.

Refer to :DATA:SEQuence:BREak on page 526 for information on setting the break command using this SCPI command.

When a branch to a different block is desired, use the "goto" attribute:

1. In the M8070A software interface, select Patterns > Sequence Editor.
2. Under the Block Branches settings, click in the Go To Block field and enter the block name.

Other potential conditions are SysInA, SysInB, CtrlInA, or CtrlInB. Break, SysInA and SysInB are global events and affect all locations of a sequence. CtrlInA and CtrlInB are local to a module and are most suitable when all locations of a sequence are using a single module.
It is also possible to specify a reaction based on a static level or a transition. Using the “event” attribute, you can select “low”, “high”, “positiveEdge” and “negativeEdge”.

1. In the M8070A software interface, select Patterns > Sequence Editor.
2. Under the Block Branches settings, change Transition to Positive, Negative, Static High, or Static Low.

### Link Training Events

- **Link Training Target State**: indicates that the Link Training is in the target state (that is loopback)
  ```xml
  <if source="LinkTrainingTargetState"/>
  ```

- **Link Training Detect State**: indicates that the Link Training is in the detect state (that is the reset state)
  ```xml
  <if source="LinkTrainingDetectState"/>
  ```

- **Link Training Error**: indicates that an error occurred when trying to bring up the link (that is Link Training reached the target state but not the target speed)
  ```xml
  <if source="LinkTrainingError"/>
  ```

### Block Controls

Within a block it can be specified to generate a signal on CTRL OUT A, SYS OUT A, or SYS OUT B front panel connector. This can be useful for triggering an external device such as an oscilloscope or to generate auxiliary signals such as a laser enable signal. However, be aware that there is no specified timing relationship to the DataOut.

```xml
<control sink="CtrlOutA" sinkLocation="M1" value="pulse"/>
```

In a two module configuration, the sinkLocation attribute is used to specify on what module the CTRL OUT A signal is generated.

The shape of the signal is specified with the value attribute. It can be a static 0, static 1, or a pulse.

---

**NOTE**

SysOutA and SysOutB are internally controlled from the Trigger Sequencer.

Therefore, these signals can only be used in the same sequence where the trigger out is controlled.
Global Block Controls

Global block controls are used to set up global routing of signals. Signals can be routed to the CTRL OUT A front panel connector only.

One common use could be for generating a pulse on CTRL OUT A whenever an error on the M1.DataIn1 or M1.DataIn2 of the corresponding module occurs:

```xml
<controls>
  <control sink="CtrlOutA" source="Error"
    sourceLocation="M1.DataIn1"/>
</controls>
```

With the additional "signalShape" attribute, it can be selected if a 0/1 pulse is generated whenever an error occurs.

Coding Configuration

Various aspects of coding can be configured. This is done with the `<codingConfigurations>` node. It is inserted just before the `<symbolWidth>` node. Within this node the 8b10b specific attributes are configured in a `<b8b10>` node. The align symbol is configured with an optional attribute. If not specified the default align symbol is K28.5. Also, you can specify a substitution symbol in an attribute that is inserted into the received data stream when an invalid code word is recognized. The default for this is K28.0. So the complete node would look like the following:

```xml
<b8b10 align="K28.5" substitution="K28.0"/>
```

And can be shortened to the following:

```xml
<b8b10/>
```

Up to four filler primitives can be specified. Each filler primitive can consist of up to 4 symbols, where a symbol can be any valid 8b10b symbol or a '*' that is used as a wildcard matching any symbol.

A filler symbol specification looks like the following:

```xml
<fillerPrimitive symbol1="K28.1" symbol2="K28.1"/>
```
After specifying a filler primitive, various aspects of the (de)scrambler can be configured with the `<scrambler>` node. With the polynomial attribute the USB/PCIe or SATA polynomial can be selected. The reset value for the scrambler and the reset value for the scrambler after a hold sequence can be configured with the `resetValue` and `resetValueAfterHold` attribute. In addition, it can be specified if the scrambler is reset after a filler primitive was removed with the `resetAfterFillerRemove` attribute. All these nodes are optional, with defaults suitable for USB testing. If just `<scrambler>` node is given without attributes, it is equivalent to the following:

```xml
<scrambler polynomial="USB/PCIe" resetValue="65515" resetValueAfterHold="6143" resetAfterFillerRemove="false"/>
```

Within the scrambler node, sequences of up to 4 symbols can be configured that control the scrambler when they are seen in the incoming data stream. With `resetPrimitive` the scrambler is reset, with `holdPrimitive` it is paused and with `pauseStartPrimitive` it is paused until a `pauseEndPrimitive` is seen.

The following are examples for the different standards.

**USB 3.0**

```xml
<codingConfigurations>
  <b8b10>
    <fillerPrimitive symbol1="K28.1" symbol2="K28.1"/>
    <scrambler>
      <resetPrimitive symbol1="K28.5"/>
    </scrambler>
  </b8b10>
</codingConfigurations>
```
Recommended Programming Techniques

PCIe 1/2

```xml
<codingConfigurations>
  <b8b10>
    <fillerPrimitive symbol1="K28.5" symbol2="K28.0"/>
    <fillerPrimitive symbol1="K28.0"/>
  </b8b10>
  <scrambler>
    <resetPrimitive symbol1="K28.5"/>
  </scrambler>
</codingConfigurations>
```

SATA

```xml
<codingConfigurations>
  <b8b10>
    <fillerPrimitive symbol1="K28.5" symbol2="D10.2" symbol3="D10.2" symbol4="D27.3"/>
    <fillerPrimitive symbol1="K28.0"/>
  </b8b10>
  <scrambler polynomial="SATA" resetValue="61686" resetValueAfterHold="30349">
    <resetPrimitive symbol1="K28.3" symbol2="D21.5" symbol3="D23.1" symbol4="D23.1"/>
    <holdPrimitive symbol1="K28.3" symbol2="*" symbol3="*" symbol4="*"/>
    <pauseStartPrimitive symbol1="K28.3" symbol2="*" symbol3="*" symbol4="*"/>
    <pauseEndPrimitive symbol1="K28.3" symbol2="*" symbol3="*" symbol4="*"/>
  </scrambler>
</codingConfigurations>
```
Recommended Programming Techniques

MIPI Phy

```xml
<codingConfigurations>
  <b8b10>
    <fillerPrimitive symbol1="K28.1"/>
  </b8b10>
</codingConfigurations>

SAS

```xml
<codingConfigurations>
  <b8b10>
    <fillerPrimitive symbol1="K28.5" symbol2="D10.2" symbol3="D27.3"/>
    <fillerPrimitive symbol1="K28.5" symbol2="D7.0" symbol3="D7.0" symbol4="D7.0"/>
    <fillerPrimitive symbol1="K28.5" symbol2="D1.3" symbol3="D1.3" symbol4="D1.3"/>
    <fillerPrimitive symbol1="K28.5" symbol2="D27.3" symbol3="D27.3" symbol4="D27.3"/>
  </b8b10>
</codingConfigurations>
Link Training Configuration

Link Training Configuration describes the parameters controlling the LTSSM (Link Training and Status State Machine). In the first release it is only defined and implemented for PCI Express 3.0. This is done with the <linkTrainingConfigurations> node. It is inserted just before the <codingConfigurations> node. The following parameters can be configured as attributes.

Table 18  Link Training Configuration

<table>
<thead>
<tr>
<th>Name</th>
<th>Possible Values</th>
<th>Default Value</th>
<th>Description</th>
<th>Used for DUT Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUT Type</td>
<td>AddInCard, SystemBoard</td>
<td>AddInCard</td>
<td>Specifies which role the DUT should play during link training. It can either be an upstream device for testing a downstream port or vice versa.</td>
<td>Both</td>
</tr>
<tr>
<td>Link Equalization</td>
<td>Bypass, PresetsOnly, Full</td>
<td>Bypass</td>
<td>Determines whether link equalization should be performed. It can either be aborted after phase 1 (Bypass) or fully executed. In the second case it can be determined whether only preset or all (that is, individual cursor) requests should be accepted.</td>
<td>Both</td>
</tr>
<tr>
<td>Lane Number</td>
<td>0 - 15</td>
<td>0</td>
<td>This is the lane number being used.</td>
<td>Both</td>
</tr>
<tr>
<td>Link Number</td>
<td>0 - 255</td>
<td>0</td>
<td>This is the link number being used.</td>
<td>Add-in Card</td>
</tr>
<tr>
<td>Start Preset</td>
<td>P0 - P9</td>
<td>P4</td>
<td>This is the preset used by the BERT's TX port after switching to Gen 3 operation and when operating as an upstream device.</td>
<td>Add-in Card</td>
</tr>
</tbody>
</table>
The following is an example of a Link Training Configuration for PCIe 3.0:

```xml
<linkTrainingConfigurations>
  <pcie3 dut="AddInCard" lane="0" link="0" linkEQ="PresetsOnly" generatorStartPreset="P4"
    dutRxPresetHint="Reserved" dutTxInitialPreset="P0"
    dutTxTargetPreset="P7" calibratedPresets="true" />
</linkTrainingConfigurations>
```
Recommended Programming Techniques

Downloading Pattern Sequence to Hardware

Once a sequence has been configured, it is downloaded to the hardware using the download button in the Sequence Editor shown in Figure 19 on page 60. When the sequence has been downloaded, this button turns green.

![Download button](image)

Figure 19 Download button

You can also download the pattern sequence configuration by copying and pasting the XML elements as a sequence string using the :DATA:SEQUence:VALue SCPI command. For an example, refer to :DATA:SEQUence[:SET]:VALue[?] on page 523.
Keysight M8000 Series of BER Test Solutions
Programming Guide

3 Programming Examples

Introduction / 62
Initializing the Connection / 62
SJ Example / 64
PJ Example / 70
Sampling Point Alignment Example / 76
BER Example / 80
M8070A Jitter Tolerance Measurement CSV Output / 85
Introduction

This chapter provides programming examples for setting up some of the most common functions including:

- Initialization
- SJ
- PJ
- Sampling point alignment
- BER

If you have programmed the N4903B, examples are provided for comparison to the M8020A. This will facilitate code conversion from the N4903B to the M8020A.

Initializing the Connection

N4903B Initialization

The ResourceName ("TCPIP1::10.0.0.207::inst0::INSTR") must be replaced by the instrument's address string from the VISA Assistant.

First our declarations...

Private myN490X As AgilentN490x
Private myBERT As AgilentBert
Private myPG As AgilentBertLib.IAgilentBertPG
Private myPGClock As AgilentBertLib.IAgilentBertPGClock
Private myPGOut As AgilentBertLib.IAgilentBertPGOutput
Private myEDDataIn As AgilentBertLib.IAgilentBertEDDataInput
Private Sub Form_Load()
  Set myN490X = New AgilentN490x
  Set myBERT = myN490x.IAgilentBert
  myBERT.Initialize ("TCPIP1::10.0.0.207::inst0::INSTR", True, True, "QueryInstrStatus=true")
End Sub

Private Sub Form_Unload(Cancel As Integer)
  myBERT.Close
End Sub
namespace baseInstrument.instruments
{
    class M8020A : BERT
    {
        private string m_PG_channel = "M1.DataOut1";
        private string m_ED_channel = "M1.DataIn1";
        private string m_system = "M1.ClkGen";
        double m_ED_clk = 16.0e9;
        double m_PG_clk = 16.0e9;
        double m_prevBER = 0.0;
        double m_prevBitCount = 0.0;
        double m_prevErrorCount = 0.0;
        double m_prevTimeStamp = 0.0;
        
        public M8020A(string resourceName, string module = "M1", string channel = "1") : base(resourceName)
        {
            if (module.IndexOf("MUX") == -1)
            {
                m_system = module + ".ClkGen";
                m_PG_channel = module + ".DataOut" + channel + ";";
                m_ED_channel = module + ".DataIn" + channel + ";";
            }
            else
            {
                m_system = "M1.ClkGen";
                m_PG_channel = "MUX";
            }
        }
    }
}
#region SJ
public override bool getSJEenabled()
    // SOUR8:SIN?
    {
        if ((int.Parse(this.Query("*:SOUR8:SIN?"))) == 1)
            return true;
        else
            return false;
    }

public override double getSJFreq()
    // SOUR8:SIN:FREQ?
    {
        return double.Parse(this.Query("*:SOUR8:SIN:FREQ?"));
    }

public override double getSJFreqMax()
    // SOUR8:SIN:FREQ? MAX
    {
        return double.Parse(this.Query("*:SOUR8:SIN:FREQ? MAX"));
    }

public override double getSJFreqMin()
    // SOUR8:SIN:FREQ? MIN
    {
        return double.Parse(this.Query("*:SOUR8:SIN:FREQ? MIN"));
    }

public override List<string> setSJEenabled(bool b)
    // SOUR8:SIN 1 => ON
    // SOUR8:SIN 0 => OFF
    {
        string strValue = "";

        if (b) strValue = "1";
        else strValue = "0";

        return this.Send("*:SOUR8:SIN: " + strValue);
    }
public override List<string> setSJFreq(double SJFreq)
// SOUR8:SIN:FREQ
{
    string strValue = "";
    if ((this.getSJFreqMax() > SJFreq) && (this.getSJFreqMin() <= SJFreq))
        strValue = SJFreq.ToString();
    else
        strValue = this.getSJFreq().ToString();
    return this.Send(":.SOUR8:SIN:FREQ * + strValue");
}

public override double getSJAmp()
// SOUR8:SIN:LEV?
{
    return double.Parse(this.Query(":.SOUR8:SIN:LEV?"));
}

public override double getSJAmpMax()
// SOUR8:SIN:LEV? MAX
{
    return double.Parse(this.Query(":.SOUR8:SIN:LEV? MAX"));
}

public override double getSJAmpMin()
// SOUR8:SIN:LEV? MIN
{
    return double.Parse(this.Query(":.SOUR8:SIN:LEV? MIN"));
}

public override List<string> setSJAmp(double SJAmp)
// SOUR8:SIN:LEV
{
    string strValue = "";
    if ((this.getSJAmpMax() > SJAmp) && (this.getSJAmpMin() <= SJAmp))
        strValue = SJAmp.ToString();
    else
        strValue = this.getSJAmp().ToString();
    return this.Send(":.SOUR8:SIN:LEV * + strValue");
}
#endregion
M8020A SJ Example

```csharp
// region SJ - implemented for M8020A
public override bool getSJEnabled()
// M8020A
{
    string str = "";
    str = this.Query("*:SOUR:JITT:LFR:PER:STAT? " + m_PG_channel);
    if (str.IndexOf("1") != -1)
    {
        return true;
    }
    else
    {
        return false;
    }
}

public override double getSJFreq()
// M8020A
{
    string retString = "";
    double val = 0.0;
    retString = this.Query("*:SOUR:JITT:LFR:PER:FREQ? " + m_PG_channel);
    val = double.Parse(retString);
    return val;
}

public override double getSJFreqMax()
// M8020A
{
    string retString = "";
    double val = 0.0;
    retString = this.Query("*:SOUR:JITT:LFR:PER:FREQ? " + m_PG_channel + ",MAX");
    val = double.Parse(retString);
    return val;
}
```
    public override double getSJFreqMin()
    // M8020A
    {
        string retString = "";
        double val = 0.0;

        retString = this.Query(":\SO;\JIT;\LFR;\PER;\FREQ? \+ m_PG_channel + ",MIN");
        val = double.Parse(retString);

        return val;
    }

    public override List<string> setSJEnabled(bool b)
    // M8020A
    {
        string strValue = "";

        if (b) strValue = "ON";
        else strValue = "OFF";

        return this.Send(":\SO;\JIT;\LFR;\PER;\STAT \+ m_PG_channel + ",\+ strValue");
    }

    public override List<string> setSJFreq(double SJFreq)
    // M8020A
    {
        string strValue = "";

        SJFreq = Math.Round(SJFreq, 3);

        if ((this.getSJFreqMax() > SJFreq) && (this.getSJFreqMin() <= SJFreq))
            strValue = SJFreq.ToString();
        else if (this.getSJFreqMax() < SJFreq)
            strValue = this.getSJFreqMax().ToString();
        else if (this.getSJFreqMin() > SJFreq)
            strValue = this.getSJFreqMin().ToString();
        else
            strValue = this.getSJFreq().ToString();

        return this.Send(":\SO;\JIT;\LFR;\PER;\FREQ \+ m_PG_channel + ",\+ strValue +"Hz");
    }
public override double getSJAnp()
// M8020A
{
    string retString = "";
    string unit = "";
    double val = 0.0;

    unit = this.Query(":\SOUR:JITT:LFR:UNIT? * + m_PG_channel");
    this.Send(":\SOUR:JITT:LFR:UNIT " + m_PG_channel + ",UINT");
    retString = this.Query(":\SOUR:JITT:LFR:PER:AMPL? * + m_PG_channel");
    this.Send(":\SOUR:JITT:LFR:UNIT " + m_PG_channel + "," + unit);
    val = double.Parse(retString);
    return val;
}

public override double getSJAnpMax()
// M8020A
{
    string retString = "";
    string unit = "";
    double val = 0.0;

    unit = this.Query(":\SOUR:JITT:LFR:UNIT? * + m_PG_channel");
    this.Send(":\SOUR:JITT:LFR:UNIT " + m_PG_channel + ",UINT");
    retString = this.Query(":\SOUR:JITT:LFR:PER:AMPL? * + m_PG_channel + ",MAX");
    this.Send(":\SOUR:JITT:LFR:UNIT " + m_PG_channel + "," + unit);
    val = double.Parse(retString);
    return val;
}
public override double getSJAmplMin()
// M8020A
{
    string retString = "";
    string unit = "";
    double val = 0.0;

    unit = this.Query("*:SOUR:JITT:LFR:UNIT? " + m_PG_channel);
    this.Send("*:SOUR:JITT:LFR:UNIT " + m_PG_channel + ",UINT");
    retString = this.Query("*:SOUR:JITT:LFR:PER:AMPL? " + m_PG_channel + ",MIN");
    this.Send("*:SOUR:JITT:LFR:UNIT " + m_PG_channel + "," + unit);
    val = double.Parse(retString);
    return val;
}

public override List<string> setSJAmpl(double SJAmpl)
// M8020A
{
    string strValue = "";
    string unit = "";
    double amp = 0.0;

    if (m_PG_channel == "MUX")
        SJAmpl /= 2.0;
    if ((this.getSJAmpMax() > SJAmpl) && (this.getSJAmpMin() <= SJAmpl))
        amp = SJAmpl;
    else
        amp = this.getSJAmp();

    amp = Math.Round(amp, 3);
    strValue = amp.ToString();

    unit = this.Query("*:SOUR:JITT:LFR:UNIT? " + m_PG_channel);
    this.Send("*:SOUR:JITT:LFR:UNIT " + m_PG_channel + ",UINT");
    this.Send("*:SOUR:JITT:LFR:PER:AMPL " + m_PG_channel + "," + strValue);
    return this.Send("*:SOUR:JITT:LFR:UNIT " + m_PG_channel + "," + unit);
}
#endregion
PJ Example

N4903B PJ Example

```csharp
#region PJ
public override bool getPJEnabled()
// SOUR8:PER?
{
    if ((int.Parse(this.Query(".SOUR8:PER?"))) == 1)
        return true;
    else
        return false;
}

public override double getPJFreq()
// SOUR8:PER:FREQ?
{
    return double.Parse(this.Query(".SOUR8:PER:FREQ?"));
}

public override double getPJFreqMax()
// SOUR8:PER:FREQ? MIN MAX
{
    return double.Parse(this.Query(".SOUR8:PER:FREQ? MAX"));
}

public override double getPJFreqMin()
// SOUR8:PER:FREQ? MIN MAX
{
    return double.Parse(this.Query(".SOUR8:PER:FREQ? MIN"));
}

public override List<string> setPJEnabled(bool b)
// SOUR8:PER 1 ==> ON
// SOUR8:PER 0 ==> OFF
{
    string strValue = "";
    if (b) strValue = "1";
    else strValue = "0";
    return this.Send(".SOUR8:PER: " + strValue);
}
```
public override List<string> setPJFreq(double PJFreq)
// SOUR8:PER:FREQ
{
    string strValue = "";
    if ((this.getPJFreqMax() > PJFreq) && (this.getPJFreqMin() <= PJFreq))
        strValue = PJFreq.ToString();
    else if (this.getPJFreqMax() < PJFreq)
        strValue = this.getPJFreqMax().ToString();
    else if (this.getPJFreqMin() > PJFreq)
        strValue = this.getPJFreqMin().ToString();
    else
        strValue = this.getPJFreq().ToString();
    return this.Send(":SOUR8:PER:FREQ " + strValue);
}

public override double getPJAmp()
// SOUR8:PER:LEV?
{
    return double.Parse(this.Query("~SOUR8:PER:LEV?"));
}

public override double getPJAmpMax()
// SOUR8:PER:LEV? MAX
{
    return double.Parse(this.Query("~SOUR8:PER:LEV? MAX"));
}

public override double getPJAmpMin()
// SOUR8:PER:LEV? MIN
{
    return double.Parse(this.Query("~SOUR8:PER:LEV? MIN"));
}

public override List<string> setPJAmp(double PJAmp)
// SOUR8:PER:LEV
{
    string strValue = "";
    if ((this.getPJAmpMax() > PJAmp) && (this.getPJAmpMin() <= PJAmp))
        strValue = PJAmp.ToString();
    else
        strValue = this.getPJAmp().ToString();
    return this.Send("~SOUR8:PER:LEV " + strValue);
}
#region
M8020A PJ Example

```csharp
    #region PJ - as PJ1 implemented for M8020A
    public override bool getPJEnabled()
    // M8020A - checked
    {
        string str = "";
        str = this.Query("":SOUR:JITT:HFR:PER1:STAT? " + m_PG_channel);
        if (str.IndexOf("1") != -1)
            return true;
        else
            return false;
    }

    public override double getPJFreq()
    // M8020A - checked
    {
        string retString = "";
        double val = 0.0;
        retString = this.Query("":SOUR:JITT:HFR:PER1:FREQ? " + m_PG_channel);
        val = double.Parse(retString);
        return val;
    }

    public override double getPJFreqMax()
    // M8020A - checked
    {
        string retString = "";
        double val = 0.0;
        retString = this.Query("":SOUR:JITT:HFR:PER1:FREQ? " + m_PG_channel + ",MAX");
        val = double.Parse(retString);
        return val;
    }
```

public override double getPJFreqMin()
// M8020A - checked
{
    string retString = "";
    double val = 0.0;

    retString = this.Query(":\SOUR\:JITT\:HFR\:PER1\:FREQ? " + m_PG_channel + ",MIN");
    val = double.Parse(retString);

    return val;
}

public override List<string> setPJEnabled(bool b)
// M8020A - checked
{
    string strValue = "";

    if (b) strValue = "ON";
    else strValue = "OFF";

    return this.Send(":\SOUR\:JITT\:HFR\:PER1\:STAT " + m_PG_channel + "," + strValue);
}

public override List<string> setPJFreq(double PJFreq)
// M8020A - checked
{
    string strValue = "";

    PJFreq = Math.Round(PJFreq, 3);

    if (this.get PJFreqMax() > PJFreq) && (this.get PJFreqMin() <= PJFreq))
    strValue = PJFreq.ToString();
    else if (this.get PJFreqMax() < PJFreq)
    strValue = this.get PJFreqMax().ToString();
    else if (this.get PJFreqMin() > PJFreq)
    strValue = this.get PJFreqMin().ToString();
    else
    strValue = this.get PJFreq().ToString();

    return this.Send(":\SOUR\:JITT\:HFR\:PER1\:FREQ " + m_PG_channel + "," + strValue + "Hz");
}
public override double getPJAmpl()
// M8020A - checked
{
    string retString = "";
    string unit = "";
    double val = 0.0;
    unit = this.Query("*:SOUR:JITT:HFR:UNIT? " + m_PG_channel);
    this.Send("*:SOUR:JITT:HFR:UNIT " + m_PG_channel + ",UINT");
    retString = this.Query("*:SOUR:JITT:HFR:PER1:AMPL? " + m_PG_channel);
    this.Send("*:SOUR:JITT:HFR:UNIT " + m_PG_channel + "," + unit);
    val = double.Parse(retString);
    return val;
}

public override double getPJAmplMax()
// M8020A - checked
{
    string retString = "";
    string unit = "";
    double val = 0.0;
    unit = this.Query("*:SOUR:JITT:HFR:UNIT? " + m_PG_channel);
    this.Send("*:SOUR:JITT:HFR:UNIT " + m_PG_channel + ",UINT");
    retString = this.Query("*:SOUR:JITT:HFR:PER1:AMPL? " + m_PG_channel + ",MAX");
    this.Send("*:SOUR:JITT:HFR:UNIT " + m_PG_channel + "," + unit);
    val = double.Parse(retString);
    return val;
}

public override double getPJAmplMin()
// M8020A - checked
{
    string retString = "";
    string unit = "";
    double val = 0.0;
    unit = this.Query("*:SOUR:JITT:HFR:UNIT? " + m_PG_channel);
    this.Send("*:SOUR:JITT:HFR:UNIT " + m_PG_channel + ",UINT");
    retString = this.Query("*:SOUR:JITT:HFR:PER1:AMPL? " + m_PG_channel + ",MIN");
    this.Send("*:SOUR:JITT:HFR:UNIT " + m_PG_channel + "," + unit);
    val = double.Parse(retString);
    return val;
}
public override List<string> setPJAmp(double PJamp)
// M8020A - checked
{
    string strValue = "";
    string unit = "mV";
    double amp = 0.0;

    if (m_PG_channel == "MUX")
        PJamp /= 2.0;

    if ((this.getPJAmpMax() > PJamp) && (this.getPJAmpMin() <= PJamp))
        amp = PJamp;
    else
        amp = this.getPJAmp();

    amp = Math.Round(amp, 3);
    strValue = amp.ToString();

    unit = this.Query(":SOUR:JITT:HFR:UNIT? " + m_PG_channel);
    this.Send("::SOUR:JITT:HFR:UNIT " + m_PG_channel + ",UINT");
    this.Send("::SOUR:JITT:HFR:PER1:AMPL " + m_PG_channel + "," + strValue);

    return this.Send("::SOUR:JITT:HFR:UNIT " + m_PG_channel + "," + unit);
} #endregion
Sampling Point Alignment Example

N4903B Sampling Point Alignment Example

    public bool checkN900()
    {
      return m_N4900;
    }

    # region Sampling Point Alignment
    public override string autoAlign()
    {
      string result = "";
      this.Send(".SENS1:EYE:ALIGN:AUTO 1");
      this.opc();
      do
        {
        result = this.Query(".SENS1:EYE:ALIGN:AUTO?");
        } while ((result != "CS_ABORTED") && (result != "CS_FAILED") && (result != "CS_SUCCESSFUL"));
      return result;
    }

    public override string dataCenter()
    {
      string result = "";
      this.Send(".SENS1:EYE:TCEN 1");
      this.opc();
      do
        {
        result = this.Query(".SENS1:EYE:ALIGN:AUTO?");
        } while ((result != "CS_ABORTED") && (result != "CS_FAILED") && (result != "CS_SUCCESSFUL"));
      return result;
    }

    public override string thresholdCenter()
    {
      string result = "";
      this.Send(".SENS1:EYE:ACEN 1");
      this.opc();
      do
        {
        result = this.Query(".SENS1:EYE:ALIGN:AUTO?");
        } while ((result != "CS_ABORTED") || (result != "CS_FAILED") || (result != "CS_SUCCESSFUL"));
      return result;
    }
public override List<string> sync()
{
    return this.Send("*:SENS1:SYNC 1");
}

#region sampling point
public override List<string> setSamplingPointDelay(double x)
{
    return this.Send("*:INP1:DEL * + x.ToString()" );
}

public override double getSamplingPointDelay()
{
    string str = "";
    double delay = 0.0;

    str = this.Query("*:INP1:DEL?" );
    delay = double.Parse(str);
    return delay;
}

public override List<string> setSamplingPointDelayRel(double x)
{
    double current = 0.0;

    current = this.getSamplingPointDelay();
    x += current;

    return setSamplingPointDelay(x);
}
#endregion
M8020A Sampling Point Alignment Example

public override string autoAlign()
{
    string result = "";

    /*
     * do
     * {result = this.Query(":\STATus:OPERation:RUN:EVENt?");
     * while ((int.Parse(result) % 2) == 0);
     */
    this.Send(":*INP:ALIG:EYE:TCEN " + m_ED_channel);
    this.opc();

    /*
     * do
     * {result = this.Query(":*INP:ALIG:EYE:TCEN " + m_ED_channel);
     * while (result == "")
     */
    return result;
}

public override string dataCenter()
{
    string result = "";

    this.Send(":*INP:ALIG:EYE:TCEN " + m_ED_channel);
    this.opc();

    /*
     * do
     * {result = this.Query(":*INP:ALIG:EYE:TCEN " + m_ED_channel);
     * while (result == "")
     */
    return result;
}
public override string thresholdCenter()
{
    string result = "";
    this.Send(":\INP:ALIG:EYE:ACEN * + m_ED_channel");
    this.opc();
    /*
     * do
     * {
     *     result = this.Query(":\INP:ALIG:EYE:RES:THR? * + m_ED_channel");
     * } while (result == "");
     * /
    return result;
}

public override List<string> sync()
{
    return this.Send(":\DATA:SYNC " + m_ED_channel);
}

#region sampling point - implemented for M8020A
public override List<string> setSamplingPointDelay(double x)
{
    string delay = "";
    delay = (1.0e9 * x).ToString() + "e-9";
    return this.Send(":\INP:DEL " + m_ED_channel + "," + x.ToString());
}

public override double getSamplingPointDelay()
{
    string retString = "";
    double val = 0.0;
    retString = this.Query(":\INP:DEL? " + m_ED_channel);
    val = double.Parse(retString);
    return val;
}
#endregion
# region ber

private double m_N4900_BitCount = 0.0;
private double m_N4900_ErrorCount = 0.0;

public override List<string> resetBERCounter()
{
    m_N4900_BitCount = 0.0;
    m_N4900_ErrorCount = 0.0;
    return this.startBERAcc();
}

public override List<string> setBERCounterResults()
{
    this.stopBERAcc();
    this.startBERAcc();
    double errors = 0.0;
    double bits = 0.0;
    errors = double.Parse(this.Query(":PFETCH:SENS2:ECO?"));
    bits = double.Parse(this.Query(":PFETCH:SENS2:BCO?"));
    m_ErrorCount = m_N4900_ErrorCount + errors;
    m_N4900_ErrorCount = m_ErrorCount;
    m_BitCount = m_N4900_BitCount + bits;
    m_N4900_BitCount = m_BitCount;
    m_BER = m_ErrorCount / m_BitCount;

    return this.ErrorQ();
}

public override List<string> startBERAcc()
{
    return this.Send("*:SENS1:GATE ON");
}

public override List<string> stopBERAcc()
{
    return this.Send("*:SENS1:GATE OFF");
}

public override double getInstantaneousBER()
{
    return double.Parse(this.Query("*:FETC:SENS2:ERAT?"));
}

# endregion
M8020A BER Example

private double m_prevAccBER_M8020 = 0.0;
private double m_prevAccBitCount_M8020 = 0.0;
private double m_prevAccErrorCount_M8020 = 0.0;

public override List<string> startBERAcc()
    // M8020A
    {
        List<string> errorStr = new List<string>();
        errorStr = this.ErrorQ();
        return errorStr;
    }

double m_prevBitCount;

double m_prevErrorCount;

public override List<string> stopBERAcc()
    // M8020A - error counters can't be stopped yet
    {
        List<string> errorStr = new List<string>();
        errorStr = this.ErrorQ();
        return errorStr;
    }

double m_prevTimeStamp;

double m_prevAccBER_M8020;

double m_prevAccBitCount_M8020;

double m_prevAccErrorCount_M8020;

double m_prevBitCount = bitCounter[0];
m_prevErrorCount = bitCounter[1];
m_prevTimeStamp = bitCounter[2];

return errorStr;

public override double getInstantaneousBER()
    // M8020A
    {
        double accumulatedBER = 0.0;
        List<double> result = new List<double>();
        result = this.getBERAccResults();
        accumulatedBER = result[2];
        return accumulatedBER;
    }
    public override List<string> setBERCounterResults()
    // M8020A
    {
        List<double> berResults = new List<double>();
        List<double> bitCounter = new List<double>();
        double bitCount = 0.0;
        double errorCount = 0.0;
        double elapsedTime = 0.0;
        double timeStamp = 0.0;
        double ber = 0.5;
        bitCounter = this.getErrorCounter();
        bitCount = bitCounter[0] - m_prevBitCount;
        errorCount = bitCounter[1] - m_prevErrorCount;
        timeStamp = bitCounter[2];
        elapsedTime = 1000.0 * (timeStamp - m_prevTimeStamp);
        ber = errorCount / bitCount;
        m_BitCount = bitCount;
        m_ErrorCount = errorCount;
        m_BER = ber;
        return this.ErrorQ();
    }
    private List<double> getErrorCounter()
    // M8020A
    {
        List<double> bitCounter = new List<double>();
        double bitCount = 0.0;
        double errorCount = 0.0;
        double timeStamp = 0.0;
        double counted1s = 0.0;
        double counted0s = 0.0;
        double erroneous1s = 0.0;
        double erroneous0s = 0.0;
        string identifier = "";
        string resultStr = "";
        string valStr = "";
        int index = 0;
resultStr = this.Query("*:FETC:BCO? * + m_ED_channel");

// timestamp
index = resultStr.IndexOf("*");
valStr = resultStr.Substring(1, index - 1);
timeStamp = double.Parse(valStr);
resultStr = resultStr.Substring(index + 1, resultStr.Length - index - 1);

// identifier
index = resultStr.IndexOf("*");
valStr = resultStr.Substring(0, index);
identifier = valStr;
resultStr = resultStr.Substring(index + 1, resultStr.Length - index - 1);

// counted 1s
index = resultStr.IndexOf("*");
valStr = resultStr.Substring(0, index);
counted1s = double.Parse(valStr);
resultStr = resultStr.Substring(index + 1, resultStr.Length - index - 1);

// counted 0s
index = resultStr.IndexOf("*");
valStr = resultStr.Substring(0, index);
counted0s = double.Parse(valStr);
resultStr = resultStr.Substring(index + 1, resultStr.Length - index - 1);

// erroneous 1s
index = resultStr.IndexOf("*");
valStr = resultStr.Substring(0, index);
erroneous1s = double.Parse(valStr);
resultStr = resultStr.Substring(index + 1, resultStr.Length - index - 1);

// erroneous 0s
index = resultStr.IndexOf("*");
valStr = resultStr.Substring(0, index);
erroneous0s = double.Parse(valStr);
resultStr = resultStr.Substring(index + 1, resultStr.Length - index - 1);

bitCount = counted0s + counted1s;
//bitCount -= m_prevBitCount;
errorCount = erroneous0s + erroneous1s;
//errorCount -= m_prevErrorCount;

bitCounter.Add(bitCount);
bitCounter.Add(errorCount);
bitCounter.Add(timeStamp);

return bitCounter;

#endregion
NOTE

If the M8070A software is running on Windows 8 or 8.1, the VXI-11 protocol is not supported.

Example:

"SCPI Access (VXI-11): TCPIP0::localhost::inst0::INSTR"

does not work on Windows 8 or 8.1.

In this case, the HiSLIP protocol should be used as shown below:

"SCPI Access (HiSLIP): TCPIP0::localhost::hislip0::INSTR"
M8070A Jitter Tolerance Measurement CSV Output

The following python script generates the results from all open jitter tolerance measurements to different CSV files. This script can be easily modified to generate the output of the highest-passing point at each jitter modulation frequency value measured. This script can be run from the M8070A Script Editor or can also be run externally.

```python
# M8020A jtol CSV output
# Run from the M8070A script editor

def WaitForCompletion():
    res = M8000.Scpi.Query(':syst:err?')
    if (res != '0,"No error"):
        print res
    WaitForCompletion()
M8000.Scpi.Timeout = 120000

# loop through all jitter tolerance measurements
for id in M8000.Scpi.Query(':PLUG:JTOL:CAT?').Split(',):
    print "Measurement: " + id + "":"
    #print results
    # size of results
    print "" + str(len(results)) + " data items, " + str(int(len(results)/6)) + " rows;",
    # if there are any rows (and not just a blank measurement), then
    # use the name of the jtol measurement as the filename
    # this will overwrite the file if it already exists!
    # the file will be located in \Program Files (x86)\Agilent\M8070A\bin\, maybe
    if (int(len(results)/6) == 0):
        # nothing to write to file
        print "no file created"
    else:
        with open(id.Trim(""),Replace("","_.") + ".csv", 'w') as f:
            # output header to file
            f.write("Modulation Frequency (GHz),Sinusoidal Jitter (UI),Number of Error Ratio,Measurement results\n")
```
# iterate over the data
# first element ([0]) is a module identifier, discard
# items are then grouped in bundles of six, per header above
# we need to insert a comma or LF after each item, depending on pos
for idx, data in enumerate(results):
    if (idx==0):
        # do nothing, this is the module identifier
        pass
    elif (idx%6==0):
        # output ending in a newline
        #print data
        f.write(data + "\n")
    else:
        # output ending in a comma
        #print data + ","
        f.write(data + ",")

# close file and move on!
f.close()
print "written to file"
4 SCPI Command Language

SCPI Command Language - Introduction / 88
Important Points about SCPI / 91
Sending Commands to the M8020A/M8030A/M8040A / 95
Command Line Arguments / 96
Communication / 97
SCPI Command Language - Introduction

The Serial BERT is compatible with the standard language for remote control of instruments. Standard Commands for Programmable Instruments (SCPI) is the universal programming language for instrument control.

SCPI can be subdivided into the following command sets:
- SCPI Common Commands
- SCPI Instrument Control Commands
- IEEE 488.2 Mandatory Commands

SCPI Common Commands

This is a common command set. It is compatible with IEEE 488.2 and contains general housekeeping commands. The common commands are always preceded by an asterisk. A typical example is the reset command:

*RST

The IEEE 488.2 command set also contains query commands. Query commands always end with a question mark.

SCPI Instrument Control Commands

The programming commands are compatible with the Standard Commands for Programmable Instruments (SCPI) standard. For more detailed information regarding the GPIB, the IEEE 488.2 standard, or the SCPI standard, refer to the following books:
IEEE 488.2 Mandatory Commands

In order to comply with the SCPI model as described in IEEE 488.2, the Serial BERT implements certain mandatory commands. Other commands are implemented optionally. For more detail on the IEEE 488.2 mandatory and optional commands, see IEEE Commands - Reference on page 102.

Overlapped and Sequential Commands

IEEE 488.2 defines the distinction between overlapped and sequential commands. A sequential command is one which finishes executing before the next command starts executing. An overlapped command is one which does not finish executing before the next command starts executing.

NOTE

It is not reliable to use wait statements in the control program to facilitate the use of overlapped commands.

Because these commands may allow the execution of more than one command at a time, special programming techniques must be used to ensure valid results. The common commands *OPC, *WAI and *OPC? can be used for this purpose. They help synchronize a device controller with the execution of overlapped commands.

The behaviors of these commands, in brief, are as follows:

• *OPC
  The *OPC command sets the Operation Complete (OPC) bit of the Event Register when the No Operation Pending flag is TRUE (No Operation Pending flag is attached to each overlapped command). Until that time, the controller may continue to parse and execute previous commands. It is good technique, then, to periodically poll the OPC bit to determine if the overlapped command has completed.

• *WAI
  The *WAI command allows no further execution of commands or queries until the No Operation Pending flag is true, or receipt of a Device Clear (dcas) message, or a power on. The *WAI command can be used for overlapped commands. It stops the program execution until any pending overlapped commands have finished. Specifically, it waits until the No Operation Pending flag is TRUE, or receipt of a dcas message, or a power on.
• *OPC?  
The *OPC? query returns the ASCII character "1" in the Output Queue when the No Operation Pending flag is TRUE. At the same time, it also sets the Message Available (MAV) bit in the Status Byte Register. The *OPC? will not allow further execution of commands or queries until the No Operation Pending flag is true, or receipt of a Device Clear (dcas) message, or a power on.

NOTE  
The command behaviors described above are for overlapped commands. When the same commands are used with sequential commands, the behaviors may be different.

Data Types  
The M8020A/M8030A/M8040A has the capability of receiving and returning data in the following formats:

STRING  
A string of human-readable ASCII characters, either quoted or nonquoted.

NUMERIC  
The M8020A/M8030A/M8040A handles the following numeric formats:
- <NR1>: Integer (0, 1, 2, -1, etc.)
- <NR2>: Number with an embedded decimal point (0.1, 0.001, 3.3, etc.)
- <NR3>: Number with an embedded decimal point and exponent (1e33, 1.3e-12, etc.)
- <NRf>: Represents <NR1>, <NR2> and <NR3>
- Binary preceded by #b (#b10101, #b011111, etc.)
- Octal preceded by #q (#q777111, #q7331777, etc.)
- Hex preceded by #h (#haff, #h8989ffff, etc.)

BOOLEAN  
Boolean values can be sent to the M8020A/M8030A/M8040A as either ON | OFF or 0 | 1. The M8020A/M8030A/M8040A answers queries with 0 | 1.
**Definite Length Arbitrary Block Data**

Block data is used when a large quantity of related data is being returned. A definite length block is suitable for sending blocks of 8-bit binary information when the length is known beforehand. An indefinite length block is suitable for sending blocks of 8-bit binary information when the length is not known beforehand or when computing the length beforehand is undesirable.

It has the following format:

```
#<Length of length><Length of data><data>
```

`<Length of length>` is a single integer that contains the number of digits in `<Length of data>`, which in turn contains the length of the data. For example, a 512-byte pattern would be defined as:

```
#3512<data>
```

**Important Points about SCPI**

**Important Points about SCPI - Concepts**

There are a number of key areas to consider when using SCPI for the first time. These are as follows:

- Instrument Model
- Command Syntax
- Optional Parts of Commands
- Sending Commands
- Command Separators
- SCPI Command Structure

**Instrument Model**

SCPI guidelines require that the M8020A/M8030A/M8040A is compatible with an instrument model. This ensures that when using SCPI, functional compatibility is achieved between instruments that perform the same tasks. For example, if two different instruments have a programmable clock frequency setting, then both instruments would use the same SCPI commands to set their frequency. The instrument model is made up of a number of subsystems.

The sub-system defines a group of functions within a module and has a unique identifier under SCPI, which is called the Root Keyword.
For more details on the instrument model, see STATus Subsystem on page 112.

**Command Syntax**

Commands may be up to twelve characters long. A short-form version is also available which has a preferred length of four characters or less. In this document the long-form and short-form versions are shown as a single word with the short-form being shown in upper-case letters.

For example, the long-form node command SOURce has the short-form SOUR. Using the short form saves time when entering a program; however, using the long form makes a program more descriptive and easier to understand.

SCPI commands may be commands only, commands and queries, or queries only. A question mark at the end of a command indicates that it is a query. If the question mark appears in brackets ([?]), the command has a command and query form.

**Optional Command Keywords**

Some layers in the SCPI command structure are optional. These optional keywords are indicated by square brackets ([ ]). A typical use for these types of keywords is with a command that is unique to one module. In this case, the top layer (Root Keyword) of the command structure may be omitted.

For example, the following command code segments are functionally identical:

```plaintext
[:SOURce]:JITTer[:GLOBal][:STATe] <ON|OFF|1|0>
:JITTer <ON|OFF|1|0>
:JITT <ON|OFF|1|0>
:jitt <ON|OFF|1|0>
```

Note that it is not necessary to include the syntax inside the square brackets ([ ]).
Query Responses

It is possible to interrogate the individual settings and status of a device using query commands. Retrieving data is a two-stage operation.

The query command is sent from the controller using the OUTPUT statement and the data is read from the device using the ENTER statement. A typical example is the SCPI IEEE 488.2 Common Command *IDN? which queries the identity of a device.

NOTE

When sending strings to the instrument, either the double quote (") or the single quote may be used ('), the latter being more suited to PASCAL programs, which make use of a single quote; the former being more suited to use in BASIC programs, which use a double quote as a delimiter.

Command Separators

The SCPI command structure is hierarchical and is governed by commas, semicolons and colons:

- Commas are used to separate parameters in one command.
- Colons are used to separate levels.
- Semicolons are used to send more than one command to the instrument at a time.

It is possible to send several commands in one pass, as long as the commands all belong to the same node in the SCPI tree. The commands have to be separated by semicolons.

The following SCPI commands provide examples of this.

SOURce:VOLTage:OFFSet 'M2.DataOut2',-0.99
SOURce:VOLTage:AMPLitude 'M2.DataOut2',1.11

These commands can also be sent as follows:

VOLT:OFFS 'M2.DataOut2',-0.99; 'M2.DataOut2',AMPL 1.11
SCPI Command Structure Example

The SCPI command structure can be best examined by means of an example. For example, the command to set the pattern generator’s output amplitude is:

[:SOURce]:VOLTage[:AMPLitude] ‘M1.DataOut1’,1.11

The structure of this command can be illustrated as follows:

| [:SOURce] | This is the top layer of the command structure and identifies the source subsystem. |
| :VOLTage | This is the next layer and defines the subnode for setting a voltage level. |
| [:AMPLitude] | This is the command itself for setting the output amplitude level. |
| ‘M1.DataOut1’,1.11 | This specifies pattern generator 1, channel 1 and specifies an amplitude of 1.11. |

**NOTE**

Any optional commands are enclosed in square brackets [ ] and any optional characters are shown in lower case.

A colon indicates a change of level in the command hierarchy. Commands at the same level in the hierarchy may be included in the same command line, if separated by a semi-colon.

The bar symbol (|) indicates mutually exclusive commands.

To translate this syntax into a command line, follow the convention described above. Remember, however, that the command line can be created in several different ways. It can be created with or without optional keywords and in a long or short form. The following example gives three possible forms of the command line; all are acceptable.

In long form:

:SOURce:VOLTage:AMPLitude ‘M1.DataOut1’,1.11

In short form:

With the optional commands removed:

```
:VOLT 'M1.DataOut1',1.11
```

The long form is the most descriptive form of programming commands in SCPI.

### Sending Commands to the M8020A/M8030A/M8040A

**Sending Commands to the M8020A/M8030A/M8040A - Overview**

A command is invalid and will be rejected if:

- It contains a syntax error.
- It cannot be identified.
- It has too few or too many parameters.
- A parameter is out of range.
- It is out of context.

**Sending Commands using VISA**

The following is a list of the available hardware interfaces for sending commands to the M8020A/M8030A/M8040A firmware:

- SCPI Access (HiSLIP): TCPIP0::localhost::hislip0::INSTR (High-Speed LAN Instrument Protocol)
- SCPI Access (VXI-11): TCPIP0::localhost::inst0::INSTR (VXI-11 is a TCP/IP instrument protocol defined by the VXIbus Consortium)
- SCPI Access (Socket): TCPIP0::localhost::5025::SOCKET (Standard SCPI-over-sockets port)
- SCPI Access (Telnet): telnet localhost 5024 (Communication with LAN instrument through SCPI Telnet port)

**NOTE**

If you use the VXI-11 (TCP/IP instrument protocol) in your test programs, you must change the resource string to the HiSLIP protocol if the software is running on Windows 8 or Windows 8.1. VXI-11 is not supported at this time on Windows 8 or Windows 8.1.

For example:

```
"TCPIP0::192.17.34.0::inst0::INSTR" -> "TCPIP0::192.17.34.0::hislip0::INSTR"
```
Command Line Arguments

(See Communication on page 97 for details about /Socket, /Telnet, /Inst, /AutoID, /NoAutoID, /FallBack).

### Table 19 Command line arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/AutoId</td>
<td>Start in auto ID mode (this is the default) [optional]</td>
</tr>
<tr>
<td>/NoAutoId</td>
<td>Do not start in auto ID mode - use communication parameters from command line [optional]</td>
</tr>
<tr>
<td>/FallBack</td>
<td>Use auto ID mode if the communication parameters from the command line don’t work [optional]</td>
</tr>
<tr>
<td>/Socket &lt;socket port&gt;</td>
<td>Set the socket port for the SCPI communication (only used with /NoAutoId) [optional]</td>
</tr>
<tr>
<td>/Telnet &lt;telnet port&gt;</td>
<td>Set the telnet port for the SCPI communication (only used with /NoAutoId) [optional]</td>
</tr>
<tr>
<td>/Inst &lt;instrument number&gt;</td>
<td>Set the instrument number for VXI-11.3 and HiSLIP SCPI communication [optional]</td>
</tr>
<tr>
<td>/rcl</td>
<td>Recall last used setting [optional]</td>
</tr>
<tr>
<td>/rst</td>
<td>Reset to factory default [optional]</td>
</tr>
<tr>
<td>/IgnoreAwg</td>
<td>M8070A software don’t grab M8195A modules [optional]</td>
</tr>
</tbody>
</table>
Communication

Depending on the command line arguments /Socket, /Telnet, /Inst, /AutoID, /NoAutoID, /FallBack, the M8070A software starts several servers to handle SCPI commands. (Refer to the table above.)

/Socket, /Telnet, /Inst: If -1, don’t start the respective servers

*Defaults:
Socket port: 5025 (e.g. TCPIP0::localhost::5025::SOCKET)
Telnet port: 5024
HiSLIP, VXI-11.3: 0 (e.g. TCPIP0::localhost::hislip0::INSTR, TCPIP0::localhost::inst0::INSTR)

/FallBack: If starting a server fails because of a conflict, try using another port or number

*HiSLIP, VXI-11.3: increase the index until a server can be started successfully

*Socket, Telnet: start with port 60000, then increase it until the servers can be started successfully. If neither socket nor telnet is disabled the M8070A software tries to start the servers on two consecutive ports (socket port = telnet port + 1)

/AutoID: Automatically select ports and number for the connections, which are unique per instrument.

*This is the default behavior; it is not necessary to specify this argument on the command line.

* /Socket, /Telnet, /Inst are ignored (unless they are -1 and a server is disabled)

*If the M8070A software detects more than one AXIe module, use a special mechanism to obtain a number for the HiSLIP and VXI-11.3 servers, which makes sure that the M8070A software always uses the same VISA resource string per module

*The socket and telnet port are then calculated from the HiSLIP index:
telnet port = 60000 + 2 * <HiSLIP index>
socket port = 60000 + 2 * <HiSLIP index> + 1
Note: Ports may already be in use by Windows or other applications, so they are not available for M8195A.

/noAutoID: Do not automatically select ports and number for the connections, use the values specified with /Socket, /Telnet, /Inst or their respective default values instead.

If both /NoAutoID and /AutoID are specified, /AutoID overrides /NoAutoID.

**NOTE**

The first port not assigned by IANA is 49152 (IANA, Internet Assigned Numbers Authority, http://www.iana.org).
Keysight M8000 Series of BER Test Solutions
Programming Guide

5 SCPI Command Reference

Subsystems / 100
IEEE Commands - Reference / 102
SCPI Standard Commands / 108
Miscellaneous Commands / 109
Command Syntax to Find Min/Max Values / 110
Location and Module Mapping - Reference / 111
STATus Subsystem / 112
TRIGger Subsystem / 133
Source Subsystem / 140
OUTPut Subsystem / 211
EVENT Subsystem / 211
INPut Subsystem / 242
SENSe Subsystem / 276
CLOCk Subsystem / 277
N1000 Subsystem / 281
FETCH Subsystem / 290
MMEMory Subsystem / 295
ACQuisition subsystem / 301
PLUGin Subsystem / 304
SYSTem Subsystem / 493
DATA Subsystem / 511
Subsystems

The SCPI commands are divided into subsystems, which reflect the various functionality of the instrument. The following figure shows where the port-related subsystems are located.

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATus</td>
<td>The STATus subsystem reports the current system condition, when an event changes and sets transition filter states to report when a passing event occurs.</td>
</tr>
</tbody>
</table>
| TRIGger   | The TRIGger subsystem selects the following trigger modes:  
- INTERNAL selects the 100 MHz reference clock of the AXIe chassis.  
- REFERENCE selects a 10 MHz/100 MHz external reference clock.  
- DIRECT selects the clock in the M8041A and can be set manually from 8.1 GHz to 16.2 GHz as the system frequency.  
- CMULTIPLIER selects a multiplied or divided clock to specify the system frequency. |
<p>| SOURce    | The SOURce subsystem controls output signals (for example, for setting frequency and levels). |
| OUTPut    | The OUTPut subsystem controls the output ports of the pattern generator. |
| INPut     | The INPut subsystem controls the input ports of the error detector. |
| CLOCK     | The CLOCK subsystem is used to detect clock source or to set the clock frequency applied to input or output channel. |
| FETCH     | The FETCH subsystem is used to query the error detector’s results. |
| MMEMory   | The MMEMory subsystem is used to store and recall predefined or custom created settings and patterns. |
| ACQuisition | The acquisition subsystem controls how a real-time oscilloscope is being configured in order to capture the signal for BER measurements. |</p>
<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLUGin</td>
<td>The PLUGin subsystem is a plugin interface for integrating C# assemblies into the M8070A user interface and instrument software.</td>
</tr>
<tr>
<td>SYSTem</td>
<td>The SYSTem subsystem is used for general system functions.</td>
</tr>
<tr>
<td>DATA</td>
<td>The DATA subsystem is used to select patterns, define symbol parameters, define sequence blocks and loops and synchronize the pattern.</td>
</tr>
</tbody>
</table>
IEEE Commands - Reference

Mandatory Commands

The following mandatory IEEE 488.2 commands are implemented:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>*CLS</td>
<td>*CLS on page 103</td>
</tr>
<tr>
<td>*CAL?</td>
<td>*CAL? on page 103</td>
</tr>
<tr>
<td>*LRN?</td>
<td>*LRN? on page 103</td>
</tr>
<tr>
<td>*ESE[?]</td>
<td>*ESE[?] on page 103</td>
</tr>
<tr>
<td>*ESR?</td>
<td>*ESR? on page 103</td>
</tr>
<tr>
<td>*IDN?</td>
<td>*IDN? on page 104</td>
</tr>
<tr>
<td>*OPC</td>
<td>*OPC on page 104</td>
</tr>
<tr>
<td>*OPC?</td>
<td>*OPC? on page 104</td>
</tr>
<tr>
<td>*OPT?</td>
<td>*OPT? on page 105</td>
</tr>
<tr>
<td>*RCL</td>
<td>*RCL on page 105</td>
</tr>
<tr>
<td>*RST</td>
<td>*RST on page 105</td>
</tr>
<tr>
<td>*SAV</td>
<td>*SAV on page 105</td>
</tr>
<tr>
<td>*SRE[?]</td>
<td>*SRE[?] on page 106</td>
</tr>
<tr>
<td>*STB?</td>
<td>*STB? on page 106</td>
</tr>
<tr>
<td>*TST?</td>
<td>*TST? on page 106</td>
</tr>
<tr>
<td>*WAI</td>
<td>*WAI on page 107</td>
</tr>
</tbody>
</table>
*CLS
Syntax: *CLS
Description: This command clears all status register structures in a device. These registers include:
- OPERation Status Register structure
- QUEStionable Status Register structure
The corresponding enable registers are unaffected.

*CAL?
Syntax: *CAL?
Description: This command returns calibration data.

*LRN?
Syntax: *LRN?
Description: This command gets the device setup query. It return the instrument settings by binary block data.

*ESE[?]?
Syntax: *ESE <NRf>
Description: The Standard Event Status Enable Command (*ESE) sets the Standard Event Enable Register. This register acts like a mask, so that the next time a selected bit goes high, the ESB bit in the status byte is set.

The query (*ESE?) returns the contents of the Standard Event Enable Register.

*ESR?
Syntax: *ESR?
Description: This query returns the Standard Event Status Register content. The register is cleared after it is read.
The identification query (*IDN?) response semantics are organized into four fields, separated by commas. The field definitions are as follows:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturer</td>
<td>Agilent Technologies</td>
</tr>
<tr>
<td>Model</td>
<td>M80xx</td>
</tr>
<tr>
<td>Serial number</td>
<td>MYxxxxxxxxxx</td>
</tr>
<tr>
<td>Firmware level</td>
<td>X.x.x.xxx</td>
</tr>
</tbody>
</table>

The *OPC command sets the Operation Complete (OPC) bit of the Standard Event Status Register (ESR) when the No Operation Pending flag is TRUE (No Operation Pending flag is attached to each overlapped command). Until that time, the controller may continue to parse and execute previous commands. It is good technique, then, to periodically poll the OPC bit to determine if the overlapped command has completed.

The *OPC? query returns the ASCII character "1" in the Output Queue when the No Operation Pending flag is TRUE. At the same time, it also sets the Message Available (MAV) bit in the Status Byte Register. The *OPC? will not allow further execution of commands or queries until the No Operation Pending flag is true, or receipt of a Device Clear (dcas) message, or a power on.
**OPT?**

**Syntax**  
*OPT?

**Description**  
The *OPT? query returns the installed options for each installed module. If no options are installed, only the model number(s) of the installed module(s) are returned.

**Example**  
*OPT?  
(M1-M8041A c08,c16)(M2-M8061A,004,008)

---

**RCL**

**Syntax**  
*RCL

**Description**  
The *RCL command recalls a predefined read only instrument state setting. This command recalls the same device-specific functions effected by the *RST and *SAV commands. The range is *RCL 1 to *RCL 10.

---

**RST**

**Syntax**  
*RST

**Description**  
The Reset Command (*RST) sets the device-specific functions to a known state that is independent of the past-use history of the device.

---

**SAV**

**Syntax**  
*SAV

**Description**  
The *SAV command saves a predefined read only instrument state setting. This command saves the same device-specific functions effected by the *RST and *RCL commands. The range is *SAV 1 to *SAV 10.
**SRE[?]**

**Syntax**  
*SRE[?] <NRf>

**Description**  
The Service Request Enable Command (*SRE) sets the Service Request Enable Register. This acts as a mask on the Status Byte, defining when the instrument can issue a service request. For a service request to be issued, the summary bit in the Status Byte must match the bit in the Service Request Enable Register. More than one bit may be set by the *SRE command.

The query returns the current contents of the Service Request Enable Register.

**STB?**

**Syntax**  
*STB?

**Description**  
The Read Status Byte Query (*STB?) allows the programmer to read the status byte and Master Summary Status bit. When the status byte is read using the *STB command, bit 6 of the status byte is referred to as the Master Summary (MSS) bit. With this query, the status byte is not cleared when the value is read. It always reflects the current status of all the instrument's status registers.

**TST?**

**Syntax**  
*TST?

**Description**  
The self-test query starts all internal self-tests and places a response into the output queue indicating whether or not the device completed the self-tests without any detected errors. It returns a 0 for success; a 1 if a failure was detected.

Upon successful completion of *TST?, the device settings are restored to their values prior to the *TST?
**WAI**

**Syntax**

*WAI*

**Description**

The *WAI commands allows no further execution of commands or queries until the No Operation Pending flag is true, or receipt of a Device Clear (dcas) message, or a power on.

The *WAI command can be used for overlapped commands. It stops the program execution until any pending overlapped commands have finished. Specifically, it waits until the No Operation Pending flag is TRUE, or receipt of a dcas message, or a power on.
SCPI Standard Commands

The following SCPI standard commands are implemented:

**Table 23  SCPI standard commands**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SYSTem:ERRor[:NEXT]?</td>
<td>:SYSTem:ERRor[:NEXT]? on page 108</td>
</tr>
</tbody>
</table>

**:SYSTem:ERRor[:NEXT]?**

**Syntax**: :SYSTem:ERRor[:NEXT]?

**Description**: Queries and at the same time deletes the oldest entry in the error queue.

**:SYSTem:HELP:HEADers?**

**Syntax**: :SYSTem:HELP:HEADers?

**Description**: This query returns all SCPI commands, queries and IEEE 488.2 common commands and queries currently implemented in the instrument.

**:SYSTem:VERSion?**

**Syntax**: :SYSTem:VERSion?

**Description**: This query returns a numeric value corresponding to the SCPI version number for which the instrument complies. The format is YYYY.V where Ys correspond to the year and the V corresponds to an approved version number for that year.
Miscellaneous Commands

The following miscellaneous commands are implemented:

Table 24  Miscellaneous commands

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
</table>

:TEST:RESults?

Syntax   :TEST:RESults?

.Description  This command returns the results of self test.
Command Syntax to Find Min/Max Values

The following example illustrates the syntax to find min/max values for any system parameter:

In the below example, “Voltage Amplitude” is the parameter for which min/max values are to be found.

:SOURce:VOLTage:AMPLitude?

Syntax for Min/Max Value

Example for Min Value

Example for Max Value

:SOURce:VOLTage:AMPLitude? 'identifier', <MIN/MAX>


Location and Module Mapping - Reference

The following table lists locations and their respective modules.

<table>
<thead>
<tr>
<th>Location</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Out</td>
<td>M8041A, M8051A, M8061A, M8062A, M8195A and M8045A</td>
</tr>
<tr>
<td>Data In</td>
<td>M8041A, M8051A, M8061A, M8062A and M8046A</td>
</tr>
<tr>
<td>System</td>
<td>M8041A</td>
</tr>
<tr>
<td>Clk Gen</td>
<td>M8041A, M8062A, M8195A and M8045A</td>
</tr>
<tr>
<td>Ref Clk Out</td>
<td>M8041A and M8045A</td>
</tr>
<tr>
<td>Clk Out</td>
<td>M8041A and M8045A</td>
</tr>
<tr>
<td>Cln Clk Out</td>
<td>M8062A</td>
</tr>
<tr>
<td>Trig Out</td>
<td>M8041A and M8045A</td>
</tr>
<tr>
<td>Sys Out A</td>
<td>M8041A and M8045A</td>
</tr>
<tr>
<td>Sys Out B</td>
<td>M8041A and M8045A</td>
</tr>
<tr>
<td>Ctrl Out A</td>
<td>M8041A, M8051A, M8045A and M8046A</td>
</tr>
<tr>
<td>Sys In A</td>
<td>M8041A and M8045A</td>
</tr>
<tr>
<td>Sys In B</td>
<td>M8041A and M8045A</td>
</tr>
<tr>
<td>Ctrl In A</td>
<td>M8041A, M8051A, M8045A and M8046A</td>
</tr>
<tr>
<td>Ctrl In B</td>
<td>M8041A and M8051A</td>
</tr>
<tr>
<td>Elect Idle In</td>
<td>M8061A and M8062A</td>
</tr>
<tr>
<td>Clk In</td>
<td>M8046A</td>
</tr>
</tbody>
</table>

**NOTE**
The “System” location is only available for few SCPIs. It is not visible on the M8070A user interface.
STATus Subsystem

This subsystem has the following SCPI structure:

```
:STATus
    :QUESTionable
    ...
    :OPERation
    ...
    :INSTRument
    ...
```

This subsystem has the following commands and subnodes:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:QUESTionable</td>
<td>:STATus:QUESTionable Subnode on page 113</td>
</tr>
<tr>
<td>:OPERation</td>
<td>:STATus:OPERation Subnode on page 122</td>
</tr>
<tr>
<td>:INSTRument</td>
<td>:STATus:INSTRument Subnode on page 126</td>
</tr>
</tbody>
</table>
This subnode has the following SCPI structure:

```
:STATus:QUEStionable
  :CONDition?                      
  ::STATus:QUEStionable:CONDition? on page 115
  :ENABle[?]                      
  ::STATus:QUEStionable:ENABle[?] on page 115
  [:EVENT]?                       
  ::STATus:QUEStionable[:EVENT]? on page 115
  :NTRansition[?]                
  :PTRansition[?]                
  :INSTrument[?]                 
  :CONDition?                     
  ::STATus:QUEStionable:CONDition? on page 115
  :ENABle[?]                      
  ::STATus:QUEStionable:ENABle[?] on page 115
  [:EVENT]?                       
  ::STATus:QUEStionable[:EVENT]? on page 115
  :NTRansition[?]                
  :PTRansition[?]                
  :PROTection                     
  :CONDition?                     
  ::STATus:QUEStionable:CONDition? on page 115
  :ENABle[?]                      
  ::STATus:QUEStionable:ENABle[?] on page 115
  [:EVENT]?                       
  ::STATus:QUEStionable[:EVENT]? on page 115
  :NTRansition[?]                
  :PTRansition[?]                
  :SYMBo!                          
  :CONDition?                     
  ::STATus:QUEStionable:CONDition? on page 115
  :ENABle[?]                      
  ::STATus:QUEStionable:ENABle[?] on page 115
  [:EVENT]?                       
  ::STATus:QUEStionable[:EVENT]? on page 115
  :NTRansition[?]                
  :PTRansition[?]                
```

This subnode has the following commands:

Table 27

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:ENABle[?]</td>
<td>:STATus:QUEStionable:ENABle[?] on page 115</td>
</tr>
<tr>
<td>[:EVENT]?</td>
<td>:STATus:QUEStionable[:EVENT]? on page 115</td>
</tr>
<tr>
<td>Name</td>
<td>Description under</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>:NTRansition[?]</td>
<td>:STATus:QUESTionable:NTRansition[?] on page 116</td>
</tr>
<tr>
<td>:PTRansition[?]</td>
<td>:STATus:QUESTionable:PTRansition[?] on page 116</td>
</tr>
<tr>
<td>:INSTrument[?]</td>
<td>:STATus:QUESTionable:INSTrument on page 116</td>
</tr>
<tr>
<td>:INSTrument[:EVENT]?</td>
<td>:STATus:QUESTionable:INSTrument[:EVENT]? on page 117</td>
</tr>
<tr>
<td>:PROTection</td>
<td>:STATus:QUESTionable:PROTection on page 118</td>
</tr>
<tr>
<td>:PROTection:ENABLE[?]</td>
<td>:STATus:QUESTionable:PROTection:ENABLE[?] on page 119</td>
</tr>
<tr>
<td>:PROTection[:EVENT]?</td>
<td>:STATus:QUESTionable:PROTection[:EVENT]? on page 119</td>
</tr>
<tr>
<td>:PROTection:NTRansition[?]</td>
<td>:STATus:QUESTionable:PROTection:NTRansition[?] on page 119</td>
</tr>
<tr>
<td>:PROTection:PTRansition[?]</td>
<td>:STATus:QUESTionable:PROTection:PTRansition[?] on page 119</td>
</tr>
<tr>
<td>:SYMBol</td>
<td>:STATus:QUESTionable:SYMBol on page 120</td>
</tr>
</tbody>
</table>
### :STATus:QUEStionable:CONDition?

**Syntax**
:STATus:QUEStionable:CONDition?

**Description**
This command query returns the contents of the condition register of the Questionable Status Register structure.

### :STATus:QUEStionable:ENABLE[?]

**Syntax**
:STATus:QUEStionable:ENABLE <NRf>

:STATus:QUEStionable:ENABLE?

**Input Parameters**
- `<NRf>`: Set enable mask.

**Description**
This command sets the enable mask in the Questionable Status Register structure, which allows true conditions in the event register to be reported in the summary bit. The query returns the weighted value of the bits that are set in the enable register.

### :STATus:QUEStionable:[EVENT]?

**Syntax**
:STATus:QUEStionable:EVENT?

**Description**
This command query returns the contents of the Questionable Status event register.
:STATus:QUESTIONable:NTRansition?

Syntax

:STATus:QUESTIONable:NTRansition <NRf>

:STATus:QUESTIONable:NTRansition?

Input Parameters

<NRf>: Set transition filter state.

Description

This command sets the transition filter state in the Questionable Status Register structure. The query returns the weighted value of the bits that are set to pass negative transitions in the transition filter.

:STATus:QUESTIONable:PTRansition?

Syntax

:STATus:QUESTIONable:PTRansition <NRf>

:STATus:QUESTIONable:PTRansition?

Input Parameters

<NRf>: Set transition filter state.

Description

This command sets the transition filter state in the Questionable Status Register structure. This is the default setting of the instrument. The query returns the weighted value of the bits that are set to pass positive transitions in the transition filter.

:STATus:QUESTIONable:INSTrument

The summary bit of this register set is reflected in bit 9 of the Questionable status register.

The :STATus:QUESTIONable:INSTrument contains conditions which reflect states of the instrument’s normal operation. These bits are summary bits. For example, if a Data Loss is detected by one or more error detectors, the Data Loss condition will be set.

The definition of each of these bits (condition register) is as follows:

**Bit 0 – Error**

One or more error detectors detect an error.

**Bit 1 – Data Loss**
This bit is set when the data source is turned off, not connected, or the cables or device is faulty. This bit can also be set when the 0/1 threshold is not within the eye limits of the incoming data signal. In this last case, use Auto Align or select Avg 0/1 Threshold.

**Bit 2 – Sync Loss**

This bit is set when the error detector pattern does not match the incoming data pattern or the BER of your device is higher than the sync threshold.

**Bit 3 – Clock Loss**

This bit is set when the pattern generator receives no external clock signal or the error detector receives no clock input signal.

**Bit 4 – CDR Unlocked**

One or more CDRs of the instrument are unlocked.

**:STATus:QUEStionable:INSTrument:CONDition?**

**Syntax**: :STATus:QUEStionable:INSTrument:CONDition?

**Description**: This command returns the contents of the condition register of the Questionable Status Register (bit 9), which reflects the state of the instrument.

**:STATus:QUEStionable:INSTrument:ENABle[?]**

**Syntax**: :STATus:QUEStionable:INSTrument:ENABle <NRf>  
:STATus:QUEStionable:INSTrument:ENABle?

**Description**: This command sets the bits in the event enable register that can generate a summary bit used for the instrument state. The query returns the weighted value of the bits that are set in the event enable register.

**:STATus:QUEStionable:INSTrument[:EVENt]?**

**Syntax**: :STATus:QUEStionable:INSTrument[:EVENt]?

**Description**: This command queries the contents of the Questionable Status event register.
:STATus:QUEStionable:INSTrument:NTRansition[?]

Syntax :STATus:QUEStionable:INSTrument:NTRansition[?] <NRf>

Input <NRf>: Set transition filter state.

Parameters

Description This command sets the transition filter state in the Questionable Status Register structure. The query returns the weighted value of the bits that are set to pass negative transitions in the transition filter.

:STATus:QUEStionable:INSTrument:PTRansition[?]

Syntax :STATus:QUEStionable:INSTrument:PTRansition[?] <NRf>

Input <NRf>: Set transition filter state.

Parameters

Description This command sets the transition filter state in the Questionable Status Register structure. The query returns the weighted value of the bits that are set to pass positive transitions in the transition filter.

:STATus:QUEStionable:PROTection

The summary bit of this register set is reflected in bit 11 of the Questionable Status Register.

The :STATus:QUEStionable:PROTection contains conditions which reflect states of the instrument’s normal operation. These are summary bits. For example, if one or more outputs of the instrument detect an overload condition, the OverloadDetection bit will be set.

The definition of each of these bits (condition register) is as follows:

**Bit 0 - OverloadDetection**
Indicates that an overload condition has been detected.

:STATus:QUEStionable:PROTection:CONDition?

Syntax :STATus:QUEStionable:PROTection:CONDition?

Description This command returns the contents of the condition register of the Questionable Status Register (bit 11) to detect an overload condition.
`:STATus:QUEStionable:PROTection:ENABle[?]`

**Syntax**

`:STATus:QUEStionable:PROTection:ENABle <NRf>`

`:STATus:QUEStionable:PROTection:ENABle?`

**Description**

This command sets the bits in the event enable register that can generate a summary bit used for monitoring a symbol alignment loss. The query returns the weighted value of the bits that are set in the event enable register.

`:STATus:QUEStionable:PROTection[:EVENt]?`

**Syntax**

`:STATus:QUEStionable:PROTection[:EVENt]?`

**Description**

This command queries the contents of the Questionable Status event register.

`:STATus:QUEStionable:PROTection:NTRansition[?]`

**Syntax**

`:STATus:QUEStionable:PROTection:NTRansition <NRf>`

`:STATus:QUEStionable:PROTection:NTRansition?`

**Input Parameters**

- `<NRf>`: Set transition filter state.

**Description**

This command sets the transition filter state in the Questionable Status Register structure. The query returns the weighted value of the bits that are set to pass negative transitions in the transition filter.

`:STATus:QUEStionable:PROTection:PTRansition[?]`

**Syntax**

`:STATus:QUEStionable:PROTection:PTRansition <NRf>`

`:STATus:QUEStionable:PROTection:PTRansition?`

**Input Parameters**

- `<NRf>`: Set transition filter state.

**Description**

This command sets the transition filter state in the Questionable Status Register structure. The query returns the weighted value of the bits that are set to pass positive transitions in the transition filter.
The summary bit of this register set is reflected in bit 12 of the Questionable Status Register.

The :STATus:QUEStionable:SYMBol contains conditions which reflect states of the instrument’s normal operation. These are summary bits. For example, if an error detector(s) of the instrument detects a symbol alignment loss, the SymbolAlignmentLoss bit will be set.

The definition of each of these bits (condition register) is as follows:

**Bit 0 - SymbolAlignmentLoss**
Indicates whether the error detector(s) has experienced a symbol alignment loss.

**:STATus:QUEStionable:SYMBol:CONDition?**
**Syntax** :STATus:QUEStionable:SYMBol:CONDition?
**Description** This query returns the contents of the condition register of the Questionable Status Register (bit 12) to detect a symbol alignment loss.

**:STATus:QUEStionable:SYMBol:ENABle[?]**
**Syntax** :STATus:QUEStionable:SYMBol:ENABle <NRf>
:STATus:QUEStionable:SYMBol:ENABle?
**Description** This command sets the bits in the event enable register that can generate a summary bit used for monitoring a symbol alignment loss. The query returns the weighted value of the bits that are set in the event enable register.

**:STATus:QUEStionable:SYMBol[:EVENt]?**
**Syntax** :STATus:QUEStionable:SYMBol[:EVENt]?
**Description** This command queries the contents of the Questionable Status event register.
**:STATus:QUESTionable:SYMBol:NTRansition?**

**Syntax**

**:STATus:QUESTionable:SYMBol:NTRansition?** <NRf>

**Input Parameters**

<NRf>: Set transition filter state.

**Description**

This command sets the transition filter state in the Questionable Status Register structure. The query returns the weighted value of the bits that are set to pass negative transitions in the transition filter.

**:STATus:QUESTionable:SYMBol:PTRansition?**

**Syntax**

**:STATus:QUESTionable:SYMBol:PTRansition** <NRf>

**:STATus:QUESTionable:SYMBol:PTRansition?**

**Input Parameters**

<NRf>: Set transition filter state.

**Description**

This command sets the transition filter state in the Questionable Status Register structure. The query returns the weighted value of the bits that are set to pass positive transitions in the transition filter.
:STATus:OPERation Subnode

This subnode has the following SCPI structure:

```
:STATus
  :OPERation
    :CONDition?
    :ENABle[?]
    [:EVEN]?
    :NTRansition[?]
    :PTRansition[?]
    :RUN
      :CONDition?
      :ENABle[?]
      [:EVEN]?
      :NTRansition[?]
      :PTRansition[?]
```

### Table 28

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:ENABle[?]</td>
<td>:STATus:OPERation:ENABle[?] on page 123</td>
</tr>
<tr>
<td>[:EVEN]?</td>
<td>:STATus:OPERation[:EVEN]? on page 123</td>
</tr>
<tr>
<td>:NTRansition[?]</td>
<td>:STATus:OPERation:NTRansition[?] on page 123</td>
</tr>
<tr>
<td>:PTRansition[?]</td>
<td>:STATus:OPERation:PTRansition[?] on page 124</td>
</tr>
<tr>
<td>:RUN</td>
<td>:STATus:OPERation:RUN on page 124</td>
</tr>
<tr>
<td>:RUN[:EVEN]?</td>
<td>:STATus:OPERation:RUN[:EVEN]? on page 125</td>
</tr>
<tr>
<td>:RUN:NTRansition[?]</td>
<td>:STATus:OPERation:RUN:NTRansition[?] on page 125</td>
</tr>
</tbody>
</table>
**:STATus:OPERation:CONDition?**

**Syntax**: :STATus:OPERation:CONDition?

**Description**: This query returns the contents of the condition register of the Operation Status Register structure.

**:STATus:OPERation:ENABle[?]**

**Syntax**: :STATus:OPERation:ENABle <NRf>  
:STATus:OPERation:ENABle?

**Input Parameters**:  
<NRf>: Set enable mask.

**Description**: This command sets the enable mask in the Operation Status Register structure, which allows true conditions in the event register to be reported in the summary bit. The query returns the weighted value of the bits that are set in the enable register.

**:STATus:OPERation[:EVENt]?**

**Syntax**: :STATus:OPERation[:EVENt]?

**Description**: This query returns the content of Operation Status Event Register.

**:STATus:OPERation:NTRansition[?]**

**Syntax**: :STATus:OPERation:NTRansition <NRf>  
:STATus:OPERation:NTRansition?

**Input Parameters**:  
<NRf>: Set transition filter state.

**Description**: This command sets the transition filter state in the Operation Status Register structure. The query returns the weighted value of the bits that are set to pass negative transitions in the transition filter.
**:STATus:OPERation:PTRansition[?]**

**Syntax**

:STATus:OPERation:PTRansition <NRf>

:STATus:OPERation:PTRansition?

**Input Parameters**

<NRf>: Set transition filter state.

**Description**

This command sets the transition filter state in the Operation Status Register structure. This is the default setting of the instrument. The query returns the weighted value of the bits that are set to pass positive transitions in the transition filter.

**:STATus:OPERation:RUN**

The summary bit of this register set is reflected in bit 8 of the Operation Status Register.

The :STATus:OPERation:RUN contains conditions which reflect states of the instrument’s normal operation. These are summary bits. For example, if the pattern generators, error detectors and clock generator are functioning normally, the Running bit will be set.

The definition of each of these bits (condition register) is as follows:

**Bit 0 - Running**

Indicates that the instrument is in normal operation mode; generators are working, all error detectors are running and the clock generator is functioning properly.

**:STATus:OPERation:RUN:CONDition?**

**Syntax**

:STATus:OPERation:RUN:CONDition?

**Description**

This command returns the contents of the condition register of the Operation Status Register (bit 8) to detect if all components are operating normally.
**:STATus:OPERation:RUN:ENABle**

**Syntax**

```plaintext
:STATus:OPERation:RUN:ENABle <NRf>
:STATus:OPERation:RUN:ENABle?
```

**Description**

This command sets the bits in the event enable register that can generate a summary bit used to determine if the hardware is functioning properly. The query returns the weighted value of the bits that are set in the event enable register.

**:STATus:OPERation:RUN[:EVENt]**

**Syntax**

```plaintext
:STATus:OPERation:RUN[:EVENt]?
```

**Description**

This query returns the contents of the Operation Status event register.

**:STATus:OPERation:RUN:NTRansition**

**Syntax**

```plaintext
:STATus:OPERation:RUN:NTRansition <NRf>
:STATus:OPERation:RUN:NTRansition?
```

**Input Parameters**

- `<NRf>`: Set transition filter state.

**Description**

This command sets the transition filter state in the Operation Status Register structure. The query returns the weighted value of the bits that are set to pass negative transitions in the transition filter.

**:STATus:OPERation:RUN:PTRansition**

**Syntax**

```plaintext
:STATus:OPERation:RUN:PTRansition <NRf>
:STATus:OPERation:RUN:PTRansition?
```

**Input Parameters**

- `<NRf>`: Set transition filter state.

**Description**

This command sets the transition filter state in the Operation Status Register structure. The query returns the weighted value of the bits that are set to pass positive transitions in the transition filter.
:STATus:INStrument Subnode

The :STATus:INStrument commands return the 'current' value/condition of several predefined device states. These commands are a set of SCPI queries which return true (1) or false (0).

This subnode has the following SCPI structure:

```
:STATus
  :INStrument
    :CLOSe?
    :DLOSs?
    :SLOSS?
    :SALoss?
    :ERRor?
    :OVERload?
    :RCINput?
    :RUN?
    :WAIT?
    :CDR
    :ULOCKed?
```

This subnode has the following commands:

**Table 29**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SLOSS?</td>
<td>:STATus:INStrument:SLOSS? on page 128</td>
</tr>
<tr>
<td>:ERRor?</td>
<td>:STATus:INStrument:ERRor? on page 128</td>
</tr>
</tbody>
</table>
### :STATus:INStrument:CLOSs?

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
</table>

- **Syntax**: :STATus:INStrument:CLOSs? 'identifier'
- **Input Parameters**: 'identifier': 'M*.DataIn1' or 'M*.DataIn2' or 'M*.DataOut1' or 'M*.DataOut2' or 'M1.System'
- **Return Range**: 0|1
- **Description**: This query indicates if the clock has been lost. A true (1) indicates a clock loss.
- **Example**: :STAT:INST:CLOS? 'M1.DataIn1'

### :STATus:INStrument:DLOSs?

- **Syntax**: :STATus:INStrument:DLOSs? 'identifier'
- **Input Parameters**: 'identifier': 'M*.DataIn1' or 'M*.DataIn2' or 'M*.DataOut1' or 'M*.DataOut2' or 'M1.System'
- **Return Range**: 0|1
- **Description**: This query indicates if the data has been lost. A true (1) indicates data loss.
- **Example**: :STAT:INST:DLOS? 'M1.DataIn1'
5  SCPI Command Reference

**:STATus:INStrument:SLOSs?**

Syntax: 

**:STATus:INStrument:SLOSs? 'identifier'**

Input Parameters

| 'identifier': 'M*.DataIn1' or 'M*.DataIn2' or 'M1.System' |

Return Range

0|1

Description

This query indicates if the sync has been lost. A true (1) indicates a sync loss.

Example: 

**:STAT:INST:SLOS? 'M1.DataIn1'**

**:STATus:INStrument:SALoss?**

Syntax: 

**:STATus:INStrument:SALoss? 'identifier'**

Input Parameters

| 'identifier': 'M*.DataIn1' or 'M*.DataIn2' or 'M1.System' |

Return Range

0|1

Description

This query indicates if the symbol alignment has been lost. A true (1) indicates a symbol alignment loss.

Example: 

**:STAT:INST:SAL? 'M1.DataIn1'**

**:STATus:INStrument:ERRor?**

Syntax: 

**:STATus:INStrument:ERRor? 'identifier'**

Input Parameters

| 'identifier': 'M*.DataIn1' or 'M*.DataIn2' or 'M1.System' |

Return Range

0|1

Description

This query indicates if an error condition has occurred. A true (1) indicates an error condition.

Example: 

**:STAT:INST:ERR? 'M1.DataIn1'**
:STATus:INStrument:OVERload?

Syntax: :STATus:INStrument:OVERload? ‘identifier’

Parameters:
- ‘identifier’: ‘M*.DataIn1’ or ‘M*.DataIn2’ or ‘M*.DataOut1’ or ‘M*.DataOut2’
- or ‘M1.System’

Return Range: 0|1

Description: This query indicates an overload condition exists at an input or output. A true (1) indicates an overload condition.


:STATus:INStrument:R Cinput?

Syntax: :STATus:INStrument:RCINput? ‘identifier’

Parameters:
- ‘identifier’: ‘M1.ClkGen’

Return Range: 0|1

Description: This query indicates the input state of the reference clock.


:STATus:INStrument:RUN?

Syntax: :STATus:INStrument:RUN? ‘identifier’

Parameters:
- ‘identifier’: ‘M*.DataIn1’ or ‘M*.DataIn2’ or ‘M1.System’

Return Range: 0|1

Description: This query indicates if the addressed channel is in running state. A true (1) indicates the input is in running state.

:STAT:INSTR:RUN:WAIT?


Input Parameters

'identifier': 'M*.DataOut1' or 'M*.DataOut2'

Description

This query waits until the addressed channel is in running state. Using this query is recommended before starting any measurement that relies on the operation of the addressed channel.

The optional timeout argument is having a default value of 90 and the accepted value range is 2 to 300. The timeout is specified in terms of seconds.

Example

Change the system frequency and wait until the pattern generator channels are running again


:STAT:INSTR:CDR:ULOCked?


Input Parameters

'identifier': 'M*.DataIn1' or 'M*.DataIn2' or 'M1.System'

Return Range: 0|1

Description

This query indicates the lock state of the clock data recovery. A true (1) indicates an unlocked state.

Example

:STAT:INSTR:CDR:UNLOC? 'M1.DataIn1'

**NOTE**

- The VISA timeout must be set to at least the timeout value that is used for :STAT:INSTR:RUN:WAIT?
- *OPC? does not wait until the individual channels are running, since this cannot be ensured for channels that are clocked by externally provided signals, or for error detector channels that operate on a recovered clock signal.
:STATus:INStrument:ALIGnment Subnode

This subsystem is reporting alignment specific status information, and offers an alternative way of retrieving the same status information that is available via the :INPut:ALIngent:STATus Subnode on page 268.

This subsystem has the following SCPI structure:

```
:STATus
  :INStrument
    :ALIGNment[:VALue]?
```

<table>
<thead>
<tr>
<th>Table 30</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>:STATus:INStrument:ALIGNment[:VALue]?</td>
</tr>
</tbody>
</table>

:STATus:INStrument:ALIGNment[:VALue]?

**Syntax**: :STATus:INStrument:ALIGNment[:VALue]? 'identifier',

**Input**:
- 'identifier': Any error detector location and group name. E.g. 'M1.DataIn1', 'M2.DataIn', 'INF1.DataIn1'

**Parameters**: 'identifier', 'identifier'

**Description**: This command retrieves alignment status information for an error detector or error detector group.

This command can be used to retrieve information about the alignment procedure that is working on the addressed error detector channel, or a group of error detectors. This command is reporting the status independently of the executing alignment procedure. There is no differentiation in respect to different types of alignments.

This command returns the following values:

- INPROGRESS – At least one addressed error detector is executing an alignment procedure.
- ABORTED – The alignment procedure has been aborted by the user before all aligning error detector channels finished the procedure.
• FAILED - All addressed error detector channels have finished the alignment procedure, but at least one was unable to finish successfully.
• SUCCESS - The addressed error detector channels have successfully finished the alignment procedure. Error detector channels that are addressed by the command but return UNKNOWN state, do not contribute to the result.
• UNKNOWN - None of the addressed error detectors has executed an alignment procedure yet.

**NOTE**
This command cannot be used in the same compound SCPI statement that is starting the alignment procedure. If this command is sent to the instrument right after the command that is initiating the alignment procedure, then the returned value is undefined and can reflect the state that was valid before the alignment. This can be overcome by reading any value from the instrument (e.g. *ESE?), except for *OPC? as this will block until the alignment procedure is finished.

Example: `:INP:INST:ALIGN? 'M1.DataIn1'`
TRIGger Subsystem

The TRIGger subsystem selects the trigger mode at the TRIG OUT port. This subsystem has the following SCPI structure:

![SCPI Command Structure Diagram]

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:SOURce][?]</td>
<td>:TRIGger[:SOURce][?] on page 134</td>
</tr>
<tr>
<td>[:SOURce]:FREQuency?</td>
<td>:TRIGger[:SOURce]:FREQuency? on page 135</td>
</tr>
<tr>
<td>[:SOURce]:FREQuency?</td>
<td>:TRIGger:INTernal[:SOURce][?] on page 135</td>
</tr>
<tr>
<td>[:FREQuency][?]</td>
<td>:TRIGger:REFeReference[:FREQuency][?] on page 136</td>
</tr>
<tr>
<td>[:FREQuency]:DETect</td>
<td>:TRIGger:DIRect[:FREQuency][?] on page 136</td>
</tr>
<tr>
<td>[:FREQuency]:DETect</td>
<td>:TRIGger:DIRect[:FREQuency][?] on page 136</td>
</tr>
<tr>
<td>[:MULTiplier][?]</td>
<td>:TRIGger:CMULtiplier[:LBWidth][?] on page 137</td>
</tr>
</tbody>
</table>
The trigger source modes include the following:

- **INTernal** 100 MHz reference clock or AXIe 100 MHz (M9505A AXIe Chassis)
- **REFerence** 10 MHz / 100 MHz external reference clock
- **DIRect** 8.1 GHz to 16.207 GHz clock used as the system frequency directly in the M8041A
- **CMULTiplier** Multiplied/divided clock used to specify system frequency

This SCPI is applicable for M8041A, M8062A, M8045A, DSO634A, DSA634A, DSA96204A and DSOX96204A.
**:TRIGger[:SOURce]:FREQuency?**

**Syntax**: :TRIGger[:SOURce]:FREQuency? 'identifier'

**Input Parameters**

- 'identifier': 'M1.TrigOut'

**Description**

This query returns the effective frequency of the trigger out clock signal. This parameter is only active if the subrate clock operating mode is selected. The mentioned frequency value depends upon the system frequency & divider value.

This SCPI is applicable for M8041A.

**Example**

:TRIG:FREQ? 'M1.TrigOut'

---

**:TRIGger:INTernal[:SOURce][:?]**

**Syntax**: :TRIGger:INTernal[:SOURce] 'identifier', <AXIFrame|INTernal>

- :TRIGger:INTernal[:SOURce]? 'identifier'

**Input Parameters**

- 'identifier': M1.ClkGen

**Return Range**

AXIF|INT

**Description**

In the INternal mode different reference clock sources are available for the internal oscillator. Select between the 100 MHz reference clock of the M9505A AXIe Chassis or the 100 MHz reference clock of the M8041A module.

This SCPI is applicable for M8041A and M8045A.

**Example**


:TRIGger:REference[:FREQuency]?

Syntax: :TRIGger:REference[:FREQuency] 'identifier', <REF10|REF100>
        :TRIGger:REference[:FREQuency]? 'identifier'

Input Parameters:
'identifier': 'M1.ClkGen', 'INF*.Common'

Return Range: REF10|REF100

Description: In the REReference mode two different expected external reference clocks can be specified: 10 MHz / 100 MHz.
This SCPI is applicable for M8041A, M8045A and DSO634A, DSA634A, DSAX6204A and DSOX6204A.


:TRIGger:DIRect[:FREQuency]?

Syntax: :TRIGger:DIRect[:FREQuency] 'identifier', <NRf>
        :TRIGger:DIRect[:FREQuency]? 'identifier'

Input Parameters:
'identifier': 'M1.ClkGen'

Return Range: 8.1 GHz to 16.207 GHz

Description: In the DIRect mode the frequency value expected at the REF CLK IN port can be set manually. Acceptable units include MHz, GHz and exponents (for example, 10E9 is the same as 10 GHz).
This SCPI is applicable for M8041A and M8045A.

:TRIGger:DIRect[:FREQuency]:DETect

Syntax: :TRIGger:DIRect[:FREQuency]:DETect 'identifier'

Parameters:
- 'identifier': 'M1.ClkGen'

Description: The externally provided clock on the REF CLK IN port is measured once and used as the new system frequency (data rate), as required. This SCPI is applicable for M8041A and M8045A.

Example: :TRIG:DIR:DET 'M1.ClkGen'

:TRIGger:CMULtiplier:LBWidth[?]

Syntax: :TRIGger:CMULtiplier:LBWidth 'identifier', <BW100|BW2|BW5>

:TRIGger:CMULtiplier:LBWidth? 'identifier'

Parameters:
- 'identifier': 'M1.ClkGen'
- <BW100|BW2|BW5>: Specify loop bandwidth.

Return Range: BW100|BW2|BW5

Description: Different loop bandwidths can be selected. Modulated signals (for example SSC) at the REF CLK IN port can be provided depending on the loop bandwidth. This SCPI is applicable for M8041A.

Example: :TRIG:CMUL:LBW 'M1.ClkGen', BW2

:TRIGger:CMULtiplier:FREQuency:MULTiplier[?]

Syntax: :TRIGger:CMULtiplier:FREQuency:MULTiplier 'identifier', <NRf>

:TRIGger:CMULtiplier:FREQuency:MULTiplier? 'identifier'

Parameters:
- 'identifier': 'M1.ClkGen'
- <NRf>: Specify the multiplier value.

Return Range: 2 to 1620; the range is dependent on other system parameters. See Command Syntax to Find Min/Max Values on page 110.
Description: This command specifies the external reference clock frequency multiplier. This SCPI is applicable for M8041A and M8045A.


:TRIGger:CMULtiplier:FREQuency:DIVider[?]

Syntax: :TRIGger:CMULtiplier:FREQuency:DIVider 'identifier', <NRf>
:TRIGger:CMULtiplier:FREQuency:DIVider? 'identifier'

Input Parameters:
- 'identifier': 'M1.ClkGen'
- <NRf>: Specify divider value.

Return Range: 1 to 63; Range is dependent on other system parameters.

Description: This command specifies the external reference clock frequency divider. This SCPI is applicable for M8041A and M8045A.

Example: :TRIG:CMUL:FREQ:DIV 'M1.ClkGen', 1

:TRIGger:CMULtiplier:FREQuency[?]

Syntax: :TRIGger:CMULtiplier:FREQuency 'identifier', <NRf>
:TRIGger:CMULtiplier:FREQuency? 'identifier'

Input Parameters:
- 'identifier': 'M1.ClkGen'
- <NRf>: Specify reference frequency value.

Return Range: 10 MHz to 202.5875 MHz; the range is dependent on other system parameters. See Command Syntax to Find Min/Max Values on page 110.

Description: This command is used to set the expected frequency value of the external provided clock at REF CLK IN manually. The sent value will be multiplied / divided for calculating the system frequency (data rate). Acceptable units include kHz, MHz and exponents (for example, 200E6 is the same as 200 MHz).

This SCPI is applicable for M8041A and M8045A.

:TRIG:CMUL:FREQ? 'M1.ClkGen'
**.:TRIGger:CMULtiplier:FREQuency:DETect**

**Syntax**

`:TRIGger:CMULtiplier:FREQuency:DETect 'identifier'`

**Input**

'identifier': 'M1.ClkGen'

**Parameters**

**Description**

The external provided clock at the REF CLK IN port is measured once and will be multiplied/divided for calculating the system frequency (data rate).

This SCPI is applicable for M8041A and M8045A.

**Example**

`:TRIG:CMUL:FREQ:DET 'M1.ClkGen'`
Source Subsystem

The SOURce subsystem controls output signals (for example, for setting frequency and levels).

This subsystem has the following SCPI structure:

```
[:SOURce]
  - :FREQuency
  - :PERiod[?]
  - :DRATe
  - :MEMoryCONFigure
    - [:VALue][?]
    - :COUPling[?]
  - :PULSe
    - ...
  - :VOLTage
    - ...
  - :JITTer
    - ...
    - HFRequency
      - :PERiodic(*)[?]
      - ...
      - :BUNCorrelate
      - ...
      - :RANDom
      - ...
      - :SPECtrally
      - ...
      - LFRequency
        - :PERiodic
        - ...
        - RSSClocking
        - ...
        - :SWEep
        - ...
      - :SSCLocking
        - ...
      - :INTerference
        - ...
```
This subsystem has the following commands and subnodes:

Table 32

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:FREQuency?]</td>
<td>[:SOURce]:FREQuency?] on page 141</td>
</tr>
<tr>
<td>[:PERiod?]</td>
<td>[:SOURce]:PERiod?] on page 142</td>
</tr>
<tr>
<td>[:DRate?]</td>
<td>[:SOURce]:DRate?] on page 142</td>
</tr>
<tr>
<td>[:MEMory:CONFigure:VALue?]</td>
<td>[:SOURce]:MEMory:CONFigure:VALue?] on page 143</td>
</tr>
<tr>
<td>[:MEMory:CONFigure:COUPling?]</td>
<td>[:SOURce]:MEMory:CONFigure:COUPling?] on page 143</td>
</tr>
</tbody>
</table>

Subnodes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:PULse]</td>
<td>[:SOURce]:PULSe Subnode on page 144</td>
</tr>
<tr>
<td>[:VOLTage]</td>
<td>[:SOURce]:VOLTage Subnode on page 149</td>
</tr>
<tr>
<td>[:JITTer]</td>
<td>[:SOURce]:JITTer Subnode on page 154</td>
</tr>
<tr>
<td>[:JITTer:HFRequency]</td>
<td>[:SOURce]:JITTer:HFRequency Subnode on page 157</td>
</tr>
<tr>
<td>[:JITTer:LFRequency]</td>
<td>[:SOURce]:JITTer:LFRequency Subnode on page 174</td>
</tr>
<tr>
<td>[:JITTer:SWEep]</td>
<td>[:SOURce]:JITTer:SWEep Subnode on page 180</td>
</tr>
<tr>
<td>[:SSCLocking]</td>
<td>[:SOURce]:SSCLocking Subnode on page 187</td>
</tr>
<tr>
<td>[:INTerference]</td>
<td>[:SOURce]:INTerference Subnode on page 193</td>
</tr>
</tbody>
</table>

[:SOURce]:FREQuency[?]

Syntax  [:SOURce]:FREQuency 'identifier', <NRf>
[:SOURce]:FREQuency? 'identifier'

Input  'identifier': 'M1.ClkGen'
Parameters  <NRf>: Set system frequency.

Return Range  For M8041A: 256 MHz to 16.207 GHz
For M8195A: 256 MHz to 32.5 GHz
For M8045A: 2.025 GHz to 64.828 GHz

Description
Sets the frequency of the synthesizer in the M8020A/M8030A/M8040A. Acceptable units include kHz, MHz, GHz and exponents (for example, 10E9 is the same as 10 GHz). This SCPI is applicable for M8041A, M8195A, M8062A, M8196A and M8045A.

The SCPI is read only for M8062A.

Example
:FREQ 'M1.ClkGen', 10e9

[:SOURce]:PERiod[?]]

Syntax
[:SOURce]:PERiod 'identifier', <NRf>
[:SOURce]:PERiod? 'identifier'

Input
parameters

'identifier': 'M1.ClkGen'

<NRF>: Set system period.

Return Range
For M8041A: 61.702 ps to 3.906249 ns
For M8045A: 15.426 ps to 493.827

Description
Sets the M8020A/M8030A/M8040A system period. The period is the reciprocal value of the system frequency. This is provided as a convenience for those who prefer period instead of frequency. Acceptable units include ps, ns and exponents (for example, 61.702E-12 is the same as 61.702 ps).

This SCPI is applicable for M8041A, M8195A, M8196A and M8045A.

Example
:PER 'M1.ClkGen', 61.702ps

[:SOURce]:DRATe[?]

Syntax
[:SOURce]:DRATe 'identifier', <NRf>
[:SOURce]:DRATe? 'identifier'

Input
parameters

'identifier': 'M2.ClkGen'

<NRF>: Set data rate.

Return Range
512 Mb/s to 32.414 Gb/s

Description
This command/query sets/gets the data rate.
This SCPI is applicable for M8062A.

Example: :DRAT 'M2.ClkGen', 10e9

[SOURce]:MEMory:CONFigure[:VALue]?]
Syntax: [:SOURce]:MEMory:CONFigure 'Identifier', <SINGLE | DUAL | FOUR | RANDom | SINusoidal>
[:SOURce]:MEMory:CONFigure? 'Identifier'

Input Parameters:
'identifier': 'M1'
SINGLE | DUAL | FOUR | RANDom | SINusoidal

Case 1 - The following description is applicable when the four channel license has been purchased:
- SINGLE: DataOut1 will be sourced from extended memory and the other channels will be sourced from module internal memory. The data rate range in this mode is 256 Mb/s ... 32.5 Gb/s.
- DUAL: DataOut1 and DataOut2 will be sourced from extended memory and the other channels will be sourced from module internal memory. The data rate range in this mode is 256 Mb/s ... 16.25 Gb/s.
- FOUR: All four channels will be sourced from extended memory. The data rate range in this mode is 256 Mb/s ... 8.125 Gb/s.
- RANDom: All the available channels are used to generate random interference.
- SINusoidal: All the available channels are used to generate random interference.

[SOURce]:MEMory:CONFigure:COUPling?]
Syntax: [:SOURce]:MEMory:CONFigure:COUPling 'Identifier', NONe|ONE|FIVE|SIX
[:SOURce]:MEMory:CONFigure:COUPling? 'Identifier'

Input Parameters:
'identifier': 'M*.DataOut1' or 'M*.DataOut2' or 'M*.DataOut3' or 'M*.DataOut4'
NONe | ONE | FIVE | SIX

Description: This command sets the Channel Coupling for Channel One & Two and Channel Three & Four. In case of two channel license mode, Channel...
One and Four will be coupled.
This query returns the current setting.
This SCPI is applicable for M8195A and M8196A.

Example

:SOUR:MEM:CONF:COUP? 'M1'
ONET

[:SOURce]:PULSe Subnode
This subnode has the following SCPI structure:

```
[:SOURce]
  _PULSe
    _FILTER
      _TYPE
        :
          :ROFFfactor
      _DELAY
    _TRANSition
      :
        :FIXed
      [:LEADING]
```

This subnode has the following commands:

### Table 33

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:SOURce]:PULSe:FILTER:TYPE[?]</td>
<td>[:SOURce]:PULSe:FILTER:TYPE[?] on page 145</td>
</tr>
<tr>
<td>[:SOURce]:PULSe:ROFFfactor[?]</td>
<td>[:SOURce]:PULSe:ROFFfactor[?] on page 146</td>
</tr>
<tr>
<td>[:SOURce]:PULSe:DELAY[?]</td>
<td>[:SOURce]:PULSe:DELAY[?] on page 146</td>
</tr>
<tr>
<td>[:SOURce]:PULSe:TRANSition:FIXed[?]</td>
<td>[:SOURce]:PULSe:TRANSition:FIXed[?] on page 147</td>
</tr>
<tr>
<td>[:SOURce]:PULSe:TRANSition[:LEADING][?]</td>
<td>[:SOURce]:PULSe:TRANSition[:LEADING][?] on page 147</td>
</tr>
</tbody>
</table>

### [:SOURce]:PULSe:FILTER:TYPE[?]

**Syntax**

[:SOURce]:PULSe:FILTER:TYPE 'identifier', <GAUS | RCOS>

[:SOURce]:PULSe:FILTER:TYPE? 'identifier'

**Input Parameters**

'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', or 'M*.DataOut4'

**Return Range**

GAUS | RCOS
Description
This command specifies filter type used on the waveform. Depending upon the specific waveform used, the waveform gets properties like transition time or bandwidth limits.
This query returns the present setting.
This SCPI is applicable for M8196A.

Example

[:SOURce]:PULSe:ROFFfactor[?]

Syntax
[:SOURce]:PULSe:ROFFfactor 'identifier', <NRf>
[:SOURce]:PULSe:ROFFfactor? 'identifier'

Input Parameters
'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', or 'M*.DataOut4'
<NRF>: Sets the roll-off-factor value. The value should be between 0 to 1.

Description
This command sets the roll-off factor for the raised cosine pulse shaping filter. It is a measure for the excess bandwidth the waveform contains beyond Nyquist bandwidth.
This query returns the present setting.
This SCPI is applicable for M8196A.

Example
:PULS:ROFF 'M1.DataOut2', 0.50

[:SOURce]:PULSe:DElay[?]

Syntax
[:SOURce]:PULSe:DElay 'identifier', <NRf>
[:SOURce]:PULSe:DElay? 'identifier'

Input Parameters
'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', or 'M*.DataOut4'
<NRF>: Set the pulse delay time.

Return Range
For M8041A: 0.0 ps to 100 ns
For M8195A: -1ns to 1ns
For M8062A: 0 ps to 100 ns
For M8045A: 0 ps to 100 ns

Description
This command sets the time from the start of the period to the first edge of the pulse. Acceptable units include ps, ns and exponents (for example, 25E-9 is the same as 25 ns).
This SCPI is applicable for M8041A, M8051A, M8195A, M8062A and M8045A.

Example :PULS:DEL 'M1.DataOut2', 25ns

[:SOURce]:PULSe:TRANsition:FIXed?]

Syntax [:SOURce]:PULSe:TRANsition:FIXed 'identifier', <SMOoth | MODerate | STEep>

[:SOURce]:PULSe:TRANsition:FIXed? 'identifier'

Input Parameters

'identifier': 'M*.DATAOUT1', 'M*.DataOut2', or 'M*.DataOut'

<SMOoth | MODerate | STEep>

Return Range SMOoth | MODerate | STEep

Description Use this command to set/read the transition time of the data output stream.

Following is the expected transition time (tr) for the below mentioned settings:

1. Smooth: 20 ps typical (20%-80%)
2. Moderate: 17 ps typical (20%-80%)
3. Steep: 12 ps typical (20%-80%)

This SCPI is applicable for M8041A, M8051A and M8062A.

Example :PULS:TRAN:FIX 'M1.DataOut2', SMO


[:SOURce]:PULSe:TRANsition[:LEADing]?]

Syntax [:SOURce]:PULSe:TRANsition[:LEADing] 'identifier', <Nrf>

[:SOURce]:PULSe:TRANsition[:LEADing]? 'identifier'

Input Parameters

'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3' and 'M*.DataOut4'

<Nrf>: Specify transition time value.

Description Use this command to set/read the transition time of the data output stream. The fundamental units for transition time is seconds.

This SCPI is applicable for M8195A.
Example

:PUIS:TRAN 'M1.DataOut2',15e-12

:PUIS:TRAN? 'M1.DataOut2'

**NOTE**

It is not possible to define rise and fall times separately. In other words, the rise and fall time will be in the same range.
[:SOURce]:VOLTage Subnode

This subnode has the following SCPI structure:

```
[:SOURce]
- VOLTage
  - [:AMPLitude][?]
  - :CFAMplitude[?]
  - :OFFSet[?]
  - :HIGH[?]
  - :LOW[?]
  - :RANGe
    - [:SELect][?]
    - :AUTO[?]
```

This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:AMPLitude][?]</td>
<td>[:SOURce]:VOLTage[:AMPLitude][?] on page 150</td>
</tr>
<tr>
<td>CFAMplitude[?]</td>
<td>[:SOURce]:VOLTage:CFAMplitude[?] on page 150</td>
</tr>
<tr>
<td>:OFFSet[?]</td>
<td>[:SOURce]:VOLTage:OFFSet[?] on page 151</td>
</tr>
<tr>
<td>:HIGH[?]</td>
<td>[:SOURce]:VOLTage:HIGH[?] on page 151</td>
</tr>
<tr>
<td>:LOW[?]</td>
<td>[:SOURce]:VOLTage:LOW[?] on page 152</td>
</tr>
<tr>
<td>:RANGe[:SELect][?]</td>
<td>[:SOURce]:VOLTage:RANGe[:SELect][?] on page 152</td>
</tr>
<tr>
<td>:RANGe:AUTO[?]</td>
<td>[:SOURce]:VOLTage:RANGe:AUTO[?] on page 153</td>
</tr>
</tbody>
</table>
[:SOURce]:VOLTage[:AMPLitude][?]  
Syntax  
[:SOURce]:VOLTage[:AMPLitude] 'identifier', <NRf>  
[:SOURce]:VOLTage[:AMPLitude]? 'identifier'  
Input  
'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', 'M*.DataOut4',  
'M*.DataOut', or 'M*.CtrlOutA'  
Parameters  
<NRf>: Set peak to peak voltage amplitude.  
Description  
This command sets the peak to peak value of an output signal in Volts  
addressed by an identifier. Acceptable units include mV, V and exponents  
(for example, 500E-3 is the same as 500 mV).  
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8195A  
and M8045A.  
Example  
:SOUR:VOLT 'M1.DataOut1', 500mv  
:SOUR:VOLT? 'M1.DataOut1'

[:SOURce]:VOLTage:CFAMplitude[?]  
Syntax  
[:SOURce]:VOLTage:CFAMplitude 'Identifier', <NRf>  
[:SOURce]:VOLTage:CFAMplitude? 'Identifier'  
Input  
'identifier': 'M*.DataOut1' or 'M*.DataOut2' or 'M*.DataOut3' or  
'M*.DataOut4'  
Parameters  
This command sets the amplitude correction factor for each channel.  
Amplitude correction factor parameter will be functional only in channel  
coupled mode. For details on how to set channel coupling mode, see  
[:SOURce]:MEMory:CONFigure:COUPling[?] on page 143.  
This SCPI is applicable for M8195A and M8196A.  
Example  
:SOUR:VOLT:CFAM 'M1.DataOut1',1  
:SOUR:VOLT:CFAM? 'M1.DataOut1'

1
[:SOURce]:VOLTage:OFFSet[?]  
SYNTAX [:SOURce]:VOLTage:OFFSet 'identifier', <NRf>  
[:SOURce]:VOLTage:OFFSet? 'identifier'  

INPUT  

<NRf>: Set offset voltage.  

RETURN RANGE <NRf>  

DESCRIPTION This command sets the offset value of an output signal in Volts addressed by an identifier. Acceptable units include mV, V and exponents (for example, 500E-3 is the same as 500 mV). This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8195A and M8045A.  

Example :VOLT:OFFS 'M1.DataOut1', 500mv  
:VOLT:OFFS? 'M1.DataOut1'  

[:SOURce]:VOLTage:HIGH[?]  
SYNTAX [:SOURce]:VOLTage:HIGH 'identifier', <NRf>  
[:SOURce]:VOLTage:HIGH? 'identifier'  

INPUT  

<NRf>: Set upper voltage level.  

RETURN RANGE <NRf>  

DESCRIPTION This command sets the upper voltage level of an output signal in Volts addressed by an identifier. Acceptable units include mV, V and exponents (for example, 500E-3 is the same as 500 mV). This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8195A and M8045A.  

Example :VOLT:HIGH 'M1.DataOut1', 500mv  
:VOLT:HIGH? 'M1.DataOut1'
[:SOURce]:VOLTage:LOW['identifier', <NRf>]

Syntax [:SOURce]:VOLTage:LOW? 'identifier'


Parameters <NRf>: Set lower voltage level.

Return Range <NRf>

Description This command sets the lower voltage level of an output signal in Volts addressed by an identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8195A and M8045A.

Example :VOLT:LOW 'M2.DataOut2', 100 mv

[:SOURce]:VOLTage:RANGE[:SELection'] 'identifier', <R1|R2|R3|R4|R5|R6|R7|R8>

Syntax [:SOURce]:VOLTage:RANGe[:SELection]? 'identifier'

Input 'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

Parameters <R1|R2|R3|R4|R5|R6|R7|R8>: Select amplitude ranges.

Return Range R1|R2|R3|R4|R5|R6|R7|R8

Description Amplitude ranges guarantee a glitch free change of the amplitude value within a specified range. The upper and lower limits of these ranges are specified in the data sheet. Ranges are only specified for the DataOut channel.

This SCPI is applicable for M8041A, M8051A, M8061A and M8062A.

Example :VOLT:RANG 'M2.DataOut2', R4
[:SOURce]:VOLTage:RANGe:AUTO[?]

Syntax

[:SOURce]:VOLTage:RANGe:AUTO 'identifier', <ON|OFF>|0>

[:SOURce]:VOLTage:RANGe:AUTO? 'identifier'

Input

'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

Parameters

<ON|OFF>|0>: Enable/disable auto tracking.

Return Range

1|0

Description

Disable/enable the automatic tracking of the amplitude ranges.

This SCPI is applicable for M8041A, M8051A, M8061A and M8062A.

Example

:VOLT:RANG:AUTO 'M2.DataOut2', off
[:SOURce]:JITTer Subnode

This subnode has the following SCPI structure:

```
[:SOURce]
  :JITTer
    [:GLOBal]
      [:STATe][?]
    :CONFigure
      [:DELay][?]
    :HMODE[?]
    :HFRequency
    [...]
    :LFRequency
```

This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:GLOBal][:STATe][?]</td>
<td>[:SOURce]:JITTer[:GLOBal][:STATe][?] on page 155</td>
</tr>
<tr>
<td>:CONFigure[:DELay][?]</td>
<td>[:SOURce]:JITTer:CONFigure[:DELay][?] on page 155</td>
</tr>
<tr>
<td>:CONFigure:HMODE[?]</td>
<td>[:SOURce]:JITTer:CONFigure:HMODE[?] on page 155</td>
</tr>
<tr>
<td>:HFRequency</td>
<td>[:SOURce]:JITTer:HFRequency Subnode on page 157</td>
</tr>
<tr>
<td>:LFRequency</td>
<td>[:SOURce]:JITTer:LFRequency Subnode on page 174</td>
</tr>
</tbody>
</table>
[:SOURce]:JITTer[:GLOBal]:STATe[?] 
Syntax [:SOURce]:JITTer[:GLOBal]:STATe 'identifier', <ON|OFF|1|0> 
[:SOURce]:JITTer[:GLOBal]:STATe? 'identifier' 
Input 'identifier': 'M1.System' 
Parameters <ON|OFF|1|0>: Enable/disable global state. 
Return Range 1|0 
Description This command enables or disables global jitter. It is a global SCPI; working for all modules. 
Example :JITT:STAT 'M1.System', off 

[:SOURce]:JITTer:CONFigure[:DELay][?] 
Syntax [:SOURce]:JITTer:CONFigure[:DELay] 'identifier', <NRf> 
[:SOURce]:JITTer:CONFigure[:DELay]? 'identifier' 
Input 'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M1.ClkOut', or 'M*.DataOut' 
Parameters <NRf>: Set jitter profile delay. 
Return Range -40 ns to 40 ns 
Description Sets the delay of the jitter profile of the corresponding output. This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A. 

[:SOURce]:JITTer:CONFigure:HMODe[?] 
Syntax [:SOURce]:JITTer:CONFigure:HMODe 'identifier', < ON | OFF | 1 | 0 > 
[:SOURce]:JITTer:CONFigure:HMODe? 'identifier' 
Input 'identifier': 'M*.DataOut' 
Parameters < ON | OFF | 1 | 0 > 
Return Range 1 | 0
Description  Sets the High Jitter Injection Mode of the corresponding output.
   This SCPI is applicable for M8062A.
Example   JITT:CONF:HMOD 'M2.DataOut', OFF
[:SOURce]:JITTer:HFRequency Subnode

This subnode has the following SCPI structure:

```
[:SOURce]
  :JITer
   :HFRequency
    :UNIT[?]?
    :EXTernal
     [:STATe][?]?
    :PMMode[?]
    :PERiodic(*)[?]
    ...
    :BUNCorrelate
    ...
    :RANDom
    ...
    :SPECTrally
    ...
```

This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:UNIT[?]</td>
<td>[:SOURce]:JITTer:HFRequency:UNIT[?] on page 158</td>
</tr>
<tr>
<td>:EXTernal[:STATe][?]</td>
<td>[:SOURce]:JITTer:HFRequency:EXTernal[:STATe][?] on page 158</td>
</tr>
<tr>
<td>:PMMode[?]</td>
<td>[:SOURce]:JITTer:HFRequency:PMMode[?] on page 159</td>
</tr>
<tr>
<td>:PERiodic(*)[?]</td>
<td>[:SOURce]:JITTer:HFRequency:PERiodic(*)[:STATe][?] on page 160</td>
</tr>
<tr>
<td>:BUNCorrelate</td>
<td>[:SOURce]:JITTer:HFRequency:BUNCorrelate Subnode on page 163</td>
</tr>
</tbody>
</table>
### [:SOURce]:JITTER:HFRequency:UNIT[?] Syntax

- [:SOURce]:JITTER:HFRequency:UNIT 'identifier', <UINTerval|TIME>
- [:SOURce]:JITTER:HFRequency:UNIT? 'identifier'

### Input Parameters

- 'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'
- <UINTerval|TIME>: Specify jitter parameter units.

### Return Range

| UINT| TIME

### Description

Specifies whether the jitter parameters are to be specified/returned in seconds (TIME) or unit intervals (UINTerval).

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.

This SCPI is also available on M8195A for DataOut1 to 4.

### Example

:JITT:HFR:UNIT 'M1.DataOut1', TIME

### [:SOURce]:JITTER:HFRequency:EXTernal[:STATe][?] Syntax

- [:SOURce]:JITTER:HFRequency:EXTernal[:STATe] 'identifier', <ON|OFF|1|0>
- [:SOURce]:JITTER:HFRequency:EXTernal[:STATe]? 'identifier'

### Input Parameters

- 'identifier': 'M1.ClkOut', 'M*.DataOut1' or 'M*.DataOut2'
- <ON|OFF|1|0>: Enable/disable external jitter input.

### Return Range

| 1| 0

### Description

Enable/disable the external jitter input connector for Data Out1, Data Out2 and Clock Out. The name of the connectors are DATA MOD IN and CLK MOD IN.

This SCPI is applicable for M8041A, M8051A and M8045A.

### Example

:JITT:HFR:EXT:STAT 'M1.DataOut2', on
[\texttt{[:SOURce\textsubscript{Source}]:JITTer:HFRequency:PMMode[?]}]

Syntax
\texttt{[:SOURce\textsubscript{Source}]:JITTer:HFRequency:PMMode 'identifier', \textless\texttt{UI|TIME}\textgreater}
\texttt{[:SOURce\textsubscript{Source}]:JITTer:HFRequency:PMMode? 'identifier'}

Input Parameters
'identifier': 'M*.DataOut1' or 'M*.DataOut2'
\texttt{\langle\texttt{UI|TIME}\rangle}: Specify unit interval or time.

Return Range
\texttt{UI|TIME}

Description
The internal delay line is optimized for either 'UI' (1UI; the maximum of the delay line is not used) or for 'TIME' rather than the whole range of the delay line, which can be used independent of the data rate. PMMode stands for Phase Modulation Mode.

This SCPI is applicable for M8041A, M8051A and M8045A.

Example
\texttt{:JITT:HFR:PM 'M1.DataOut2', UI}
This subnode has the following SCPI structure:

```
[:SOURce]
  JITTer
  HFRequency
  PERiodic(*)
    [:STATe][?]
    :AMPLitude[?]
    :FREQuency[?]
```

This subnode has the following commands:

**Table 37**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:STATe][?]</td>
<td>[:SOURce]:JITTer:HFRequency:PERiodic(*)[:STATe][?] on page 160</td>
</tr>
<tr>
<td>:AMPLitude[?]</td>
<td>[:SOURce]:JITTer:HFRequency:PERiodic(*):AMPLitude[?] on page 161</td>
</tr>
<tr>
<td>:FREQuency[?]</td>
<td>[:SOURce]:JITTer:HFRequency:PERiodic(*)::FREQuency[?] on page 162</td>
</tr>
</tbody>
</table>

**[:SOURce]:JITTer:HFRequency:PERiodic(*)[:STATe][?]**

**Syntax**

```
[:SOURce]:JITTer:HFRequency:PERiodic(*)[:STATe] 'identifier', <ON|OFF>[1|0>]
```

**Input Parameters**

- 'identifier': 'M1.ClkOut', 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'
- <ON|OFF>[1|0]: Enable/disable periodic jitter.

**Return Range**

1|0

**Description**

This command is used to enable/disable periodic jitter high frequency output at the given location.
The "(*)" suffix specifies which of the two periodic sources (1 or 2) for the specified channel to use. If a suffix is not specified, the suffix 1 is assumed.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.

This SCPI is also available on M8195A for DataOut1 to 4.

Example: JITT:HFR:PER 'M2.DataOut2', off

[:SOURce]:JITTer:HFRequency:PERiodic(*):AMPLitude[?]

Syntax [:SOURce]:JITTer:HFRequency:PERiodic(*):AMPLitude 'identifier', <NRf>
[:SOURce]:JITTer:HFRequency:PERiodic(*):AMPLitude? 'identifier'

Input Parameters
- 'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'
- <NRf>: Set periodic jitter amplitude.

Return Range 0 mUI to 1.102 UI; The range is dependent on other system parameters. See Command Syntax to Find Min/Max Values on page 110.

Range for M8195A is 0 to 10 UI.

Description This command sets the amplitude of the periodic jitter in seconds or unit intervals for the specified output. The units are set using the [:SOURce]:JITTer:HFRequency:UNIT command.

The "(*)" suffix specifies which of the two periodic sources (1 or 2) for the specified channel to use. If a suffix is not specified, the suffix 1 is assumed.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.

This SCPI is also available on M8195A for DataOut1 to 4.

Example: JITT:HFR:PER:AMPL 'M2.DataOut2', 100e-12
URce]:JITTer:HFRequency:PERiodic(*):FREQuency[?

Syntax

[:SOURce]:JITTer:HFRequency:PERiodic(*):FREQuency 'identifier', <NRf>
[:SOURce]:JITTer:HFRequency:PERiodic(*):FREQuency? 'identifier'

Input Parameters

'identifier': 'M*.DataOut1' or 'M*.DataOut2', or 'M*.DataOut'

<NRF>: Set periodic jitter frequency.

Return Range

1 kHz to 500 MHz

Description

This command sets the frequency of the periodic jitter for the specified output.

The "(*)" suffix specifies which of the two periodic sources (1 or 2) for the specified channel to use. If a suffix is not specified, the suffix 1 is assumed.

Acceptable units include Hz, kHz, MHz and exponents (for example, 1E3 is the same as 1,000 Hz).

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.

This SCPI is also available on M8195A for DataOut1 to 4.

Example

This subnode has the following SCPI structure:

```
[:SOURce]
  .JITter
    .HFReactency
      .BUNCorrelate
        .:STATe[?]
        .:AMPLitude[?]
        .:DRAte[?]
        .:FILTER[:SELECT][?]
        .:PRBSequence[:SELECT][?]
```

This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:STATe][?]</td>
<td>[:SOURce]:JITter:HFRequency:BUNCorrelate[:STATe][?] on page 164</td>
</tr>
<tr>
<td>:AMPLitude[?]</td>
<td>[:SOURce]:JITter:HFRequency:BUNCorrelate:AMPLitude[?] on page 164</td>
</tr>
<tr>
<td>:DRAte[?]</td>
<td>[:SOURce]:JITter:HFRequency:BUNCorrelate:DRAte[?] on page 165</td>
</tr>
<tr>
<td>:FILTER[:SELECT][?]</td>
<td>[:SOURce]:JITter:HFRequency:BUNCorrelate:FILTER[:SELECT][?] on page 165</td>
</tr>
<tr>
<td>:PRBSequence[:SELECT][?]</td>
<td>[:SOURce]:JITter:HFRequency:BUNCorrelate:PRBSequence[:SELECT][?] on page 166</td>
</tr>
</tbody>
</table>
[:SOURce]:JITTER:HFRequency:BUNCorrelate[:STATe]?  
Syntax [:SOURce]:JITTER:HFRequency:BUNCorrelate[:STATe]? 'identifier'  
Input Parameters
identifier: 'M*.DataOut1', 'M*.DataOut2', 'M1.ClkOut', or 'M*.DataOut' 
<ON|OFF|1|0>: Enable/disable bounded uncorrelated jitter. 
Return Range 1|0 
Description Enables/disables the generation of bounded uncorrelated jitter at the specified output. 
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A. 
Example :JITT:HFR:BUNC 'M2.DataOut2', off

[:SOURce]:JITTER:HFRequency:BUNCorrelate:AMPLitude[]  
Syntax [:SOURce]:JITTER:HFRequency:BUNCorrelate:AMPLitude 'identifier', <NRf>  
[:SOURce]:JITTER:HFRequency:BUNCorrelate:AMPLitude? 'identifier' 
Input Parameters 
identifier: 'M*.DataOut1' or 'M*.DataOut2', or 'M*.DataOut' 
<NRf>: Set bounded uncorrelated jitter amplitude. 
Return Range NRf 
Description This command sets the bounded uncorrelated jitter amplitude in seconds (TIME) or unit intervals (UINTerval). The units are set using the [:SOURce]:JITTER:HFRequency:UNIT command. 
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A. 
Example :JITT:HFR:BUNC:AMPL 'M2.DataOut2', 400e-12
[[:SOURce]:JITTer:HFRequency:BUNCorrelate:DRATe?] Syntax

[[:SOURce]:JITTer:HFRequency:BUNCorrelate:DRATe 'identifier' 'RATE625|RATE1250|RATE2500']

Input Parameters

'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

Return Range RATE625|RATE1250|RATE2500

Description

This command sets the bounded uncorrelated PRBS data rate to 625 MBps, 1250 MBps, or 2500 MBps.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.

Example :JITT:HFR:BUNC:DRAT 'M2.DataOut2', RATE625

[[:SOURce]:JITTer:HFRequency:BUNCorrelate:FILTer[:SELect]?]

Syntax

[[:SOURce]:JITTer:HFRequency:BUNCorrelate:FILTer[:SELect] 'identifier' 'LP50|LP100|LP200']

Input Parameters

'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

Return Range LP50|LP100|LP200

Description

This command selects the low-pass filter for bounded uncorrelated jitter (50, 100, or 200 MHz).

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.

Example :JITT:HFR:BUNC:FILT 'M2.DataOut2', LP50
[:SOURce]:JITTer:HFRequency:BUNCorrelate:PRBSequence[:SELect]?

Syntax

[:SOURce]:JITTer:HFRequency:BUNCorrelate:PRBSequence[:SELect]
'identifier',
<PRBS7|PRBS8|PRBS9|PRBS10|PRBS11|PRBS15|PRBS23|PRBS31
[:SOURce]:JITTer:HFRequency:BUNCorrelate:PRBSequence[:SELect]?
'identifier'

Input Parameters

'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'
<PRBS7|PRBS8|PRBS9|PRBS10|PRBS11|PRBS15|PRBS23|PRBS31>: Select the PRBS polynomial.

Return Range

PRBS7|PRBS8|PRBS9|PRBS10|PRBS11|PRBS15|PRBS23|PRBS31

Description

This command selects polynomial of the PRBS for bounded uncorrelated jitter source.
This query returns the current setting.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.

Example

This subnode has the following SCPI structure:

```
[:SOURce]::JTTer::HFRequency::RANDom
```

This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:STATe]?</td>
<td>[:SOURce]:JTTer::HFRequency::RANDom[:STATe]? on page 168</td>
</tr>
<tr>
<td>:AMPLitude[?]</td>
<td>[:SOURce]:JTTer::HFRequency::RANDom::AMPLitude[?]</td>
</tr>
<tr>
<td>:FILTer[:LPASs][?]</td>
<td>[:SOURce]:JTTer::HFRequency::RANDom::FILTer[:LPASs][?] on page 169</td>
</tr>
<tr>
<td>:FILTer[:LPASs]:VALue[?]</td>
<td>[:SOURce]:JTTer::HFRequency::RANDom::FILTer[:LPASs]:VALue[?] on page 169</td>
</tr>
<tr>
<td>:FILTer:HPASs[?]</td>
<td>[:SOURce]:JTTer::HFRequency::RANDom::FILTer:HPASs[?]   on page 170</td>
</tr>
<tr>
<td>:FILTer:HPASs:VALue[?]</td>
<td>[:SOURce]:JTTer::HFRequency::RANDom::FILTer:HPASs:VALue[?] on page 170</td>
</tr>
</tbody>
</table>
[:SOURce]:JITTer:HFRequency:RANDom[:STATe]?[]

Syntax [:SOURce]:JITTer:HFRequency:RANDom[:STATe] 'identifier' <ON|OFF|1|0>
[:SOURce]:JITTer:HFRequency:RANDom[:STATe]? 'identifier'

Input 'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M1.ClkOut', or 'M*.DataOut'

Parameters <ON|OFF|1|0>: Enable/disable random jitter source.

Return Range 0|1

Description This command enables/disables the generation of random jitter.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.

This SCPI is also available on M8195A for DataOut1 to 4.

Example :JITT:HFR:RAND 'M2.DataOut2', on

[:SOURce]:JITTer:HFRequency:RANDom:AMPLitude[]

Syntax [:SOURce]:JITTer:HFRequency:RANDom:AMPLitude 'identifier' <NRf>
[:SOURce]:JITTer:HFRequency:RANDom:AMPLitude? 'identifier'

Input 'identifier': 'M*.DataOut1' or 'M*.DataOut2', or 'M*.DataOut'

Parameters <NRf>: Set random jitter amplitude.

Return Range 0 uUI to 13.08 mUI or 0 ps to 78.57 ps

Description This command sets the root mean square (rms) RJ amplitude in seconds or unit intervals based on the selected amplitude unit. The units are set using the [:SOURce]:JITTer:HFRequency:UNIT command.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.

This SCPI is also available on M8195A for DataOut1 to 4.

Example :JITT:HFR:RAND:AMPL 'M2.DataOut2', 0.013
[:SOURce]:JITTer:HFRequency:RANDom:FILTER:LPASSes? 

Syntax [:SOURce]:JITTer:HFRequency:RANDom:FILTER:LPASSes 'identifier'

Input Parameters 'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

Return Range OFF|LP100|LP500

Description This command enables/disables the low-pass filter for random jitter. LP100 enables a 100 MHz low-pass filter; LP500 enables a 500 MHz low-pass filter.

This query returns the current settings.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.

Example :JITT:HFR:RAND:FILT 'M1.DataOut2', LP100

[:SOURce]:JITTer:HFRequency:RANDom:FILTER:LPASSes:VALUE? 

Syntax [:SOURce]:JITTer:HFRequency:RANDom:FILTER:LPASSes:VALUE <identifier>,<NRf>

Input Parameters 'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', 'M*.DataOut4', or 'M*.DataOut'

Return Range 1 MHz to 10 GHz

Description This command controls the low pass filter of the random jitter source. The limit is dynamic, it depends on the high-pass setting.

The query returns the current settings.

This SCPI is applicable for M8195A and M8196A.

[:SOURce]:JITTer:HFRequency:RANDom:FIltTer:HPAsS[?]?

Syntax

[:SOURce]:JITTer:HFRequency:RANDom:FIltTer:HPAsSs 'identifier'
   <OFF|HP10>

[:SOURce]:JITTer:HFRequency:RANDom:FIltTer:HPAsSs? 'identifier'

Input Parameters

'identifier': 'M*.DataOut1' or 'M*.DataOut2', or 'M*.DataOut'

<OFF|HP10>: Enable/disable high-pass filter.

Return Range

OFF|HP10

Description

This command enables/disables the high-pass filter for random jitter.
HP10 enables a 10 MHz high-pass filter.
This query returns the current settings.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.

Example


[:SOURce]:JITTer:HFRequency:RANDom:FIltTer:HPAsSs:VALue[?]?

Syntax

[:SOURce]:JITTer:HFRequency:RANDom:FIltTer:HPAsSs 'identifier' <NRf>

[:SOURce]:JITTer:HFRequency:RANDom:FIltTer:HPAsSs:VALue? 'identifier'

Input Parameters

'identifier': 'M*.DataOut1' or 'M*.DataOut2', or 'M*.DataOut'

<NRf>

Return Range

1 kHz to 10 MHz
The limit is dynamic, it depends on the low-pass setting.

Description

This command controls the high pass filter value of the random jitter source.
This SCPI is applicable for M8195A and M8196A.
This query returns the current settings.

Example

Spectrally distributed random jitter is composed of two jitter sources: low frequency jitter and high frequency jitter. It is characterized by the amplitudes of the low and the high frequency jitter part.

This subnode has the following SCPI structure:

```
[:SOURce]:JITTer:HFRequency:SPECtrally Subnode
```

This subnode has the following commands:

<table>
<thead>
<tr>
<th>Table 40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>[:STATe]?</td>
</tr>
<tr>
<td>:AMPLitude]?</td>
</tr>
<tr>
<td>:FILTer:LPASs[:STATe]?</td>
</tr>
</tbody>
</table>
[:SOURce]:JITTer:HFRequency:SPECtrally[:STATe]?

Syntax [:SOURce]:JITTer:HFRequency:SPECtrally[:STATe] 'identifier'
<ON|OFF|1|0>

[:SOURce]:JITTer:HFRequency:SPECtrally[:STATe]? 'identifier'

Input 'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M1.ClkOut', or 'M*.DataOut'

Parameters <ON|OFF|1|0>: Enable/disable spectrally distributed jitter.

Return Range 0|1

Description This command enables/disables the generation of spectrally distributed random jitter.

This SCPI is applicable for M8041A, M8051A, M8061A and M8062A.

Example :JITT:HFR:SPEC 'M2.DataOut2', off

[:SOURce]:JITTer:HFRequency:SPECtrally:AMPLitude(*)?

Syntax [:SOURce]:JITTer:HFRequency:SPECtrally:AMPLitude(*) 'identifier' <NRf>

[:SOURce]:JITTer:HFRequency:SPECtrally:AMPLitude(*)? 'identifier'

Input 'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

Parameters <NRf>: Set spectrally distributed jitter amplitude.

Return Range 0 mUI to 78.71 mUI

Description This command sets the spectrally distributed jitter amplitude in unit intervals or time.

A suffix attached to the :AMPLitude(*) command is used to address two jitter amplitudes. The :AMPL1 <NR3> command changes the level of the internal RJLF resource and the :AMPL2 <NR3> command changes the level of the internal RJHF resource. Both levels are specified in rms.

The units are set using the [:SOURce]:JITTer:HFRequency:UNIT command.

This SCPI is applicable for M8041A, M8051A, M8061A and M8062A.

Example :JITT:HFR:SPEC:AMPL2 'M2.DataOut2', 0.013
[:SOURce]:JITTer:HFRequency:SPECtrally:FILTer[:LPASs][:STATe]?}

**Syntax**

[:SOURce]:JITTer:HFRequency:SPECtrally:FILTer[:LPASs] [:STATe] 'identifier' <ON|OFF|1|0>

[:SOURce]:JITTer:HFRequency:SPECtrally:FILTer[:LPASs] [:STATe]?

**Input Parameters**

'identifier': 'M*.DataOut1' or 'M*.DataOut2'

<ON|OFF|1|0>: Enable/disable low-pass filter.

**Return Range**

1|0

**Description**

Enables/disables the 100 MHz low-pass filter for spectrally distributed random jitter.

This SCPI is applicable for M8041A, M8051A, M8061A and M8062A.

**Example**

:JITT:HFR:SPEC:FILT 'M2.DataOut2', off
This subnode has the following SCPI structure:

```
[:SOURce]::JITTer::LFRequency
```

This subnode has the following commands:

Table 41

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:UNIT[?]</td>
<td>[:SOURce]:JITTer::LFRequency:UNIT[?] on page 175</td>
</tr>
<tr>
<td>:PERiodic</td>
<td>[:SOURce]:JITTer::LFRequency:PERiodic Subnode on page 176</td>
</tr>
<tr>
<td>:RSSClocking</td>
<td>[:SOURce]:JITTer::LFRequency:RSSClocking Subnode on page 178</td>
</tr>
</tbody>
</table>
[:SOURce]:JITTER:LFRequency:UNIT[?]

**Syntax**

[:SOURce]:JITTER:LFRequency:UNIT 'identifier', <UINTerval|TIME>

[:SOURce]:JITTER:LFRequency:UNIT? 'identifier'

**Input Parameters**

- 'identifier': 'M1.ClkOut', 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'
- <UINTerval|TIME>: Specify jitter parameter units.

**Return Range**

 UINT|TIME

**Description**

Specifies whether the jitter parameters are to be specified/returned in seconds (TIME) or unit intervals (UINTerval).

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.

**Example**

:JITT:LFR:UNIT 'M1.DataOut2',TIME
This subnode has the following SCPI structure:

```
[:SOURce][:JITTer][:LFRequency][:PERiodic][:STATe][?]
```

This subnode has the following commands:

### Table 42

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:STATe][?]</td>
<td>[:SOURce][:JITTer][:LFRequency][:PERiodic][:STATe][?] on page 176</td>
</tr>
<tr>
<td>:AMPLitude[?]</td>
<td>[:SOURce][:JITTer][:LFRequency][:PERiodic]:AMPLitude[?] on page 177</td>
</tr>
<tr>
<td>:FREQuency[?]</td>
<td>[:SOURce][:JITTer][:LFRequency][:PERiodic]:FREQuency[?] on page 177</td>
</tr>
</tbody>
</table>

### [:SOURce]:JITTer:LFRequency:PERiodic[:STATe][?]

**Syntax**

```
[:SOURce]:JITTer:LFRequency:PERiodic[:STATe][?] 'identifier', <ON|OFF|1|0>
```

**Input Parameters**

- <ON|OFF|1|0>: Enable/disable the low frequency periodic jitter source.

**Return Range**

1|0
Description
This command enables/disables the low frequency periodic jitter source. This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.

Example
```
```

```
[:SOURce]:JITTer:LFRequency:PERiodic:AMPLitude?[?]
```

Syntax
```
[:SOURce]:JITTer:LFRequency:PERiodic:AMPLitude 'identifier', <NRf>
```

Input
- `identifier`: 'M*.DataOut1', 'M*.DataOut2', 'M1.ClkOut', or 'M*.DataOut'
- `<NRf>`: Set periodic low frequency jitter amplitude.

Return Range
NRf

Description
This command sets the amplitude of the periodic low frequency jitter in unit intervals or time for the specified output. The units are set using the [:SOURce]:JITTer:LFRequency:UNIT command.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.

Example
```
:JITT:LFR:PER:AMPL 'M2.DataOut2', 1
```

```
[:SOURce]:JITTer:LFRequency:PERiodic:FREQuency?[?]
```

Syntax
```
[:SOURce]:JITTer:LFRequency:PERiodic:FREQuency 'identifier', <NRf>
```

Input
- `identifier`: 'M*.DataOut1', 'M*.DataOut2', 'M1.ClkOut', or 'M*.DataOut'
- `<NRf>`: Set periodic jitter frequency.

Return Range
NRf

Description
This command sets the frequency of the periodic jitter. Acceptable units include Hz, kHz, MHz and exponents (for example, 1E3 is the same as 1,000 Hz).

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.

Example
```
:JITT:LFR:PER:FREQ 'M2.DataOut2', 1e6
```
[:SOURce]:JITTter:LFRequency:RSSClocking Subnode

This subnode has the following SCPI structure:

This subnode has the following commands:

Table 43

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:STATe]?</td>
<td>[:SOURce]:JITTter:LFRequency:RSSClocking[:STATe]? on page 178</td>
</tr>
<tr>
<td>:AMPLitude</td>
<td>[:SOURce]:JITTter:LFRequency:RSSClocking:AMPLitude on page 179</td>
</tr>
</tbody>
</table>

[:SOURce]:JITTter:LFRequency:RSSClocking[:STATe]? Syntax

[:SOURce]:JITTter:LFRequency:RSSClocking[:STATe]? 'identifier',

<ON|OFF|1|0>

[:SOURce]:JITTter:LFRequency:RSSClocking[:STATe]? 'identifier'

Input Parameters

'identifier': 'M1.ClkOut', 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

Return Range

1|0

Description

This command enables/disables the residual spread spectrum clocking (rSSC).
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8145A.

Example :JITT:LFR:RSSC 'M2.DataOut2', on

[ :SOURce]:JITTer:LFRequency:RSSClocking:AMPLitude[?]

Syntax [ :SOURce]:JITTer:LFRequency:RSSClocking:AMPLitude 'identifier', <NRf>
[ :SOURce]:JITTer:LFRequency:RSSClocking:AMPLitude? 'identifier'

Input Parameters
'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M1.ClkOut', or 'M*.DataOut'

<NRf>: Set residual spread spectrum clocking amplitude.

Return Range NRf

Description This command sets the amplitude of the residual spread spectrum clocking (rSSC). The units are set using the [ :SOURce]:JITTer:LFRequency:UNIT command. This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8145A.

Example :JITT:LFR:RSS:AMPL 'M2.DataOut2', 0.016

[ :SOURce]:JITTer:LFRequency:RSSClocking:FREQuency[?]

Syntax [ :SOURce]:JITTer:LFRequency:RSSClocking:FREQuency 'identifier', <NRf>
[ :SOURce]:JITTer:LFRequency:RSSClocking:FREQuency? 'identifier'

Input Parameters
'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M1.ClkOut', or 'M*.DataOut'

<NRf>: Set residual spread spectrum clocking frequency.

Return Range 10 kHz to 100 kHz

Description This command sets the frequency of the residual spread spectrum clocking (rSSC). Acceptable units include Hz, kHz and exponents (for example, 10E3 is the same as 10,000 Hz).

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8145A.

Example :JITT:LFR:RSSC:FREQ 'M2.DataOut1', 10khz
[:SOURce]:JITTer:SWEep Subnode

This subnode has the following SCPI structure:

```
[:SOURce]
  [:JITTer]
  [:SWEep]
    [:STATe]? 
    [:FREQuency]
      [:STARt]? 
      [:STOP]? 
    [:TIME]?
    [:STEP]
      [:DISTance]? 
      [:VALue]?
    [:AMPLitude]
    [:MODE]? 
    [:VALue]?
    [:PLOT]
      [:INPut]? 
      [:VALue]?
    [:DATA]
    [:FILE]?
```

This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:STATe]?</td>
<td>[:SOURce]:JITTER:SWEep[:STATe]? on page 182</td>
</tr>
<tr>
<td>[:FREQuency:STARt]?</td>
<td>[:SOURce]:JITTER:SWEep:FREQuency:STARt? on page 182</td>
</tr>
<tr>
<td>[:FREQuency:STOP]?</td>
<td>[:SOURce]:JITTER:SWEep:FREQuency:STOP? on page 182</td>
</tr>
<tr>
<td>[:TIME]?</td>
<td>[:SOURce]:JITTER:SWEep:TIME? on page 183</td>
</tr>
<tr>
<td>Name</td>
<td>Description under</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>:STEP:[VALUE]?</td>
<td>[:SOURce]:JITTer:SWEep:STEP:[VALUE]? on page 183</td>
</tr>
<tr>
<td>:STEP:DISTance[]?</td>
<td>[:SOURce]:JITTer:SWEep:STEP:DISTance[]? on page 184</td>
</tr>
<tr>
<td>:AMPLitude[:VALUE]?</td>
<td>[:SOURce]:JITTer:SWEep:AMPLitude[:VALUE]? on page 184</td>
</tr>
<tr>
<td>:AMPLitude:MODE[]?</td>
<td>[:SOURce]:JITTer:SWEep:AMPLitude:MODE[]? on page 184</td>
</tr>
<tr>
<td>:AMPLitude[:VALUE]?</td>
<td>[:SOURce]:JITTer:SWEep:AMPLitude[:VALUE]? on page 184</td>
</tr>
<tr>
<td>:PLOT[:VALUE]?</td>
<td>[:SOURce]:JITTer:SWEep:PLOT[:VALUE]? on page 185</td>
</tr>
<tr>
<td>:DATA:FILE[]?</td>
<td>[:SOURce]:JITTer:SWEep:DATA:FILE[]? on page 186</td>
</tr>
</tbody>
</table>
[:SOURce]:JITTer:SWEep[:STATe]?  
Syntax  [:SOURce]:JITTer:SWEep[:STATe] "identifier", <ON|OFF|1|0>  
[:SOURce]:JITTer:SWEep[:STATe]? 'identifier'  
Input  'identifier': 'M*.DataOut1' or 'M*.DataOut2'  
Parameters  <ON|OFF|1|0>: Enable/disable periodic jitter sweep.  
Return Range  1|0  
Description  This command enables/disables the jitter sweep.  
This SCPI is applicable for M8041A, M8051A and M8045A.  
Example  :JITT:SWE 'M1.DataOut2', on

[:SOURce]:JITTer:SWEep:FREQuency:STARt?  
Syntax  [:SOURce]:JITTer:SWEep:FREQuency:STARt 'identifier', <NRf>  
[:SOURce]:JITTer:SWEep:FREQuency:STARt? 'identifier'  
Input  'identifier': 'M*.DataOut1' or 'M*.DataOut2'  
Parameters  <NRf>: Set the start frequency.  
Return Range  1 kHz to 500 MHz  
Description  This command defines the start frequency of the jitter sweep. The start frequency must be lower than the stop frequency and the range should be in accordance with the waveform. Acceptable units include Hz, kHz, MHz, GHz and exponents (for example, 1E3 is the same as 1,000 Hz).  
This SCPI is applicable for M8041A, M8051A and M8045A.  
Example  :JITT:SWE:FREQ:STAR 'M1.DataOut2', 500kHz

[:SOURce]:JITTer:SWEep:FREQuency:STOP?  
Syntax  [:SOURce]:JITTer:SWEep:FREQuency:STOP 'identifier', <NRf>  
[:SOURce]:JITTer:SWEep:FREQuency:STOP? 'identifier'  
Input  'identifier': 'M*.DataOut1' or 'M*.DataOut2'  
Parameters  <NRf>: Set the stop frequency.  
Return Range  1 kHz to 500 MHz
Description: This command defines the stop frequency of the jitter sweep. The stop frequency must be higher than the start frequency and the range should be in accordance with the waveform. Acceptable units include Hz, kHz, MHz, GHz and exponents (for example, 1E3 is the same as 1,000 Hz).


[:SOURce]:JITTer:SWEep:TIME[\?]

Syntax: `[:SOURce]:JITTer:SWEep:TIME 'identifier', <NRf>`

Input Parameters:
- `'identifier'`: 'M*.DataOut1' or 'M*.DataOut2'
- `<NRf>`: Set duration of sweep.

Return Range: 100 ms to 1.2 ks

Description: This command defines the time duration of the jitter sweep. Acceptable units include ms (milliseconds), s (seconds), ks (kiloseconds) and exponents (for example, 1E3 is the same as 1,000 seconds).

Example: `:JITT:SWE:TIME 'M1.DataOut2', 100ms`

[:SOURce]:JITTer:SWEep:STEP:[VALue][\?]

Syntax: `[:SOURce]:JITTer:SWEep:STEP:[VALue] 'identifier', <NRf>`

Input Parameters:
- `'identifier'`: 'M*.DataOut1' or 'M*.DataOut2'
- `<NRf>`: Set number of steps for a complete sweep.

Return Range: 2 to 100

Description: This command defines the number of steps to fulfill a complete jitter sweep.

Example: `:JITT:SWE:STEP 'M1.DataOut2', 50`
[:SOURce]:JITTER:SWEep:STEP:DISTance[?]

Syntax [:SOURce]:JITTER:SWEep:STEP :DISTance 'identifier', <EQUidistant|AUTOmatic>

[:SOURce]:JITTER:SWEep:STEP :DISTance? 'identifier'

Input 'identifier': 'M*.DataOut1' or 'M*.DataOut2'

Parameters <EQUidistant|AUTOmatic>: Specify log equidistant or frequency steps.

Return Range EQU|AUTO

Description The command defines whether the frequency steps are log equidistant (EQUidistant) along the periodic jitter curve or a frequency step matches a corner frequency on the jitter curve.

This SCPI is applicable for M8041A, M8051A and M8045A.

Example :JITT:SWE:STEP:DIST 'M1.DataOut2', AUTO

[:SOURce]:JITTER:SWEep:AMPLitude[:VALue]?

Syntax [:SOURce]:JITTER:SWEep:AMPLitude[:VALue] 'identifier', <NR3>

[:SOURce]:JITTER:SWEep:AMPLitude[:VALue]? 'identifier'

Input 'identifier': 'M*.DataOut1' or 'M*.DataOut2'

Parameters <NR3>: Specify constant jitter amplitude of the jitter sweep.

Return Range 0 mUI to 1.102 UI

Description This command sets the constant jitter amplitude of the jitter sweep. If CONStant jitter amplitude is selected as the jitter sweep mode, this command is used to define the value of the constant jitter amplitude of the jitter sweep. Acceptable units include mUI, UI and exponents (for example, 100e-3 is the same as 100 mUI).

This SCPI is applicable for M8041A, M8051A and M8045A.

Example :JITT:SWE:AMPL 'M1.DataOut2', 1UI

[:SOURce]:JITTER:SWEep:AMPLitude:MODE[?]

Syntax [:SOURce]:JITTER:SWEep:AMPLitude:MODE 'identifier',<CONStant|VARiable>

[:SOURce]:JITTER:SWEep:AMPLitude:MODE? 'identifier'
Input Parameters: 'identifier': 'M*.DataOut1' or 'M*.DataOut2'

Description: The command defines whether a sweep is defined with CONStant jitter level (amplitude) or with the user defined VARiable jitter level (amplitude). This SCPI is applicable for M8041A, M8051A and M8045A.

Example: JITT:SWE:AMPL:MODE 'M1.DataOut2', CONS

[:SOURce]:JITTer:SWEep:PLOT[:VALue]?

Syntax: [:SOURce]:JITTer:SWEep:PLOT[:VALue]? 'identifier'

Input Parameters: 'identifier': 'M*.DataOut1' or 'M*.DataOut2'

Return Range: <frequency/jitter amplitude value pairs>

Description: This query returns a comma separated list (specified as an expression) of frequency/jitter amplitude value pairs. This list of values represents the 'real' steps performed by the jitter sweep. This SCPI is applicable for M8041A, M8051A and M8045A.

Example: JITT:SWE:PLOT? 'M1.DataOut2'

[:SOURce]:JITTer:SWEep:PLOT:INPut?

Syntax: [:SOURce]:JITTer:SWEep:PLOT:INPut? 'identifier'

Input Parameters: 'identifier': 'M*.DataOut1' or 'M*.DataOut2'

Description: This query returns a comma separated list (specified as an expression) of frequency/jitter amplitude value pairs. This list of values represents the value pairs of the corner frequencies only. This SCPI is applicable for M8041A, M8051A and M8045A.

[[:SOURce]:JITTer:SWEep:DATA:FILE[?]]

Syntax
[:SOURce]:JITTer:SWEep:DATA:FILE 'identifier' '<FileName>'
[:SOURce]:JITTer:SWEep:DATA:FILE? 'identifier'

Input Parameters
'identifier': 'M*.DataOut1' or 'M*.DataOut2'
'<FileName>': Specify path to a data file.

Description
Every point of the periodic jitter sweep curve can be specified by a group of 2 value pairs Frequency, JitterLevel, Frequency, JitterLevel, etc. This command specifies the location of the file to be loaded within the workspace.

Jitter sweep profiles can be stored in three different areas:
1. Local to current setting ('current/')
2. Shared between settings ('shared/')
3. Factory supplied standard sweep profiles ('factory/').
   These sweep profiles are read only and cannot be modified.

Below these root nodes there is a folder structure using '/' as separation character. So, a complete file name might be something like "factory/PCIe_8G_CC.jcs", or "shared/subfolder/mySweepProfile.jcs".

This SCPI is applicable for M8041A, M8051A and M8045A.

Example
:JITT:SWE:DATA:FILE 'M1.DataOut2', 'factory/PCIe_8G_CC.jcs'
Spread Spectrum Clocking (SSCL) is used for reducing the peak electromagnetic radiation. The clock out signal and data out stream are modulated with a repetitive low-frequency waveform.

This subnode has the following SCPI structure:

```
[:SOURce]
  [:SSCLocking
    [:STATe][?]
    [:GLOBal[:STATe][?]
    [:DEViation][?]
    [:TYPE][?]
    [:PROFile][?]
    [:SHAPe][?]
    [:FREQuency][?]]
```

This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:STATe][?]</td>
<td>[:SOURce]:SSCLocking[:STATe][?] on page 188</td>
</tr>
<tr>
<td>GLOBal[:STATe][?]</td>
<td>[:SOURce]:SSCLocking:GLOBal[:STATe][?] on page 188</td>
</tr>
<tr>
<td>[:DEViation][?]</td>
<td>[:SOURce]:SSCLocking:DEViation[:VALue][?] on page 189</td>
</tr>
<tr>
<td>[:TYPE][?]</td>
<td>[:SOURce]:SSCLocking:TYPE[?] on page 190</td>
</tr>
<tr>
<td>[:PROFile][?]</td>
<td>[:SOURce]:SSCLocking:PROFile[?] on page 191</td>
</tr>
<tr>
<td>[:SHAPe][?]</td>
<td>[:SOURce]:SSCLocking:SHAPe[?] on page 192</td>
</tr>
<tr>
<td>[:FREQuency][?]</td>
<td>[:SOURce]:SSCLocking:FREQuency[?] on page 192</td>
</tr>
</tbody>
</table>
[:SOURce]:SSCLocking[:STATe]?[]

Syntax [:SOURce]:SSCLocking[:STATe] 'identifier', <ON|OFF|1|0>
[:SOURce]:SSCLocking[:STATe]? 'identifier'

Input 'identifier': 'M1.ClkGen'

Parameters <ON|OFF|1|0>: Enable/disable spread spectrum clocking.

Return Range 1|0

Description This command enables/disables spread spectrum clocking (SSC) state. The SSC mode can be only enabled if Global SSC state is activated.

This SCPI is applicable for M8041A, M8195A, M8196A and M8045A.

Example The below mentioned SCPI command turns ON the SSC state:
:SCL 'M1.ClkGen', ON
The below mentioned SCPI command queries the SSC:
:SCL? 'M1.ClkGen'
1

:SOURce]:SSCLocking:GLOBal[:STATe]?[]

Syntax [:SOURce]:SSCLocking:GLOBal[:STATe] 'identifier', <ON|OFF|1|0>
[:SOURce]:SSCLocking:GLOBal[:STATe]? 'identifier'

Input 'identifier': 'M1.System'

Parameters <ON|OFF|1|0>: Enable/disable spread spectrum clocking.

Return Range 1|0

Description This command turns the spread spectrum clocking (SSC) mode globally on or off. This means that SSC will be only enable for a specific module if the module specific SSC switch and the global SSC switch are both turned on, otherwise SSC is disabled.

[:SOURce]:SSCLocking:DEViation[:VALue]?\n
Syntax  [:SOURce]:SSCLocking:DEViation[:VALue] 'identifier', <NRf>  [:SOURce]:SSCLocking:DEViation[:VALue]? 'identifier'

Input  'identifier': 'M1.ClkGen'

Parameters  <NRf>: Set spread spectrum clocking deviation.

Return Range  0% to 1%

Description  This command sets the spread spectrum clocking (SSC) deviation in percent. The effective deviation is ±0.5% (or 1% peak-to-peak).

This SCPI is applicable for M8041A, M8195A, M8196A and M8045A.

Example  The below mentioned SCPI command sets the spread spectrum clocking (SSC) deviation to '0.2' percent for M8041A/M8045A:

```
SSCL:DEV 'M1.ClkGen', 0.2
```

The below mentioned SCPI command queries the spread spectrum clocking (SSC) deviation for M8041A/M8045A:

```
SSCL:DEV? 'M1.ClkGen'
```

0.2

[:SOURce]:SSCLocking:DEViation:UP[?]\n
Syntax  [:SOURce]:SSCLocking:DEViation:UP 'identifier', <NRf>  [:SOURce]:SSCLocking:DEViation:UP? 'identifier'

Input  'identifier': 'M1.ClkGen'

Parameters  <NRf>: Set peak up spread spectrum clocking deviation.

Description  This command sets the up deviation from the currently set clock rate. Value range is from 0.0 % to +1.0 %. Sending this command will automatically change the SSC type to ASYMmetric.

This SCPI is applicable for M8041A and M8045A.

Example  The below mentioned SCPI command sets the up deviation to '0.25' percent for M8041A/M8045A:

```
SSCL:DEV:UP 'M1.ClkGen', 0.25
```
The below mentioned SCPI command queries the up deviation for M8041A/M8045A:

SSCL:DEV:UP? 'M1.ClkGen'

0.25

[[:SOURce]:SSCLocking:DEViation:DOWN[?]]

Syntax  
[:SOURce]:SSCLocking:DEViation:DOWN 'identifier', <NRf>
[:SOURce]:SSCLocking:DEViation:DOWN? 'identifier'

Input Parameters

'identifier': 'M1.ClkGen'

<NRf>: Set spread spectrum clocking deviation.

Description

This command sets the down deviation from the currently set clock rate. Value range is from -1.0 % to 0.0 %. Sending this command will automatically change the SSC type to ASYMmetric.

This SCPI is applicable for M8041A and M8045A.

Example

The below mentioned SCPI command sets down deviation to '-0.45' percent for M8041A/M8045A:

SSCL:DEV 'M1.ClkGen', -0.45

The below mentioned SCPI command queries the down deviation for M8041A/M8045A:

SSCL:DEV? 'M1.ClkGen'

-0.45

[[:SOURce]:SSCLocking:TYPE[?]]

Syntax  
[:SOURce]:SSCLocking:TYPE 'identifier', <DOWNspread|UPSPread|CENTerspread|ASYMmetric>
[:SOURce]:SSCLocking:TYPE? 'identifier'

Input Parameters

'identifier': 'M1.ClkGen'

<DOWNspread|UPSPread|CENTerspread|ASYMmetric>: Set spread spectrum clocking type.

Return Range

DOWN|UPSP|CENT|ASYM
This command sets method used to modulate the clock frequency and data. Choose between center spread, down spread, up spread or asymmetric. For SSC type as ASYMmetric, you have to individually specify up and down deviation. For details, see [:SOURce]:SSCLocking:DEViation:UP[?] on page 189 and [:SOURce]:SSCLocking:DEViation:DOWN[?] on page 190.

This SCPI is applicable for M8041A, M8095A and M8045A.

It is also applicable for M8195A but only in query mode.

Example
The below mentioned SCPI command sets the method used to modulate clock frequency and data to 'DOWN' for M8041A/M8045A:

[:SOURce]:SSCLocking:PROFile 'identifier', <TRIangular|ARBitrary>

[:SOURce]:SSCLocking:PROFile? 'identifier'

Input
Parameters
Return Range
Description
Example

[:SOURce]:SSCLocking:PROFile[?]

This command sets up the spread spectrum clocking (SSC) using a triangular or arbitrary profile.

This SCPI is applicable for M8041A and M8045A.

It is also applicable for M8195A but only in query mode.

Example

:SSCL:PROF 'M1.ClkGen', TRI
[:SOURce]:SSCLocking:SHAPe[?]

Syntax  
[:SOURce]:SSCLocking:SHAPe 'identifier', <'Filename'>
[:SOURce]:SSCLocking:SHAPe? 'identifier'

Input Parameters

'identifier': 'M1.ClkGen'

<'Filename'>: Set up spread spectrum clocking shape specified in 'Filename'.

Return Range 'Filename'

Description  
This command sets up the spread spectrum clocking (SSC) using the specified shape in "Filename". A simple text file contains the data points which define the arbitrary waveform of the SSC profile. This SCPI is applicable for M8041A and M8045A.

Example  
:SSCL:SHAP 'M1.ClkGen','Factory/Sinusoidal.txt'

[:SOURce]:SSCLocking:FREQuency[?]

Syntax  
[:SOURce]:SSCLocking:FREQuency 'identifier', <NRf>
[:SOURce]:SSCLocking:FREQuency? 'identifier'

Input Parameters

'identifier': 'M1.ClkGen'

<NRf>: Set spread spectrum clocking frequency.

Return Range 100 Hz to 200 kHz

Description  
This command sets the spread spectrum clocking (SSC) frequency in hertz. Acceptable units include Hz, kHz and exponents (for example, 1E3 is the same as 1 kHz).

This SCPI is applicable for M8041A, M8195A, M8196A and M8045A.

Example  
:SSCL:FREQ 'M1.ClkGen', 1khz
This subnode has the following SCPI structure:

```
[:SOURce]
  :INTerference
    [:LEVEL]
      [:HFRequency[:SOURce]
      [:LFRequency[:SOURce]
      CMODe
        [:STATE][?]
        AMPLitude[?]
        SOURce[?]
        GAIN[?]
    DMODe
      [:STATE][?]
      AMPLitude[?]
      SOURce[?]
      GAIN[?]
    ISYMbol
      [:STATE][?]
      :MODE[?]
      FREQuency{1:2}[?]
      :ILOSs{1:2}[?]
      PRESet[?]
```
This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:LEV]l:HFRequency[SOURce]</td>
<td>[:SOURce]:INTERference[:LEV]:HFRequency[:SOURce] on page 195</td>
</tr>
<tr>
<td>[:LEV]l:LFRequency[SOURce]</td>
<td>[:SOURce]:INTERference[:LEV]:LFRequency[:SOURce] on page 195</td>
</tr>
<tr>
<td>[:LEV]l:CMODE[:STATe]</td>
<td>[:SOURce]:INTERference[:LEV]:CMODE[:STATe] on page 195</td>
</tr>
<tr>
<td>[:LEV]l:CMODE:AMPLitude</td>
<td>[:SOURce]:INTERference[:LEV]:CMODE:AMPLitude on page 196</td>
</tr>
<tr>
<td>[:LEV]l:CMODE:SOURce</td>
<td>[:SOURce]:INTERference[:LEV]:CMODE:SOURce on page 196</td>
</tr>
<tr>
<td>[:LEV]l:CMODE:GAIN</td>
<td>[:SOURce]:INTERference[:LEV]:CMODE:GAIN on page 197</td>
</tr>
<tr>
<td>[:LEV]l:DMODE[:STATe]</td>
<td>[:SOURce]:INTERference[:LEV]:DMODE[:STATe] on page 197</td>
</tr>
<tr>
<td>[:LEV]l:DMODE:AMPLitude</td>
<td>[:SOURce]:INTERference[:LEV]:DMODE:AMPLitude on page 197</td>
</tr>
<tr>
<td>[:LEV]l:DMODE:SOURce</td>
<td>[:SOURce]:INTERference[:LEV]:DMODE:SOURce on page 198</td>
</tr>
<tr>
<td>[:LEV]l:DMODE:GAIN</td>
<td>[:SOURce]:INTERference[:LEV]:DMODE:GAIN on page 198</td>
</tr>
<tr>
<td>:ISYMbol[:STATe]</td>
<td>[:SOURce]:INTERference:ISYMbol[:STATe] on page 199</td>
</tr>
<tr>
<td>:ISYMbol:MODE</td>
<td>[:SOURce]:INTERference:ISYMbol:MODE on page 199</td>
</tr>
<tr>
<td>:ISYMbol:FREQuency(1:2)</td>
<td>[:SOURce]:INTERference:ISYMbol:FREQuency(1:2) on page 199</td>
</tr>
<tr>
<td>:ISYMbol:ILOSs(1:2)</td>
<td>[:SOURce]:INTERference:ISYMbol:ILOSs(1:2) on page 200</td>
</tr>
<tr>
<td>:ISYMbol:PRESet</td>
<td>[:SOURce]:INTERference:ISYMbol:PRESet on page 200</td>
</tr>
</tbody>
</table>
[:SOURce]:INTERference[:LEVel]:HFRequency[:SOURce]?]

Syntax  [:SOURce]:INTERference[:LEVel]:HFRequency[:SOURce] 'identifier', <NRf>
        [:SOURce]:INTERference[:LEVel]:HFRequency[:SOURce]? 'identifier'

Input Parameters  'identifier': 'M*.DATAOUT1' or 'M*.DataOut2'
                  <NRf>: Set frequency of HF generator.

Return Range  1 GHz to 6 GHz

Description  This command sets the frequency of the high frequency generator. Acceptable units include MHz, GHz and exponents (for example, 2E9 is the same as 2 GHz).

This SCPI is applicable for M8041A and M8051A.

Example  :INT:HFR 'M2.DataOut1', 2.0e9

[:SOURce]:INTERference[:LEVel]:LFRequency[:SOURce]?]

Syntax  [:SOURce]:INTERference[:LEVel]:LFRequency[:SOURce] 'identifier', <NRf>
        [:SOURce]:INTERference[:LEVel]:LFRequency[:SOURce]? 'identifier'

Input Parameters  'identifier': 'M*.DATAOUT1' or 'M*.DataOut2'
                  <NRf>: Set frequency of LF generator.

Return Range  100 Hz to 1 GHz

Description  This command sets the frequency of the low frequency generator. Acceptable units include Hz, kHz, MHz, GHz and exponents (for example, 1E9 is the same as 1 GHz).

This SCPI is applicable for M8041A and M8051A.

Example  :INT:LFR 'M2.DataOut1', 1.0e9

[:SOURce]:INTERference[:LEVel]:CMODe[:STATe]?]

Syntax  [:SOURce]:INTERference[:LEVel]:CMODe[:STATe] 'identifier', <0|1|ON|OFF>
        [:SOURce]:INTERference[:LEVel]:CMODe[:STATe]? 'identifier'

Input Parameters  'identifier': 'M*.DATAOUT1', 'M*.DataOut2', or 'M*.DataOut'
                  <0|1|ON|OFF>: Turn common mode interference on/off.

Return Range  0|1
Description  This command turns common mode interference on/off. This SCPI is applicable for M8041A, M8051A, M8061A and M8062A.

Example  \[:\text{INTERFERENCE} \text{[:LEVEL]} \text{[:CMODE]} \text{[:AMPLITUDE]} \text{[?]}\]

Syntax \[:\text{SOURCE} \text{[:INTERFERENCE} \text{[:LEVEL]} \text{[:CMODE]} \text{[:AMPLITUDE]} \text{[?]}\]

Input Parameters

\text{'identifier': 'M*.DATAOUT1' or 'M*.DataOut2'}

Return Range 0 mV to 320 mV

Description  This command sets the amplitude of the common mode interference in volts. Acceptable units include mV, V and exponents (for example, 100E-3 is the same as 100 mV).

This SCPI is applicable for M8041A and M8051A.

Example  \[:\text{INTERFERENCE} \text{[:LEVEL]} \text{[:CMODE]} \text{[:AMPLITUDE]} \text{[?]}\]

Syntax \[:\text{SOURCE} \text{[:INTERFERENCE} \text{[:LEVEL]} \text{[:CMODE]} \text{[:SOURCE]} \text{[?]}\]

Input Parameters

\text{'identifier': 'M*.DATAOUT1' or 'M*.DataOut2'}

\text{'HFREQUENCY'|'LFREQUENCY'}: Select HF or LF generator.

Return Range \text{HFR}|\text{LFR}

Description  This command selects the high frequency or low frequency generator type for common mode interference.

This SCPI is applicable for M8041A and M8051A.

Example  \[:\text{INTERFERENCE} \text{[:LEVEL]} \text{[:CMODE]} \text{[:AMPLITUDE]} \text{[?]}\]

Syntax \[:\text{SOURCE} \text{[:INTERFERENCE} \text{[:LEVEL]} \text{[:CMODE]} \text{[:SOURCE]} \text{[?]}\]

Input Parameters

\text{'identifier': 'M*.DATAOUT1' or 'M*.DataOut2'}

\text{'HFREQUENCY'|'LFREQUENCY'}: Select HF or LF generator.

Return Range \text{HFR}|\text{LFR}

Description  This command selects the high frequency or low frequency generator type for common mode interference.

This SCPI is applicable for M8041A and M8051A.
[:SOURce]:INTERference[:LEVel]:CMODe:GAIN[?]

Syntax
[:SOURce]:INTERference[:LEVel]:CMODe:GAIN 'identifier', <NRf>
[:SOURce]:INTERference[:LEVel]:CMODe:GAIN? 'identifier'

Input
'identifier': 'M*.DATAOUT1' or 'M*.DataOut2'

Parameters
<NRF>: Set common mode interference gain.

Return Range
0.1 to 1.0

Description
This command sets the linear gain of the common mode interference.
This command is only for M8061A and M8062A.

Example
:INT:CMOD:GAIN 'M2.DataOut1', 0.5

[:SOURce]:INTERference[:LEVel]:DMODe[:STAtE][?]

Syntax
[:SOURce]:INTERference[:LEVel]:DMODe[:STAtE] 'identifier', <ON|OFF>
[:SOURce]:INTERference[:LEVel]:DMODe[:STAtE]? 'identifier'

Input
'identifier': 'M*.DATAOUT', 'M*.DATAOUT1', or 'M*.DataOut2'

Parameters
<ON|OFF>: Turn differential mode interference on/off.

Return Range
ON|OFF

Description
This command turns differential mode interference on/off.
This SCPI is applicable for M8041A, M8051A, M8061A and M8062A.

Example
:INT:DMOD 'M2.DataOut1', on

[:SOURce]:INTERference[:LEVel]:DMODe:AMPLitude[?]

Syntax
[:SOURce]:INTERference[:LEVel]:DMODe:AMPLitude 'identifier', <NRf>
[:SOURce]:INTERference[:LEVel]:DMODe:AMPLitude? 'identifier'

Input
'identifier': 'M*.DATAOUT1' or 'M*.DataOut2'

Parameters
<NRF>: Set differential mode interference amplitude.

Return Range
0 mV to 360 mV
Description  This command sets the amplitude of the differential mode interference in volts. Acceptable units include mV, V and and exponents (for example, 100E-3 is the same as 100 mV).

This SCPI is applicable for M8041A and M8051A.

Example  :INT:DMOD:AMPL 'M2.DataOut1', 100mV


Syntax  [:SOURce]:INTerference[:LEVel]:DMODE:SOURce 'identifier', <HFRequency|LFRequency>

[:SOURce]:INTerference[:LEVel]:DMODE:SOURce? 'identifier'

Input  'identifier': 'M*.DATAOUT1' or 'M*.DataOut2'

Parameters  <HFRequency|LFRequency>: Select HF or LF generator.

Return Range  HFR|LFR

Description  This command selects the high frequency or low frequency generator type for differential mode interference.

This SCPI is applicable for M8041A and M8051A.

Example  :INT:DMOD:SOUR 'M2.DataOut1', LFR


Syntax  [:SOURce]:INTerference[:LEVel]:DMODE:GAIN 'identifier', <NRf>

[:SOURce]:INTerference[:LEVel]:DMODE:GAIN? 'identifier'

Input  'identifier': 'M*.DATAOUT1' or 'M*.DataOut2'

Parameters  <NRf>: Set differential mode interference gain.

Return Range  0.01 to 1.0

Description  This command sets the linear gain of the differential mode interference.

This SCPI is applicable for M8061A and M8062A.

Example  :INT:DMOD:GAIN 'M2.DataOut1', 0.5
**[:SOURce]:INterference:ISYMbol[:STATE]?**

**Syntax**

[:SOURce]:INterference:ISYMbol[:STATE] 'Identifier', <0|1|ON|OFF>

[:SOURce]:INterference:ISYMbol[:STATE]? 'Identifier'

**Input Parameters**

'Identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

**Return Range**

0|1

**Description**

This command enables/disables the inter symbol interference state.

This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**

:SOUR:INT:ISYM 'M2.DataOut1', ON

---

**[:SOURce]:INterference:ISYMbol:MODE[?]**

**Syntax**

[:SOURce]:INterference:ISYMbol:MODE 'Identifier', <PONE | PTWO>

[:SOURce]:INterference:ISYMbol:MODE? 'Identifier'

**Input Parameters**

'Identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

**Description**

This command allows the user to adjust inter symbol interface trace via one (no offset) or two point (variable offset).

Inter symbol interference mode must be set to one or two points; then only, the value of frequency and insertion loss parameter can be changed.

This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**

:SOUR:INT:ISYM:MODE 'M2.DataOut1', PTWO

---

**[:SOURce]:INterference:ISYMbol:FREQuency1:2[?]**

**Syntax**

[:SOURce]:INterference:ISYMbol:FREQuency1 'Identifier', <NRf>

[:SOURce]:INterference:ISYMbol:FREQuency2 'Identifier', <NRf>

[:SOURce]:INterference:ISYMbol:FREQuency1? 'Identifier'

[:SOURce]:INterference:ISYMbol:FREQuency2? 'Identifier'

**Input Parameters**

 Identifier: 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

**Description**

This command sets one or two discrete user definable frequencies f1 and f2 of the inter symbol interference (ISI) trace. The mode has to be changed...
to two points in order to use the second frequency.
This SCPI is applicable for M8041A, M8051A, and M8062A.

Range 1GHz to 16GHz.

Example :SOUR:INT:ISYM:FREQ1 'M2.DataOut1', 8.0e9

[:SOURce]:INTerference:ISYMbol:ILOSs(1:2)?
Syntax [:SOURce]:INTerference:ISYMbol:ILOSs1 'Identifier', <NRf>
[:SOURce]:INTerference:ISYMbol:ILOSs2 'Identifier', <NRf>
[:SOURce]:INTerference:ISYMbol:ILOSs1? 'Identifier'
[:SOURce]:INTerference:ISYMbol:ILOSs2? 'Identifier'

Input Parameters Identifier: 'M*.DataOut', 'M*.DataOut1', or 'M*.DataOut2'

Description This command sets the insertion loss at one or two discrete user definable frequencies, f1 and f2 of the inter symbol interference (ISI) trace.
This SCPI is applicable for M8041A, M8051A, or M8062A.

Range 0.0dB to -25.0dB.

Example :SOUR:INT:ISYM:ILOS1 'M2.DataOut', -9.8

[:SOURce]:INTerference:ISYMbol:PRESet?

Input Parameters 'Identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

Description Selects an inter symbol interference (ISI) preset for the specific application.
This command should ideally not be sent in one PM (SCPI - Programm Message) with following commands...:MODE; ...:FREQuency; ....:ILOSs.
This SCPI is applicable for M8041A, M8051A, and M8062A.
For M8062A the presets are available up to P11 only; please refer to the Table 48 on page -202.
The following table describes the list of presets for selected standards and traces.

Table 47  Preset List for Selected Standards and Traces (For M8041A and M8051A)

<table>
<thead>
<tr>
<th>SCPI Name</th>
<th>f1/GHz</th>
<th>L1/dB</th>
<th>f2/GHz</th>
<th>L2/dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCIe3</td>
<td>P1</td>
<td>1</td>
<td>-3.0</td>
<td>4</td>
</tr>
<tr>
<td>PCIe3</td>
<td>P2</td>
<td>1</td>
<td>-5.6</td>
<td>4</td>
</tr>
<tr>
<td>M-PHY G3A</td>
<td>Ch1</td>
<td>P3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>M-PHY G3A</td>
<td>Ch2</td>
<td>P4</td>
<td>1.248</td>
<td>-4.1</td>
</tr>
<tr>
<td>M-PHY G3B</td>
<td>Ch1</td>
<td>P5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>M-PHY G3B</td>
<td>Ch2</td>
<td>P6</td>
<td>1.458</td>
<td>-4.2</td>
</tr>
<tr>
<td>M8048A</td>
<td>7.7&quot;</td>
<td>P7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>M8048A</td>
<td>9.4&quot;</td>
<td>P8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>M8048A</td>
<td>11.1&quot;</td>
<td>P9</td>
<td>1</td>
<td>-1.9</td>
</tr>
<tr>
<td>M8048A</td>
<td>12.8&quot;</td>
<td>P10</td>
<td>1</td>
<td>-2.3</td>
</tr>
<tr>
<td>M8048A</td>
<td>14.4&quot;</td>
<td>P11</td>
<td>1</td>
<td>-2.7</td>
</tr>
<tr>
<td>M8048A</td>
<td>16.1&quot;</td>
<td>P12</td>
<td>1</td>
<td>-3.0</td>
</tr>
<tr>
<td>M8048A</td>
<td>24.4</td>
<td>P13</td>
<td>1</td>
<td>-4.5</td>
</tr>
<tr>
<td>SAS-3 12 Gb/s</td>
<td>P14</td>
<td>1</td>
<td>-4.5</td>
<td>4</td>
</tr>
<tr>
<td>MIPI-Short</td>
<td>P15</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>MIPI-Standard</td>
<td>P16</td>
<td>1.25</td>
<td>-3.75</td>
<td>5</td>
</tr>
<tr>
<td>MIPI-Long</td>
<td>P17</td>
<td>1.25</td>
<td>-6.3</td>
<td>5</td>
</tr>
</tbody>
</table>

Example: :SOUR:INT:ISYM:PRES 'M2.DataOut1', P1
### Table 48  Preset List for Selected Standards and Traces (For M8062A)

<table>
<thead>
<tr>
<th>SCPI Name</th>
<th>SCPI Name</th>
<th>f1/GHz</th>
<th>L1/dB</th>
<th>f2/GHz</th>
<th>L2/dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>CEI-28G-VSR Min</td>
<td>P1</td>
<td>14</td>
<td>-3.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CEI-28G-VSR Max</td>
<td>P2</td>
<td>14</td>
<td>-9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CEI-28G-VSR 100GbE</td>
<td>P3</td>
<td>12.89</td>
<td>-10.25</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CEI-28G-VSR 32GFC</td>
<td>P4</td>
<td>14.025</td>
<td>-10.25</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>M8048A 7.7&quot;</td>
<td>P5</td>
<td>4</td>
<td>-4.4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>M8048A 9.4&quot;</td>
<td>P6</td>
<td>4</td>
<td>-6</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>M8048A 11.1&quot;</td>
<td>P7</td>
<td>1</td>
<td>-1.9</td>
<td>4</td>
<td>-6.7</td>
</tr>
<tr>
<td>M8048A 12.8&quot;</td>
<td>P8</td>
<td>1</td>
<td>2.3</td>
<td>4</td>
<td>-7.1</td>
</tr>
<tr>
<td>M8048A 14.4&quot;</td>
<td>P9</td>
<td>1</td>
<td>-2.7</td>
<td>4</td>
<td>-8.1</td>
</tr>
<tr>
<td>M8048A 16.1&quot;</td>
<td>P10</td>
<td>1</td>
<td>-3</td>
<td>4</td>
<td>-9</td>
</tr>
<tr>
<td>M8048A 24.4&quot;</td>
<td>P11</td>
<td>1</td>
<td>-4.5</td>
<td>4</td>
<td>-13.8</td>
</tr>
</tbody>
</table>
This subnode has the following SCPI structure:

```
[:SOURce]
  ::INTerference
    ::RANDom
      ::HFRequency
      ::LFRequency
      ::CREStfactor
      ::UNIT
```

This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:SOURce]:INTerference:RANDom:HFRequency</td>
<td>[:SOURce]:INTerference:RANDom:HFRequency on page 204</td>
</tr>
<tr>
<td>[:SOURce]:INTerference:RANDom:LFRequency</td>
<td>[:SOURce]:INTerference:RANDom:LFRequency on page 204</td>
</tr>
<tr>
<td>[:SOURce]:INTerference:RANDom:CREStfactor</td>
<td>[:SOURce]:INTerference:RANDom:CREStfactor on page 204</td>
</tr>
<tr>
<td>[:SOURce]:INTerference:RANDom:CREStfactor:UNIT</td>
<td>[:SOURce]:INTerference:RANDom:CREStfactor:UNIT on page 205</td>
</tr>
</tbody>
</table>
[:SOURce]:INTERference:RANDom:HFRequency

Syntax [:SOURce]:INTERference:RANDom:HFRequency 'identifier', <NRf>
[:SOURce]:INTERference:RANDom:HFRequency? 'identifier'

Input
'identifier': 'M*.DATAOUT1' or 'M*.DataOut2'

Parameters
<NRF>: Set random interference high frequency

Description
This command sets the random interference high frequency in Hz.
This SCPI is applicable for M8195A and M8196A.

Example
:INT:RAND:HFR 'M2.DataOut1', 1.0e9

[:SOURce]:INTERference:RANDom:LFRequency

Syntax [:SOURce]:INTERference:RANDom:LFRequency 'identifier', <NRf>
[:SOURce]:INTERference:RANDom:LFRequency? 'identifier'

Input
'identifier': 'M*.DATAOUT1' or 'M*.DataOut2'

Parameters
<NRF>: Set random interference low frequency

Description
This command sets the random interference low frequency in Hz.
This SCPI is applicable for M8195A and M8196A.

Example
:INT:RAND:LFR 'M2.DataOut1', 1.0e9

[:SOURce]:INTERference:RANDom:CREStfactor

Syntax [:SOURce]:INTERference:RANDom:CREStfactor 'identifier', <NRf>
[:SOURce]:INTERference:RANDom:CREStfactor? 'identifier'

Input
'identifier': 'M*.DATAOUT1' or 'M*.DataOut2'

Parameters
<NRF>: Set random interference crest factor value

Description
This command sets the random interference crest factor value in dB or as voltage ratio depending on the currently selected crest factor unit type.
This SCPI is applicable for M8195A and M8196A.

Example
:INT:RAND:CRES 'M2.DataOut1', 12
[:SOURce]:INTerference:RANDom:CREStfactor:UNIT[?]

Syntax
[:SOURce]:INTerference:RANDom:CREStfactor:UNIT[?] 'identifier',
< LINear | LOGarithmic >

[:SOURce]:INTerference:RANDom:CREStfactor:UNIT? 'identifier'

Input
'identifier': 'M*.DATAOUT1' or 'M*.DataOut2'

Parameters
< LINear | LOGarithmic >

Description
This command sets the random interference crest factor unit type. Following unit types are allowed:

- LINear – This is the default setting. Select this unit to specify the crest factor as voltage ratio.
- LOGarithmic – Select this unit to specify the crest factor in dB.

This SCPI is applicable for M8195A and M8196A.

Example
:INT:RAND:CRES:UNIT 'M2.DataOut1', LOG
[:SOURce]:INTerference:SINusoidal Subnode

This subnode controls the configuration when AWG is used as sinusoidal interference source. It has the following SCPI structure:

```
[:SOURce]
  :INTerference
    :SINusoidal
      :FRQuency[?]
      :AMPLitude{1:2}[?]
      :PHASE{1:2}[?]
      :TFRequency{1:2}[?]
      :TONemode{1:2}[?]
```

This subnode has the following commands:

Table 50

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:INTerference:SINusoidal:FREQuency[?]</td>
<td>[SOURce]:INTerference:SINusoidal:FREQuency[?] on page 207</td>
</tr>
<tr>
<td>:INTerference:SINusoidal:AMPLitude{1:2}[?]</td>
<td>[SOURce]:INTerference:SINusoidal:AMPLitude{1:2}[?] on page 207</td>
</tr>
<tr>
<td>:INTerference:SINusoidal:PHASE{1:2}[?]</td>
<td>[SOURce]:INTerference:SINusoidal:PHASE{1:2}[?] on page 207</td>
</tr>
<tr>
<td>:INTerference:SINusoidal:TFRequency{1:2}[?]</td>
<td>[SOURce]:INTerference:SINusoidal:TFRequency{1:2}[?] on page 208</td>
</tr>
<tr>
<td>:INTerference:SINusoidal:TONemode{1:2}[?]</td>
<td>[SOURce]:INTerference:SINusoidal:TONemode{1:2}[?] on page 208</td>
</tr>
</tbody>
</table>
[:SOURce]:INTERference:SINusoidal:FREQuency[?]

Syntax
[:SOURce]:INTERference:SINusoidal:FREQuency 'identifier', <NRf>
[:SOURce]:INTERference:SINusoidal:FREQuency? 'identifier'

Input
'identifier': 'M*.DataOut1' or 'M*.DataOut2'

Parameters

Description
This command sets the sinusoidal interference frequency in Hz.
This SCPI is applicable for M8195A and M8196A.

Example
:SINT:SIN:FREQ 'M2.DataOut1',1e9

[:SOURce]:INTERference:SINusoidal:AMPLitude{1:2}[?]

Syntax
[:SOURce]:INTERference:SINusoidal:AMPLitude{1:2} 'identifier', <NRf>
[:SOURce]:INTERference:SINusoidal:AMPLitude{1:2}? 'identifier'

Input
'identifier': 'M*.DataOut1' or 'M*.DataOut2' or 'M*.DataOut3' or 'M*.DataOut4'

Parameters

Description
This command can be used to set the Multitone Amplitude 1 and Multitone Amplitude 2 parameters value.
This SCPI is applicable for M8195A and M8196A.

Example
The following example shows how to use this command to set the Multitone Amplitude 1 parameter value:
:SOUR:INT:SIN:AMPL1 'M1.DataOut1',0.2
0.2

[:SOURce]:INTERference:SINusoidal:PHASe{1:2}[?]

Syntax
[:SOURce]:INTERference:SINusoidal:PHASe{1:2} 'identifier', <NRf>
[:SOURce]:INTERference:SINusoidal:PHASe{1:2}? 'identifier'

Input
'identifier': 'M*.DataOut1' or 'M*.DataOut2' or 'M*.DataOut3' or 'M*.DataOut4'

Parameters

Description
This command can be used to set the Multitone Phase 1 and Phase 2 parameters value.
This SCPI is applicable for M8195A and M8196A.

**Example**
The following example shows how to use this command to set the Multitone Phase 1 parameter value:

```
:SOUR:INT:SIN:PHAS1 'M1.DataOut1', 180
```

180

```[:SOURce]:INTerference:SINusoidal:TFRequency{1:2}[?]
```

**Syntax**
`[:SOURce]:INTerference:SINusoidal:TFRequency{1:2} 'Identifier', <NRf>`

`[:SOURce]:INTerference:SINusoidal:TFRequency{1:2}? 'identifier'`

**Input Parameters**
- `'identifier': 'M*.DataOut1' or 'M*.DataOut2' or 'M*.DataOut3' or 'M*.DataOut4'`

**Description**
This command can be used to set the Multitone Frequency 1 and Multitone Frequency 2 parameters value.

This SCPI is applicable for M8195A and M8196A.

**Example**
The following example shows how to use this command to set the Multitone Frequency 1 parameter value:

```
:SOUR:INT:SIN:TFR1 'M1.DataOut1', 100000000
```

100000000

```[:SOURce]:INTerference:SINusoidal:TONemode{1:2}[?]
```

**Syntax**
`[:SOURce]:INTerference:SINusoidal:TONemode{1:2} 'Identifier', OFF | COMMon | DIFFerential`

`[:SOURce]:INTerference:SINusoidal:TONemode{1:2}? 'identifier'`

**Input Parameters**
- `'identifier': 'M*.DataOut1' or 'M*.DataOut2' or 'M*.DataOut3' or 'M*.DataOut4'`

**Description**
This command can be used to set the Tone Mode 1 and Tone Mode 2 to either Common mode or Differential mode.

This SCPI is applicable for M8195A and M8196A.
Example  The following example shows how to use this command to set the Tone Mode 1 to Differential Mode:

:SOUR:INT:SIN:TON1 'M1.DataOut1',DIFFerential


DIFFerential
This subnode has the following SCPI structure:

```
[:SOURce]
 :CONFigure
  :MINTegration[?]
```

This subnode has the following command:

```
[:SOURce]:CONFigure:MINTegration[?]
```

**Syntax**
```
[:SOURce]:CONFigure:MINTegration
  'identifier',<NONe|MUX|DMUX|BOTH>[<data-location>[<visa-resource>]]
```

**Input Parameters**
- `identifier`: 'M*.MuxMode'
- `<NONe|MUX|DMUX|BOTH>`: Select mux/demux module.
- `<data-location>`: (Optional) Specify location of data module. If omitted or an empty string is specified, the first suitable module is chosen automatically.
- `<visa-resource>`: (Optional) Specify the address string of the N4877A.

**Return Range**
- NONe|MUX|DMUX|BOTH

**Description**
This command controls the integration of a multiplexer and/or demultiplexer to a data module. Also, it controls the integration of the N4877A Clock Data Recovery and Demultiplexer.

In M8062A, the parameter values 'MUX', 'DMUX' and 'BOTH' all perform the same action i.e. they enable the 32G mode. 'NONe' disables 32G mode.

This SCPI is applicable for M8061A and M8062A.

**Example**
```
:CONF:MINT 'M2.MuxMode', BOTH
```
OUTPut Subsystem

The OUTPut subsystem controls the output ports of the pattern generator. This subsystem has the following SCPI structure:

```
:OUTPut
  :GLOBal
  [:STATe]?,
  [:STATe]?
  :CONFig
  [:SOURce]?
  :DRATe?
  :POLarity?
  :TVOLTage?
  :TCONfig[?]
  :COUPLing[?]
  :DIVider[?]
  :FREQuency
  [:DIVider]?
  :CALibration
  [:SKEW]?
  :EIDLe
  [:STATe]?
  :DATA
  [:XOVer]?
  [:F2Jitter]?
  [:STATe]?
  :DEEMphasis
  [:...]
  :EINSertion
  [:...]
  :TRIGger
  [:...]
  :DEMBedding
  [:...]
  :EMBEDding
  [:...]
```
This subsystem has the following commands and subnodes:

Table 51

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>:GLOBal[:STATe]?</td>
<td>:OUTPut:GLOBal[:STATe]? on page 213</td>
<td></td>
</tr>
<tr>
<td>[:STATe]?</td>
<td>:OUTPut[:STATe]? on page 213</td>
<td></td>
</tr>
<tr>
<td>:CONFig:SOURce[?]</td>
<td>:OUTPut:CONFig:SOURce[?] on page 213</td>
<td></td>
</tr>
<tr>
<td>:POLarity[?]</td>
<td>:OUTPut:POLarity[?] on page 214</td>
<td></td>
</tr>
<tr>
<td>:TVOltag[?]</td>
<td>:OUTPut:TVOltag[?] on page 214</td>
<td></td>
</tr>
<tr>
<td>:TCONfig[?]</td>
<td>:OUTPut:TCONfig[?] on page 215</td>
<td></td>
</tr>
<tr>
<td>:COUPling[?]</td>
<td>:OUTPut:COUPling[?] on page 215</td>
<td></td>
</tr>
<tr>
<td>:DIVider[?]</td>
<td>:OUTPut:DIVider[?] on page 216</td>
<td></td>
</tr>
<tr>
<td>:FREQuency[?]</td>
<td>:OUTPut:FREQuency[?] on page 216</td>
<td></td>
</tr>
<tr>
<td>:CALibration:SKEW[?]</td>
<td>:OUTPut:CALibration:SKEW[?] on page 217</td>
<td></td>
</tr>
<tr>
<td>:EIDLe[:STATe]?</td>
<td>:OUTPut:EIDLe[:STATe]? on page 217</td>
<td></td>
</tr>
<tr>
<td>:DATA:XOVer[?]</td>
<td>:OUTPut:DATA:XOVer[?] on page 218</td>
<td></td>
</tr>
</tbody>
</table>

Subnodes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>:DEEMphasis</td>
<td>:OUTPut:DEEMphasis Subnode on page 220</td>
<td></td>
</tr>
<tr>
<td>:EINSertion</td>
<td>:OUTPut:EINSertion Subnode on page 228</td>
<td></td>
</tr>
<tr>
<td>:TRIGger</td>
<td>:OUTPut:TRIGger Subnode on page 232</td>
<td></td>
</tr>
<tr>
<td>:DEMBedding</td>
<td>:OUTPut:DEMBedding Subnode on page 234</td>
<td></td>
</tr>
<tr>
<td>:EMBedding</td>
<td>:OUTPut:EMBedding Subnode on page 237</td>
<td></td>
</tr>
</tbody>
</table>
:OUTPut:GLOBal[:STATe][?]
Syntax   :OUTPut:GLOBal[:STATe] 'identifier', <ON|OFF|1|0>
         :OUTPut:GLOBal[:STATe]? 'identifier'
Input    'identifier': 'M1.System'
Parameters <ON|OFF|1|0>: Enable/disable global state.
Return Range 1|0
Description This command sets global output state to ON or OFF.
This command works for all modules, as it is a Global command.
Example   :OUTP:GLOB 'M1.System', off

:OUTPut[:STATe][?]
Syntax   :OUTPut[:STATe] 'identifier', <ON|OFF|1|0>
         :OUTPut[:STATe]? 'identifier'
         'M*.DataOut1', 'M*.DataOut2', 'M*.CtrlOut', 'M*.CtrlOutA', or 'M*.DataOut'
Parameters <ON|OFF|1|0>: Switch an output on or off.
Return Range 1|0
Description This command switches an output on or off.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8195A
and M8045A.
Example   :OUTP 'M1.DataOut1', off

:OUTPut:CONFig:SOURce[?]
Syntax   :OUTPut:CONFig:SOURce <Identifier>, <CLEanclock|DATaclock>
         :OUTPut:CONFig:SOURce? <Identifier>
Input    'identifier': 'M1.ClkOut1' or 'M1.ClkOut2'
Parameters
Return Range  CLE|DAT
Description This command selects the signal source for the instrument’s output. This is only valid for certain clock outputs.
This query returns the present state.
This SCPI is supported by M8045A.
Example :OUTP:CONF:SOUR 'M1.ClkOut1', CLE

:OUTPut:DRATe?
Syntax :OUTPut:DRATe? 'identifier'
Input Parameters 'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut', or 'M1.ClkOut'
Description This query returns data rate for the output channel.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8195A and M8045A.
Example :OUTP:DRAT? 'M1.DataOut1'

:OUTPut:POLarity[?]
Syntax :OUTPut:POLarity 'identifier', <NORMal|INVerted>
:OUTPut:POLarity? 'identifier'
<NORMal|INVerted>: Set output polarity to normal or inverted.
Return Range NORM|INV
Description This command sets the output polarity to either normal or inverted.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.
Example :OUTP:POL 'M1.DataOut1', INV

:OUTPut:TVOLTage[?]
Syntax :OUTPut:TVOLTage 'identifier', <NRf>
:OUTPut:TVOLTage? 'identifier'

:OUTPut:TVOL 'identifier', <NRf>

Input Parameters
'identifier': 'M1.ClkOut', 'M1.TrigOut', 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

Return Range <NRf>

Description
This command specifies the external termination voltage. The default is 0 V (ground). Acceptable units include mV, V and exponents (for example, 700E-3 is the same as 700 mV).

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8195A.

Example
:OUTP:TVOL 'M1.DataOut1', 700mV

:OUTPut:TCONfig['?']

Syntax
:OUTPut:TCONfig 'identifier', <BALanced|UNBalanced>

Input Parameters
'identifier': 'M1.ClkOut', 'M1.TrigOut', 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

Return Range BAL|UNB

Description
This command selects the termination model for an output frontend:

BALanced - balanced or differential output.

UNBalanced - unbalanced with respect to ground (termination voltage).

This SCPI is applicable for M8041A, M8051A, M8061A and M8062A.

Example
:OUTP:TCON 'M2.DataOut2', BAL

:OUTPut:COUPling['?']

Syntax
:OUTPut:COUPling 'identifier', <AC|DC>

Input Parameters
'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

Return Range AC|DC

Description
Select output coupling.
Description: This command selects DC or AC output coupling. 
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.

Example: :OUTP:COUP 'M1.DataOut1', DC

:OUTPut:DIVider[?]

Syntax: :OUTPut:DIVider 'identifier', <DIV1...DIV640>
:OUTPut:DIVider? 'identifier'

Input Parameters:

- Specify a divider: DIV1 | DIV2 | DIV4 | DIV8 | DIV10 | DIV16 | DIV20 | DIV24 | DIV30 | DIV32 | DIV40 | DIV48 | DIV50 | DIV60 | DIV64 | DIV66 | DIV80 | DIV96 | DIV100 | DIV120 | DIV128 | DIV132 | DIV160 | DIV192 | DIV200 | DIV240 | DIV256 | DIV264 | DIV320 | DIV400 | DIV512 | DIV528 | DIV640

Return Range: DIV1 to DIV640

Description: This command selects a specific clock divider from DIV1 through DIV640.
Please note that the divider availability depends on identifier, product number and currently used symbol rate.
This SCPI is applicable for M8041A and M8045A.

Example: :OUTP:DIV 'M1.ClkOut', DIV1

:OUTPut:FREQuency[?]

Syntax: :OUTPut:FREQuency 'identifier', <NRf>
:OUTPut:FREQuency? 'identifier'

Input Parameters:

- 'identifier': 'M1.RefClkOut' or 'M2.ClnClkOut'

Return Range: 1e+7 | 1e+8

Description: This command can be used to fetch or set the frequency at RefClockOut.
This SCPI is applicable for M8041A and M8045A. For M8062A, it is only applicable as a query.

Example: :OUTPut:FREQuency 'm1.refclkout', 1e+8.
:OUTPut:FREQuency:DIVider[?]
Syntax :OUTPut:FREQuency:DIVider 'identifier', <NRf>
:OUTPut:FREQuency:DIVider? 'identifier'
Input 'identifier': 'M*.ClnClkOut'
Return Range <NRf>
Description This command divides the frequency at clean clock out (Cln Clk Out) on an M8062A module. This SCPI is applicable for M8062A.
Example :OUTPut:FREQuency:DIVider 'M2.ClnClkOut', 2

:OUTPut:CALibration:SKEW[?]
Syntax :OUTPut:CALibration:SKEW 'identifier', <NRf>
:OUTPut:CALibration:SKEW? 'identifier'
Input 'identifier': 'M*.DataOut1' or 'M*.DataOut2'
Return Range -2ns to 2ns
Description This command is used as offset to Data Delay and Jitter Delay. This SCPI is applicable for M8041A, M8051A and M8045A.
Example :OUTPut:CALibration:SKEW 'M1.DataOut1', 0

:OUTPut:EIDLe[:STATe][?]
Syntax :OUTPut:EIDLe[:STATe] 'identifier', <OFF|IDLE|EXTernal>
:OUTPut:EIDLe[:STATe]? 'identifier'
Input 'identifier': 'M*.DataOut'
Parameters <OFF>: Electrical idle feature is disabled.
<IDLE>: Electrical idle feature is enabled (software controlled, power saving mode).
<EXTernal>: Electrical idle input can be used to provide an electrical idle signal at the Data Output. An external generated signal, for example by a pulse generator at the electrical idle input, controls the output stream at the Data Output.
Return Range  OFF|IDLE|EXT
Description  This command enables/disables the electrical idle feature for the Data Output. The state can be OFF, IDLE or EXTernal.
This SCPI is applicable for M8061A and M8062A.
Example  :OUTP:EIDL 'M2.DataOut', EXT

:OUTPut:DATA:XOVer[?]
Syntax  :OUTPut:DATA:XOVer 'identifier', <NRf>
:OUTPut:DATA:XOVer? 'identifier'
Input Parameters
'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'
<NRF>: Specify an eye crossover.
Return Range  For M8041A and M8051A: 10% to 90%
For M8061A and M8062A: 30% to 70%
Description  This command sets the eye crossover of the transmitter signal at the DATA OUT/-DATA OUT connectors. Crossover is defined for NRZ signals.
This SCPI is applicable for M8041A, M8051A, M8061A and M8062A.

:OUTPut:DATA:F2Jitter[?]
Syntax  :OUTPut:DATA:F2Jitter 'identifier', <NRf>
:OUTPut:DATA:F2Jitter? 'identifier'
Input Parameters
'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'
<NRF>: Set the f/2 jitter.
Return Range  -20 ps to +20 ps
Description  This command sets the f/2 jitter at the output in seconds. Acceptable units include ps and exponents (for example, -10e-12 is the same as -10 ps).
This query returns the present setting.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.
Example  :OUTP:DATA:F2J 'M1.DataOut2', -10e-12
**:OUTPut:DATA:F2Jitter:STATe? [ ]

Syntax :OUTPut:DATA:F2Jitter:STATe 'identifier', <ON|OFF>|0>

 :OUTPut:DATA:F2Jitter:STATe? 'identifier'

Input 'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

Parameters Return Range ON|OFF|1|0

Description This command enables/disables f/2 jitter at the output. Enabling this command, causes stop of full clock group (all pattern generator channels and CLK/TRIG).

This query returns the present setting.

This SCPI is applicable for M8045A.

Example :OUTP:DATA:F2J:STAT 'M1.DataOut', ON
The M8070A provides an eight tap de-emphasis signal. Seven of these taps can be specified: two pre-cursors and five post-cursors. The resulting main cursor is calculated.

The taps can be programmed remotely in two different ways. Firstly, all the tap values can be set by one command. In this case all seven tap values have to be specified at once in a comma separated list of numbers. Total 31 preset registers are available. Secondly, a preset register can be determined as the current preset and the pre-and post-cursor commands work on this selected preset register. Only absolute values are accepted in both cases.

The sign of a tap value is defined by a separate command. If, for example, the sign of pre-cursor 2 is set to negative (-1), it will change the sign of every pre-cursor 2 in the 31 preset registers.

This subnode has the following SCPI structure:

```
:OUTPut
  :DEEMphasis
    :UNIT[?]
    :PRESet[?]
    :SELect
      :PRESet[?]
      :ENABle[?]
    :PRECursor{1:2}[?]
    :POSTcursor{1:5}[?]
    :CONFigure
      :SIGN[?]
    :CURSor
      :MAGNitude(0|1|2|3|4|5|6|7)[?]
      :MAIN
        :AUTO[?]
        :SUFFix?
```
This subnode has the following commands:

Table S2

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:PRECursor{1:2}?</td>
<td>:OUTPut:DEEMphasis:PRECursor{1:2}? on page 224</td>
</tr>
<tr>
<td>:POSTcursor{1:5}?</td>
<td>:OUTPut:DEEMphasis:POSTcursor{1:5}? on page 224</td>
</tr>
<tr>
<td>:CONFigure:SIGN?</td>
<td>:OUTPut:DEEMphasis:CONFigure:SIGN? on page 225</td>
</tr>
<tr>
<td>:CURSor:MAGNitude(0</td>
<td>1</td>
</tr>
</tbody>
</table>

:OUTPut:DEEMphasis:UNIT?

Syntax: :OUTPut:DEEMphasis:UNIT 'identifier', <DB|PCT>

:OUTPut:DEEMphasis:UNIT? 'identifier'

Parameters:
- 'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'
- <DB|PCT>: Select dB or percent as the unit.

Return Range: DB|PCT

Description: This command defines the tap value unit: dB or PCT (PerCenT). This command should not be sent in one PM (SCPI - Program Message) with the following de-emphasis commands:
In other words, the unit should be set before sending the values with the above de-emphasis commands.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.


`:OUTPut:DEEMphasis:PRESet[?]`

Syntax: :OUTPut:DEEMphasis:PRESet 'identifier',<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>


Input Parameters

'identifier': 'M*.DataOut1' or 'M*.DataOut2'

<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>: Set register index and tap values.

<Preset Register No.>: Specify preset register number.

Return Range <Pre2>,<Pre1>,<Post1>,<Post2>,<Post3>,<Post4>, <Post5>

Description: This command is used to enter a comma separated list representing a register index and tap values. The first number in the list specifies the index of the preset register. There are 31 preset registers available addressed by a register index of 0 up to 30. The values in the list specify the register index and taps in the following order:

<RegNr>,<Pre2>,<Pre1>,<Post1>,<Post2>,<Post3>,<Post4>,<Post5>

Supported preset values for M8041A and M8051A are 0 - 30 and for M8061A and M8062A, 0 is the only preset value.

This SCPI is applicable for M8041A, M8051A, M8061A and M8062A.

Example: :OUTP:DEEM:PRES 'M1.DataOut2', 2,1.1,2.2,4.3,6.7,5.3,6,3.3
:OUTP:DEEM:PRESet:ENABle?

Syntax :

:OUTP:DEEM:PRESet:ENABle 'identifier', <ON | OFF | 1 | 0>

:OUTP:DEEM:PRESet? 'identifier',

Input Parameters

'identifier': 'M*.DataOut1' or 'M*.DataOut2'

<ON | OFF | 1 | 0>

Return Range <ON | OFF | 1 | 0>,

Description This command enables or disables changing the deemphasis preset register number. Using this option, user can predefine separate set of cursor values and can apply it using preset register number.

This query returns the present state.

This SCPI is applicable for M8041A, M8051A and M8062A.

Example :OUTP:DEEM:PRESet:ENAB 'M2.DataOut2', ON

:OUTP:DEEM:SELect:PRESet?

Syntax :

:OUTP:DEEM:SELect:PRESet 'identifier', <NRf>

:OUTP:DEEM:SELect:PRESet? 'identifier'

Input Parameters

'identifier': 'M*.DataOut1' or 'M*.DataOut2'

<NRF>: Set current preset register.

Return Range 0 to 30

Description This command sets the current preset register. One in 31 registers can be defined as the ‘current’ preset register. The tap values of the ‘current’ preset register are shown at the output. Preset register 0 is the default register. The :PRECursor and :POSTCursor commands change the tap values only in that preselected preset register. There are 31 preset registers available addressed by a register index of 0 up to 30.

This command should not be sent in one PM (SCPI - Program Message) with the following de-emphasis commands:

- :PRESet
- :PRECursor
- :POSTCursor

In other words, the preset register should be selected before sending the values with the above de-emphasis commands.
Supported preset values for M8041A and M8051A are 0 - 30 and for M8061A and M8062A, 0 is the only preset value.

This SCPI is applicable for M8041A, M8051A, M8061A and M8062A.

Example: 
```
```

**:OUTPut:DEEMphasis:PRECursor(*)[?]**

**Syntax**: 
```
:OUTPut:DEEMphasis:PRECursor(*) 'identifier', <NRf>
:OUTPut:DEEMphasis:PRECursor(*)? 'identifier'
```

**Input Parameters**
- 'identifier': 'M*.DataOut1' or 'M*.DataOut2'
- <NRf>: Set selected pre-cursor value.

**Return Range**
- PRECursor1: -12.04 dB to 12.04 dB | 25% to 400%
- PRECursor2: -6.02 dB to 6.02 dB | 50% to 200%

**Description**
This command programs the pre-cursor value of the currently selected pre-cursor (pre-cursor 1 or pre-cursor 2).

This SCPI is applicable for M8041A, M8051A, M8061A and M8062A. For M8045A, the pre-cursor values cannot be changed. You can set the pre-cursor values by adjusting the coefficient values.

Example: 
```
:OUTP:DEEM:PREC2 'M1.DataOut2', 3.1
```

**:OUTPut:DEEMphasis:POSTcursor(*)[?]**

**Syntax**: 
```
:OUTPut:DEEMphasis:POSTcursor(*) 'identifier', <NRf>
:OUTPut:DEEMphasis:POSTcursor(*)? 'identifier'
```

**Input Parameters**
- 'identifier': 'M*.DataOut1' or 'M*.DataOut2'
- <NRf>: Set selected post-cursor value.

**Return Range**
- POSTcursor1: -20 dB to 20 dB | 10% to 1000%
- POSTcursor2, POSTcursor3: -12.04 dB to 12.04 dB | 25% to 400%
- POSTcursor4, POSTcursor5: -6.02 dB to 6.02 dB | 50% to 200%
Description

This command programs the post-cursor value of the currently selected post-cursor (post-cursor 1 through post-cursor 5).

This SCPI is applicable for M8041A, M8051A, M8061A and M8062A.

For M8045A, the post-cursor values cannot be changed. You can set the post-cursor values by adjusting the coefficient values.

Example


:OUTP:DEEMphasis:CONFigure:SIGN(*)[?] Syntax

:OUTP:DEEMphasis:CONFigure:SIGN(*) 'identifier', <enum>, <enum>, <enum>, <enum>, <enum>, <enum>, <enum>, <enum>

:OUTP:DEEMphasis:CONFigure:SIGN(*)? 'identifier'

Input Parameters

'identifier': 'M*.DataOut1' or 'M*.DataOut2'

<enum>: Specify positive/negative tap value sign for each tap.

Return Range

POS|NEG (returned for each tap)

Description

A list of "POSitive"s and "NEGative"s specify the sign of the corresponding tap.

The following list illustrates the meaning of the comma separated list:

<Pre2>,<Pre1>,<Pos1>,<Post2>,<Post3>,<Post4>,<Post5>

The order of the values corresponds to the order in the :PRESet command argument list. So sending the following comma separated list "pos,neg,pos,neg,neg,pos" determines that the pre-cursor 2 is a POSitive value, pre-cursor 1 is a NEGative value, post cursor 1 is a POSitive value and so on. This command influences all presets at once. For example, it is not possible to have a post cursor 2 with different signs in other preset registers.

This command accepts 3-taps or 7-taps sign and the meaning of the first tap changes depending upon the 3-taps or 7-taps sign sent. If 3-taps sign are sent, PreCursor1, PostCursor1, PostCursor2 will be signed. If 7-taps sign are sent, PreCursor2, PreCursor1, PostCursor1, PostCursor2, PostCursor3, PostCursor4, PostCursor5 will be signed.

This SCPI is applicable for M8041A, M8051A, M8061A and M8062A.

Example

The sign (+ve or -ve) for de-emphasis values (pre-cursor or post-cursor) is set using the command ":OUTPut:DEEMphasis:CONFigure:SIGN(*)". However, the same can now be directly programmed by using signed values with the commands ":OUTPut:DEEMphasis:PRECursor(*)", or ":OUTPut:DEEMphasis:POSTcursor(*)".

This SCPI is only supported by M8041A/M8051A (Hardware Revision 1) and M8061A modules.

:OUTPut:DEEMphasis:CURSor:MAGNitude(0|1|2|3|4|5|6|7)[?]

Syntax 
:OUTPut:DEEMphasis:CURSor:MAGNitude(0|1|2|3|4|5|6|7) 'Identifier', <NRf>
:OUTPut:DEEMphasis:CURSor:MAGNitude(0|1|2|3|4|5|6|7)? 'identifier'

Input Parameters
'identifier': 'M*.DataOut1' or 'M*.DataOut2'

Return Range <NRf>

Description This command sets the cursor magnitude as a signed value. The suffix selects the cursor to be addressed. The number of available cursors depends on the product number.

The M8045A supports cursors 0 to 4, where the main cursor is always cursor 2.

This query returns the present setting.

This SCPI is supported by M8045A.

Example 
:OUTP:DEEM:CURS:MAGN1 'M1.DataOut1','-0.05

:OUTPut:DEEMphasis:CURSor:MAIN:AUTO[?]

Syntax 
:OUTPut:DEEMphasis:CURSor:MAIN:AUTO 'Identifier', <ON|OFF|1|0>
:OUTPut:DEEMphasis:CURSor:MAIN:AUTO? 'identifier'

Input Parameters
'identifier': 'M*.DataOut1' or 'M*.DataOut2'
This command controls whether the main cursor magnitude shall be calculated automatically in order to maintain the overall peak to peak amplitude constant, or whether the main cursor magnitude can be controlled manually.

When this is set to ON, then the main cursor magnitude is calculated to fulfill the sum below.

$$\sum_{i}^{n} \text{abs}(\text{CursorMagnitude}_i) = 1$$

This query returns the present state.

This SCPI is supported by M8045A.

Example:

```
```

:OUTPut:DEEMphasis:CURSor:MAIN:SUFFix?

Syntax:

```
:OUTPut:DEEMphasis:CURSor:MAIN:SUFFix? 'Identifier'
```

Input Parameters:

- 'Identifier': 'M*.DataOut1' or 'M*.DataOut2'

Return Value: 2

Description: This query returns the main cursor’s suffix.

This SCPI is supported by M8045A.

Example:

```
```
:OUTPut:EINSertion Subnode

This subnode has the following SCPI structure:

```
:OUTPut
  EINSertion
    [:STATE][?]
    [:RATio][?]
    [:MODE][?]
    :ONCE
    :DISParity
```

This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:STATE][?]</td>
<td>:OUTPut:EINSertion[:STATE][?] on page 229</td>
</tr>
<tr>
<td>[:RATio][?]</td>
<td>:OUTPut:EINSertion[:RATio][?] on page 229</td>
</tr>
<tr>
<td>[:MODE][?]</td>
<td>:OUTPut:EINSertion[:MODE][?] on page 230</td>
</tr>
</tbody>
</table>
### :OUTPut:EINSertion[:STATe][?] Syntax

:OUTPut:EINSertion[:STATe] 'identifier', <ON|OFF|1|0>

:OUTPut:EINSertion[:STATe]? 'identifier'

| Input | Parameters
|-------|-------------
| 'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut' |
| <ON|OFF|1|0>: Switch error insertion output on or off. |

| Return Range | 1|0 |
| Description | This command switches the error insertion feature on or off for the specified output. |
| Example | :OUTP:EINS 'M1.DataOut1', on |

### :OUTPut:EINSertion:RATio[?] Syntax

:OUTPut:EINSertion:RATio 'identifier', <RM1|...|RM12>

:OUTPut:EINSertion:RATio? 'identifier'

| Input | Parameters
|-------|-------------
| 'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut' |
| <RM1|...|RM12>: Set the error insertion ratio. |

| Return Range | <RM1|...|RM12> |
| Description | This command controls the ratio of the internal fixed error insertion ratio. The range is 1E-12 (RM12) to 1E-1 (RM1). |
| Example | :OUTP:EINS:RAT 'M1.DataOut1', RM6 |
:OUTPut:EINSertion:MODE?

Syntax
:OUTPut:EINSertion:MODE 'identifier', <ERFSpacing|...|SINB>
:OUTPut:EINSertion:MODE? 'identifier'

Input Parameters
'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

<ERFSpacing|...|SINB>: Specify the error insertion mode.

Return Range
<ERFSpacing|...|SINB>

Description
This command is used to specify the error insertion mode as follows:

ERFSpacing: Generates errors with a predetermined bit error rate (fixed spacing).

ERATio: Generates errors with a predetermined bit error rate. Erroneous bits are randomly distributed.

CINA: Specifies CTRL IN A as a trigger.

CINB: Specifies CTRL IN B as a trigger.

BRK: Specifies software break command as a trigger.

SINA: Specifies SYS IN A as a trigger source.

SINB: Specifies SYS IN B as a trigger source.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.

Example
:OUTP:EINS:MODE 'M1.DataOut1', ERFSpacing

:OUTPut:EINSertion:ONCE

Syntax
:OUTPut:EINSertion:ONCE 'identifier'

Input Parameters
'identifier': 'M*.DataOut1', 'M*.DataOut2', or 'M*.DataOut'

Description
This command generates a single error (single bit error) on the specified output.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8045A.

Example
:OUTP:EINS:ONCE 'M1.DataOut1'
:OUTPut:EINSertion:DISParity

Syntax: :OUTPut:EINSertion:DISParity 'identifier'

Input

Parameters

Description: This command generates a single disparity error (only valid for 8b/10b operating mode) on the specified output.

This SCPI is applicable for M8041A, M8051A and M8045A.

Example: :OUTP:EINS:DISP 'M1.DataOut2'
This subnode has the following SCPI structure:

```
:OUTPut
  :TRIGger
    [:MODE?]?
    [:DIVider?]?
```

This subnode has the following commands:

**Table 54**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:DIVider[?]</td>
<td>:OUTPut:TRIGger:DIVider[?] on page 233</td>
</tr>
</tbody>
</table>

`:OUTPut:TRIGger:MODE[?]`

**Syntax**

:OUTPut:TRIGger:MODE 'identifier', <CLOCK|SEQUencer>

:OUTPut:TRIGger:MODE? 'identifier'

**Input Parameters**

'identifier': 'M1.TrigOut'

<CLOCK|SEQUencer>: Set the operating mode of the trigger out.

**Return Range**

CLOC|SEQ

**Description**

This command sets the operating mode of the trigger out (TRIG OUT) either to subrate clock or sequencer controlled mode.

This SCPI is applicable for M8041A and M8045A.

**Example**

:OUTP:TRIG:MODE 'M1.TrigOut', CLOC
`:OUTPut:TRIGger:DIVider[?]`

Syntax

`:OUTPut:TRIGger:DIVider 'identifier', <NRf>
`:OUTPut:TRIGger:DIVider? 'identifier`

Input Parameters

'identifier': 'M1.TrigOut'

<NRf>: Set the divider factor for the trigger out.

Return Range

2 to 65535

Description

This command sets the divider factor of the trigger out when the subrate clock operating mode is selected.

This SCPI is applicable for M8041A and M8045A.

Example

`:OUTP:TRIG:DIV 'M1.TrigOut', 8`
This subnode has the following SCPI structure:

```
:OUTPut
    :DEMBedding
        [:SCABle]
        [:STATe]?
        :SPARameter
            [:STATe]?
            :PROFile[?]
            :IPORt[?]
            :OPORt[?]
```

This subnode has the following commands:

Table 55

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SCABle:STATe[?]</td>
<td>:OUTPut:DEMBedding:SCABle:STATe[?] on page 235</td>
</tr>
<tr>
<td>:SPARameter:OPORt[?]</td>
<td>:OUTPut:DEMBedding:SPARameter:OPORt[?] on page 236</td>
</tr>
</tbody>
</table>
:OUTPut:DEMBedding:SCABle:STATe[?]

Syntax :OUTPut:DEMBedding[:SCABle][:STATe] 'identifier', <OFF | ON | 0 | 1>
:OUTPut:DEMBedding[:SCABle][:STATe]? 'identifier'

Input 'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', or 'M*.DataOut4'

Parameters OFF | ON | 0 | 1: Enables/disables standard cable compensation.

Return Range 0|1

Description This command switches on/off the standard cable compensation on the specified output.

The query return the present setting.

This SCPI is applicable for M8195A and M8196A.

Example :OUTP:DEM:SCAB:STAT 'M1.DataOut1', ON

:OUTPut:DEMBedding:SPARameter:STATe[?]

Syntax :OUTPut:DEMBedding:SPARameter[:STATe] 'identifier', <OFF | ON | 0 | 1>
:OUTPut:DEMBedding:SPARameter[:STATe]? 'identifier'

Input 'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', or 'M*.DataOut4'

Parameters <OFF | ON | 0 | 1>

Description This command switches on/off the compensation via S-parameter file on the specified output.

The query return the present setting.

This SCPI is applicable for M8195A and M8196A.

Example :OUTP:DEM:SPAR:STAT 'M1.DataOut1', ON

:OUTPut:DEMBedding:SPARameter:PROFile[?]


Input 'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', or 'M*.DataOut4'

Parameters

Description This command selects the S-Parameter profile. S-parameter is defined by frequency/real/imaginary tuples for 1 to 9 port components.

The query return the present setting.
This SCPI is applicable for M8195A and M8196A.

Example

```
'factory/huber_suhner_cable.s2p'
```

```
:OUTPut:DEMBedding:SPARameter:IPORt[?]
Syntax :OUTPut:DEMBedding:SPARameter:IPORt 'identifier', <NRf>
:OUTPut:DEMBedding:SPARameter:IPORt? 'identifier'

Input
'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', or 'M*.DataOut4'

Parameters
1|2|3|4:

Return Range
1|2|3|4; depends upon the selected S parameter file.

Description
This command selects the input port in the S-Parameter profile.
The query return the present setting.

This SCPI is applicable for M8195A and M8196A.

Example

```
:OUTP:DEMB:SPAR:IPOR 'M1.DataOut1', 1
```

```
:OUTPut:DEMBedding:SPARameter:OPORt[?]
Syntax :OUTPut:DEMBedding:SPARameter:OPORt 'identifier', <NRf>
:OUTPut:DEMBedding:SPARameter:OPORt? 'identifier'

Input
'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', or 'M*.DataOut4'

Parameters
1|2|3|4:

Return Range
1|2|3|4; depends upon the selected S-parameter file.

Description
This command selects the output port in the S-Parameter profile.
The query return the present setting.

This SCPI is applicable for M8195A and M8196A.

Example

```
```
:OUTPut:EMBedding Subnode

This subnode has the following SCPI structure:

```
:OUTPut
  :EMBedding
    :SPARameter
      [:STATE][?]
      :PROFile[?]
      :IPORT[?]
      :OPORT[?]
```

This subnode has the following commands:

Table 56

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SPARameter[:STATE][?]</td>
<td>:OUTPut:EMBedding:SPARameter[:STATE][?] on page 238</td>
</tr>
</tbody>
</table>
:OUTPut:EMBedding:SPARameter[:STATe]?  
**Syntax:** :OUTPut:EMBedding:SPARameter[:STATe] 'identifier', <OFF | ON | 0 | 1>  
:OUTPut:EMBedding:SPARameter[:STATe]? 'identifier'  
**Input Parameters**  
'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', or 'M*.DataOut4'  
**Return Range** 0|1  
**Description** This command Switches on/off the compensation via S-parameter file on the specified output.  
The query returns the present setting.  
This SCPI is applicable for M8195A and M8196A.  
**Example** :OUTP:EMB:SPAR:STAT 'M1.DataOut1', ON

:OUTPut:EMBedding:SPARameter:PROFile[?]  
**Syntax:** :OUTPut:EMBedding:SPARameter:PROFile 'identifier', <'Filename'>  
**Input Parameters**  
'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', or 'M*.DataOut4'  
**Description** This command selects the S-Parameter profile. S-parameter is defined by frequency/real/imaginary tuples for 1 to 9 port components.  
The query returns the present setting.  
This SCPI is applicable for M8195A and M8196A.  
**Example** OUTP:EMB:SPAR:PROF 'M1.DataOut1', 'factory/huber_suhner_cable.s2p'

:OUTPut:EMBedding:SPARameter:IPORt[?]  
**Syntax:** :OUTPut:EMBedding:SPARameter:IPORt 'identifier', <NRf>  
:OUTPut:EMBedding:SPARameter:IPORt? 'identifier'  
**Input Parameters**  
'identifier': 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', or 'M*.DataOut4'  
**Description** This command selects the input port in the S-Parameter profile.  
The query returns the present setting.
This SCPI is applicable for M8195A and M8196A.

Example

```
:OUTP:EMB:SPAR:IPOR 'M1.DataOut1', 1
```

### :OUTPut:EMBedding:SPARameter:OPORt[?]

**Syntax**

```
:OUTPut:EMBedding:SPARameter:OPORt 'identifier', <NRf>
:OUTPut:EMBedding:SPARameter:OPORt? 'identifier'
```

**Input Parameters**

- `identifier`: 'M*.DataOut1', 'M*.DataOut2', 'M*.DataOut3', or 'M*.DataOut4'

**Description**

This command selects the output port in the S-Parameter profile.

The query returns the present setting.

This SCPI is applicable for M8195A and M8196A.

Example

```
OUTP:EMB:SPAR:OPOR 'M1.DataOut1', 2
```
EVENt Subsystem

This subsystem has the following SCPI structure:

```
:EVENt
  :ADVance[?]
    :STATe[?]
    :SOURce[?]
```

This subsystem has the following commands:

**Table 57**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:EVENt:ADVance[?]</td>
<td>:EVENt:ADVance[?] on page 240</td>
</tr>
<tr>
<td>:EVENt:ADV:STATe[?]</td>
<td>:EVENt:ADV:STATe[?] on page 241</td>
</tr>
<tr>
<td>:EVENt:ADV:SOURce[?]</td>
<td>:EVENt:ADV:SOURce[?] on page 241</td>
</tr>
</tbody>
</table>

**:EVENt:ADVance[?]**

- **Syntax**: :EVENt:ADVance[?] 'identifier'
- :EVENt:ADVance? ‘identifier’
- **Input Parameters**: 'identifier': 'M1.EventConfig'
- **Description**: This command asserts the advance event in the M819xA Soft Front Panel. The query returns the present setting.
- **Example**: :EVEN:ADV:SOUR 'M1.EventConfig', TRIG
  TRIG
:EVENT:ADVance:STATe[?]  
**Syntax**
::EVENT:ADVance:STATe[?] 'identifier', <1 | 0 | ON | OFF>  
::EVENT:ADVance:STATe? 'identifier'  

**Input**
'identifier': 'M1.EventConfig'  

**Parameters**
1 | 0 | ON | OFF  

**Description**
This command enables/disables the hardware driven advance event in the Soft Front Panel. Triggering this event can also be done via software. The query returns the present setting. This SCPI is applicable for M8195A and M8196A.  

**Example**
::EVENT:ADV:STAT 'M1.EventConfig', 1  
::EVENT:ADV:STAT? 'M1.EventConfig'  
1  

:EVENT:ADVance:SOURce[?]  
**Syntax**
::EVENT:ADVance:SOURce[?] 'identifier', <TRIGger | EVENt>  
::EVENT:ADVance:SOURce? 'identifier'  

**Input**
'identifier': 'M1.EventConfig'  

**Parameters**
TRIGger | EVENt  

**Description**
This command allows to select whether the Trigger In or the Event In is used as hardware input for advance event. The query returns the present setting. This SCPI is applicable for M8195A and M8196A.  

**Example**
::EVENT:ADV:SOUR 'M1.EventConfig', TRIG  
::EVENT:ADV:SOUR? 'M1.EventConfig'  
TRIG
INPut Subsystem

The INPut subsystem controls the input ports of the analyzer. This subsystem has the following SCPI structure:

```
[:INPut
  [:BRM[?]
  [:DRATe?
  [:DELa[y[?]
  [:POLarity[?]
  [:CMVoltage[?]
    [:DETect
  [:THReshold[?]
  [:TVOLTage[?]
  [:TCONfig[?]
  [:CMODE[?]
  [:COUPling[?]
  [:SENSitivity[?]
  [:VOLTage
    [:WINDow[?]
      [:ALLocate[?]
      [:NRZ]:THReshold[?]
      [:PAM4]:SYMBol
        [:THReshold1|2|3][?]
      [:CDR
        ...
      [:ALIGNment
        ...
      [:EQUalization
        ...
```
The subsystem has the following commands and subnodes:

Table 58

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:BRM[?]</td>
<td>:INPut:DATA:BRMode[?] on page 244</td>
</tr>
<tr>
<td>:DRATe?</td>
<td>:INPut:DRATe? on page 244</td>
</tr>
<tr>
<td>:DElay</td>
<td>:INPut:DElay[?] on page 245</td>
</tr>
<tr>
<td>:POLarity</td>
<td>:INPut:POLarity[?] on page 245</td>
</tr>
<tr>
<td>:CMVoltage[?]</td>
<td>:INPut:CMVoltage[?] on page 246</td>
</tr>
<tr>
<td>:CMVoltage:DETECT</td>
<td>:INPut:CMVoltage:DETECT on page 246</td>
</tr>
<tr>
<td>:THReshold</td>
<td>:INPut:THReshold[?] on page 246</td>
</tr>
<tr>
<td>:TVOLtage</td>
<td>:INPut:TVOLtage[?] on page 247</td>
</tr>
<tr>
<td>:TCONfig[?]</td>
<td>:INPut:TCONfig[?] on page 247</td>
</tr>
<tr>
<td>:CMODE[?]</td>
<td>:INPut:CMODE[?] on page 248</td>
</tr>
<tr>
<td>COUPling[?]</td>
<td>:INPut:COUPling[?] on page 249</td>
</tr>
<tr>
<td>SENSitivity[?]</td>
<td>:INPut:SENSitivity[?] on page 249</td>
</tr>
<tr>
<td>[:VOLTage]:WINDow[?]</td>
<td>:INPut[:VOLTage]:WINDow[?] on page 249</td>
</tr>
<tr>
<td>[:VOLTage]:WINDow:ALLOCate[?]</td>
<td>:INPut[:VOLTage]:WINDow:ALLOCate[?] on page 250</td>
</tr>
<tr>
<td>:INPut[:VOLTage]:NRZ:THResholdd[?]</td>
<td>:INPut[:VOLTage]:NRZ:THResholdd[?] on page 251</td>
</tr>
<tr>
<td>:INPut[:VOLTage]:PAM4[:SYMBOL]:THReshold(1</td>
<td>2</td>
</tr>
</tbody>
</table>

Subnodes

- :CDR :INPut:CDR Subnode on page 252
- :ALIGNment :INPut:ALIGNment Subnode on page 262
- :EQUalization :INPut:EQUalization Subnode on page 270
:INPut:DATA:BRMode[?]

Syntax 
:INPut:DATA:BRMode 'identifier', < 0 | 1 | ON | OFF>
:INPut:DATA:BRMode? 'identifier'

Input Parameters
'identifier': 'M*.DataIn1' or 'M*.DataIn2'

Range 0 | 1 | ON | OFF

Description This command enables/disables the bit recovery mode. In BRM, the error detector does not expect any specific data. This mode can be used if the incoming data is completely unknown.

Example :INP:DATA:BRM 'M1.DataIn1',ON

NOTE The bit recovery mode is only available with CDR.

:INPut:DRATe?

Syntax :INPut:DRATe? 'identifier'

Input Parameters 'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*.DataIn'

Description This query returns the data rate for the input channel. This SCPI is applicable for M8061A M8062A and M8046A.

Example :INP:DRAT? 'M1.DataIn1'
:INPut:DELay[?]  
**Syntax:** :INPut:DELay 'identifier', <NRf>  
:INPut:DELay? 'identifier'  
**Input Parameters:** 'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*.DataIn', 'INF*.DataIn1' 
<NRf>: Specify sampling edge delay value.  
**Return Range:** For M8041A, M8051A and M8046A: -6.7 ns to +6.7 ns  
For M8061A: 0 ps to 60 ps  
For M8062A: -10 ns to 10 ns  
For M8046A: -6.7 ns to 6.7 ns  
**Description:** This command specifies the delay of the sampling edge.  
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8046A and DSO634A, DSA634A, DSAX96204A and DSOX96204A.  
**Example:** :INP:DEL 'M1.DataIn1', 1e-9

:INPut:POLarity[?]  
**Syntax:** :INPut:POLarity 'identifier', <NORMal | INVerted | EITHer>  
:INPut:POLarity? 'identifier'  
**Input Parameters:** 'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*.DataIn', 'INF*.DataIn1'  
<NORMal | INVerted | EITHer>: Set input polarity. The EITHer option only works with the M8195A AWG module.  
**Return Range:** NORM | INV | EITH  
**Description:** This command sets the input polarity of the specified input to either NORMal, INVerted or EITHer.  
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8046A, M8195A, DSO634A, DSA634A, DSAX96204A and DSOX96204A.  
**Example:** :INP:POL 'M1.DataIn1', INV
:`INPut:CMVoltage[?]`

**Syntax**

`:INPut:CMVoltage 'identifier', <NRf>`

`:INPut:CMVoltage? 'identifier'`

**Input Parameters**

'identifier': 'M*.DataIn1' or 'M*.DataIn2'

<NRf>: Set common mode voltage.

**Return Range**

-1.0 V to 2.6 V; The range is dependent on other system parameters. See Command Syntax to Find Min/Max Values on page 110.

**Description**

This command is used to measure the common mode voltage of the selected data input by the module itself. Acceptable units are mV, V and exponents (for example, 300E-3 is the same as 300 mV).

This SCPI is applicable for M8041A and M8051A.

**Example**

`:INP:CMV 'M1.DataIn1', 300mV`

:`INPut:CMVoltage:DETect`

**Syntax**

`:INPut:CMVoltage:DETect 'identifier`

**Input Parameters**

'identifier': 'M*.DataIn1' or 'M*.DataIn2'

**Description**

This command measures the common mode voltage at the selected data input and applies it to the comparator.

This SCPI is applicable for M8041A and M8051A.

**Example**

`:INP:CMV:DET 'M1.DataIn1'

:`INPut:THReshold[?]`

**Syntax**

`:INPut:THReshold 'identifier', <NRf>`

`:INPut:THReshold? 'identifier'`

**Input Parameters**

'identifier': 'M*.DataIn1', 'M*.DataIn2', 'M2.ElectIdleIn', 'M*.DataIn', 'SysIn A/B' or 'CtrlIn A/B'

<NRf>: Set receiver input threshold voltage.

**Return Range**

For M8041A and M8051A: -1 V to 1 V

For M8062A: -800 mV to 800 mV

For M8046A: -250 mV to 250 mV
Description
This command sets the threshold voltage of the selected receiver input connector (decision threshold). Acceptable units are mV, V and exponents (for example, 300E-3 is the same as 300 mV).

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example
:INP:THR 'M1.DataIn1', 200 mV

:INPut:TVOLtage[?]
Syntax
:INPut:TVOLtage 'identifier', <NRf>
:INPut:TVOLtage? 'identifier'

Input Parameters
'identifier': 'M*.DataIn1', 'M*.DataIn2', or 'M2.ElectIdleIn'

<NRf>: Specify receiver input termination voltage.

Return Range
-1 V to 2.9 V; The range is dependent on other system parameters. See Command Syntax to Find Min/Max Values on page 110.

Description
This command specifies the termination voltage of the selected receiver input connector. Acceptable units are mV, V and exponents (for example, 300E-3 is the same as 300 mV).

This SCPI is applicable for M8041A, M8051A and M8062A.

Example
:INP:TVOL 'M1.DataIn1', 300 mV

:INPut:TCONfig[?]
Syntax
:INPut:TCONfig 'identifier', <BALanced|UNBalanced>
:INPut:TCONfig? 'identifier'

Input Parameters
'identifier': 'M*.DataIn1' or 'M*.DataIn2'

<BALanced|UNBalanced>: Select input termination configuration.

Return Range
BAL|UNB

Description
This command selects the termination configuration for an input front end. The following options are available:

BALanced: balanced or differential

UNBalanced: unbalanced with respect to ground (termination voltage)

This SCPI is applicable for M8041A and M8051A.

Example
:INP:TCON 'M1.DataIn1', BAL
:INPut:CMODe[?]

Syntax
:INPut:CMODe 'identifier',
<SENormal|SEComplement|DIFFerential|SINGleended>
:INPut:CMODe? 'identifier'

Input Parameters
'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn', 'INF*.DataIn1'

<SENormal|SEComplement|DIFFerential|SINGleended>: Select compare mode.

Return Range
SEN|SEC|DIFF|SING

Description
This command defines the compare mode. This command defines which input (DATA or DATA\) is active. The following options are available:

SENormal: Data stream is compared at Normal input; Complement is terminated or connected to the complement output of the source. This allows to measure a single-ended signal as well as the non-inverted signal of a differential pair.

SEComplement: Data stream is compared at Complement input; Normal is terminated or connected to the normal output of the source. This allows to measure a single-ended signal as well as the inverted signal of a differential pair.

DIFFerential: If differential both input ports need to receive a signal. The actual data signal is measured as the voltage difference between the two incoming signals.

SINGleended: Data stream is provided at either Normal input or Complement input. The other input is either left open or terminated with 50 Ohm.

Note that the availability of the particular selection depends on the currently used hardware.

This SCPI is applicable for M8041A, M8051A, M8061A, M8046A DSO634A, DSA634A, DSOX96204A and DSAX96204A.

Example
:INP:CMOD 'M1.DataIn1', SEC
:INPut:COUPling

Syntax :INPut:COUPling 'identifier', <AC|DC>
:INPut:COUPling? 'identifier'

Input 'identifier': 'M*.DataIn1' or 'M*.DataIn2'
Parameters <AC|DC>: Select input coupling.

Return Range AC|DC
Description This command selects DC or AC input coupling.
This SCPI is applicable for M8041A and M8051A.
Example :INP:COUP 'M1.DataIn1', DC

:INPut:SENSitivity

Syntax :INPut:SENSitivity 'identifier', <HIGH|NORMal>
:INPut:SENSitivity? 'identifier'

Input 'identifier': 'M*.DataIn1' or 'M*.DataIn2'
Parameters <HIGH|NORMal>: Select input coupling.

Return Range HIGH|NORM
Description This command sets the error detector input sensitivity to high or normal.
This SCPI is applicable for M8041A, M8051A and M8062A.
Example :INP:SENS 'M1.DataIn1', HIGH

:INPut[:VOLTage]:WINDow

Syntax :INPut[:VOLTage]:WINDow? [?] 'identifier', < NRf>
:INPut[:VOLTage]:WINDow? 'identifier'

Input 'identifier': 'M*.DataIn' or 'INF*.DataIn1'
Parameters Description This command defines the input voltage window. The voltage applied at
the Data In connector needs to be within the voltage range defined by the
common mode voltage and the voltage window.
This SCPI is supported by M8046A and all supported real-time oscilloscopes.

**Example**

```plaintext
:INP:WIND 'M2.DataIn1', 0.4
```

### :INPut[:VOLTage]:WINDow:ALLocate(?)

**Syntax**

```plaintext
:INPut[:VOLTage]:WINDow:ALLocate 'Identifier', < NRf>
:INPut[:VOLTage]:WINDow:ALLocate? 'Identifier'
```

**Input Parameters**

- `'Identifier': 'INF*.DataIn1'`

**Description**

This command defines how much of the input voltage window shall be used by the received signal.

This controls how the auto-set algorithm calculates the input voltage window size based on the measured signal amplitude. Normally the input voltage window is chosen larger than the signal amplitude to avoid nonlinear distortions at the peak voltage levels.

When working with small signal amplitudes, the BER results of the real-time oscilloscopes may be better if the received signal is using the input voltage range is used to a larger extent, than what the oscilloscope choses during the built-in vertical auto-scale operation.

Changing this parameter is not directly changing the input voltage range, but has effect on the next vertical auto-scale operation.

This SCPI is only supported by for real-time oscilloscopes.

**Example**

```plaintext
:INP:WIND:ALL 'INF1.DataIn1'; 90
```
**:INPut[:VOLTage][:NRZ]:THReshold[?]**

Syntax:
```
:INPut[:VOLTage][:NRZ]:THReshold 'identifier', <NRf>
:INPut[:VOLTage][:NRZ]:THReshold? 'identifier'
```

Input Parameters:
- `<identifier>`: Any valid data input identifier. M*.DataIn1 or M*.DataIn2

Description:
The command specifies the threshold voltage of a RX input connector. This query returns the present setting.

Example:
```
:INP:THR 'M2.DataIn1', 0.2
```

---

**:INPut[:VOLTage]:PAM4[:SYMBol]:THReshold(1|2|3)[?]**

Syntax:
```
:INPut[:VOLTage]:PAM4[:SYMBol]:THReshold(1|2|3)[?] 'identifier', <NRf>
:INPut[:VOLTage]:PAM4[:SYMBol]:THReshold(1|2|3)? 'identifier'
```

Input Parameters:
- `<identifier>`: Any valid data input identifier. M*.DataIn

Description:
The command controls the decision threshold of the PAM4 decoder within the data input’s input window. The setting of the PAM4 decision thresholds always refers to the actual input voltage range.

The 3 decision thresholds are applied as described in the following table:

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Decides between symbol 0 and symbol 1.</td>
</tr>
<tr>
<td>2</td>
<td>Decides between symbol 1 and symbol 2.</td>
</tr>
<tr>
<td>3</td>
<td>Decides between symbol 2 and symbol 3.</td>
</tr>
</tbody>
</table>

This query returns the present state.

This SCPI is applicable for M8046A.

It is best practice to configure the 3 decision thresholds in a combined command transaction (see the example) to minimize hardware re-configuration duration.

Example:
```
:INP:PAM4:SYMB:THR1 'M2.DataIn',-100e-3;THR2 'M2.DataIn',-5e-3;THR3 'M2.DataIn',100e-3
```
:INPut:CDR Subnode

This subnode has the following SCPI structure:

```
:INPut
  :CDR
      :LORDer[?]
      :CTRL[?]
      [:STATe][?]
      :AUTO[?]
      :RELOck
      :HIGHTD[?]
      :OPTimize
      :FIRST
      :TDENsity[?]
      :LBANdwidth[?]
    :SECond
      :TDENsity[?]
      :LBANdwidth[?]
      :PEAKing[?]
    :OJTF
      :STATe[?]
      :SECond[?]
      :LBANdwidth[?]
      :DIVisor[?]
      :DFACtor[?]
```

This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:LORDer[?]</td>
<td>:INPut:CDR:LORDer[?] on page 254</td>
</tr>
<tr>
<td>:CTRL[?]</td>
<td>:INPut:CDR:CTRL[?] on page 254</td>
</tr>
<tr>
<td>[:STATe][?]</td>
<td>:INPut:CDR[:STATe][?] on page 255</td>
</tr>
<tr>
<td>:RELOck</td>
<td>:INPut:CDR:RELOck on page 256</td>
</tr>
<tr>
<td>Name</td>
<td>Description under</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>:HIGHTD[?]</td>
<td>:INPut:CDR:HIGHTD[?] on page 256</td>
</tr>
<tr>
<td>OPTimize</td>
<td>:INPut:CDR:OPTimize on page 256</td>
</tr>
<tr>
<td>width[?]</td>
<td></td>
</tr>
<tr>
<td>width:DIVisor[?]</td>
<td></td>
</tr>
<tr>
<td>width:DIVisor[?]</td>
<td></td>
</tr>
</tbody>
</table>
:INPut:CDR:LORDer[?]  
Syntax  :INPut:CDR:LORDer 'identifier', <FIRSt|SECond>  
         :INPut:CDR[:STATe]? 'identifier'  
Input  'identifier': 'M*.DataIn1' or 'M*.DataIn2'  
Parameters  <FIRSt|SECond>: Set the loop order of the CDR.  
Return Range  FIRS|SEC  
Description  This command determines the loop order of the CDR. Depending on this setting, defining the behavior of the CDR is done by different parameters. For first order only, transition density and bandwidth fully define its characteristics while for the second order this definition needs at least one more parameter i.e. peaking.  
           This SCPI is applicable for M8041A and M8051A.  
Example  :INP:CDR:LORD 'M1.DataIn1', FIRS

:INPut:CDR:CTRL[?]  
Syntax  :INPut:CDR:CTRL 'identifier', <MANual|SEQ>  
         :INPut:CDR:CTRL? 'identifier'  
Input  'identifier': 'M*.DataIn1' or 'M*.DataIn2'  
Parameters  <MANual|SEQ>: Set the CDR control method.  
Return Range  MAN|SEQ  
Description  This command selects whether the CDR is controlled manually or by the sequencer.  
           This SCPI is applicable for M8041A and M8051A.  
Example  :INP:CDR:CTRL 'M1.DataIn1', SEQ
:INPut:CDR[:STATe][?]

Syntax  
:INPut:CDR[:STATe] 'identifier', <ON|OFF|1|0>
:INPut:CDR[:STATe]? 'identifier'

Input Parameters
'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'

<ON|OFF|1|0>: Enable/disable the CDR.

Return Range 1|0

Description This command enables or disables CDR.

This SCPI is applicable for M8041A, M8051A and M8046A.

Example  
:INP:CDR 'M1.DataIn1', ON

:INPut:CDR:AUTO[?]

Syntax  
:INPut:CDR:AUTO 'identifier', <ON|OFF|1|0>
:INPut:CDR:AUTO? 'identifier'

Input Parameters
'identifier': 'M*.DataIn'

<ON|OFF|1|0>: Enable/disable the auto relock.

Return Range 1|0

Description This command enables or disables the auto relock.

This SCPI is applicable for M8062A.

Example The following SCPI command sets the 'auto relock' to 1.

:INPut:CDR:AUTO 'M2.DataIn1', 1

The following SCPI command queries the 'auto relock' state.

:INPut:CDR:AUTO? 'M2.DataIn1'

1
:INPut:CDR:RELOck

Syntax  :INPut:CDR:RELOck ‘identifier’

Input   ‘identifier’: ‘M*.DataIn’

Parameters

Description  This command execute a full CDR relock.
              This SCPI is applicable for M8062A.

Example    :INPut:CDR:RELOck ‘M2.DataIn’

:INPut:CDR:HIGHTD[?]

Syntax  :INPut:CDR:HIGHTD ‘identifier’, <ON|OFF|1|0>
          :INPut:CDR:HIGHTD? ‘identifier’

Input   ‘identifier’: ‘M*.DataIn’

Parameters  <ON|OFF|1|0>: Set the High Transition Density state to ON or OFF.

Return Range  0|1

Description  This command sets the High Transition Density state to ON or OFF.
              This SCPI is applicable for M8062A.

Example    :INPut:CDR:HIGHTD ‘M2.DataIn’, 1

:INPut:CDR:OPTimize

Syntax  :INPut:CDR:OPTimize ‘identifier’

Input   ‘identifier’: ‘M*.DataIn’

Parameters

Description  This command invokes the M8062A CDR optimization at the current data rate and optionally applied jitter.
              This SCPI is applicable for M8062A.

Example    :INPut:CDR:OPTimize ‘M2.DataIn’
:INPut:CDR:FIRSt:TDENsity[?]

Syntax: :INPut:CDR:FIRSt:TDENsity 'identifier', <NRf>

:INPut:CDR:FIRSt:TDENsity? 'identifier'

Input Parameters
- 'identifier': 'M*.DataIn1' or 'M*.DataIn2'
- <NRf>: Set first order transition density.

Return Range: 25% to 100%

Description: This command is used to set the first order transition density of the incoming signal from 25% to 100%. The default is 50%.

This SCPI is applicable for M8041A and M8051A.

Example: :INP:CDR:FIRS:TDEN 'M1.DataIn1', 25

:INPut:CDR:FIRSt:LBANdwidth[?]

Syntax: :INPut:CDR:FIRSt:LBANdwidth 'identifier', <NRf>

:INPut:CDR:FIRSt:LBANdwidth? 'identifier'

Input Parameters
- 'identifier': 'M*.DataIn1' or 'M*.DataIn2'
- <NRf>: Set first order loop bandwidth of the CDR.

Return Range: 102 kHz to 20 MHz

Description: This command is used to set the first order loop bandwidth of the CDR. The default is 2 MHz. Acceptable units are Hz, kHz, MHz and exponents (for example, 4E6 is the same as 4 MHz).

This SCPI is applicable for M8041A and M8051A.

Example: :INP:CDR:FIRS:LBAN 'M1.DataIn1', 4MHz
:INPut:CDR:SECond:TDENsity[?]
Syntax :INPut:CDR:SECond:TDENsity 'identifier', <NRf>
Input Parameters 'identifier': 'M*.DataIn1' or 'M*.DataIn2'
<NRf>: Set second order transition density.
Return Range 25% to 100%
Description This command is used to set the second order transition density of the incoming signal from 10% to 100%. The default is 50%.
This SCPI is applicable for M8041A and M8051A.
Example :INP:CDR:SEC:TDEN 'M1.DataIn1', 25

:INPut:CDR:SECond:LBANdwidth[?]
Syntax :INPut:CDR:SECond:LBANdwidth 'identifier', <NRf>
Input Parameters 'identifier': 'M*.DataIn1' or 'M*.DataIn2'
<NRf>: Set second order loop bandwidth of the CDR.
Return Range 102 kHz to 20 MHz
Description This command is used to set the second order loop bandwidth of the CDR. The default is 2 MHz. Acceptable units are Hz, kHz, MHz and exponents (for example, 4E6 is the same as 4 MHz).
This SCPI is applicable for M8041A and M8051A.
Example :INP:CDR:SEC:LBAN 'M1.DataIn1', 4MHz
:INPut:CDR:SECond:PEAKing[?]

Syntax  
:INPut:CDR:SECond:PEAKing 'identifier', <NRf>

Input Parameters  
'identifier': 'M*.DataIn1' or 'M*.DataIn2'

<NRf>: Adjust second order peaking of the CDR.

Return Range  
0 dB to 3 dB

Description  
This command adjusts the second order peaking of the CDR which is valid for the jitter transfer function. The default is 1 dB.

This SCPI is applicable for M8041A and M8051A.

Example  
:INP:CDR:SEC:PEAK 'M1.DataIn1', 0.5
The :INPut:CDR:OJTF subnode allows controlling the clock recovery on an 
error detector channel that is based on a real-time oscilloscope.

:INPut:CDR:OJTF[:STATe]?[

Syntax
:INPut:CDR:OJTF[:STATe] <identifier>,{OFF | ON | 0 | 1}
:INPut:CDR:OJTF[:STATe]? <identifier>

Identifier 'INF*.DataIn1'

Description This command enables or disables the use of the clock recovery. For the 
current versions of M8070A, the clock recovery will always be enabled and 
cannot be disabled.
This query returns the present setting.
This SCPI is applicable for DSO634A, DSA634A, DSOX96204A and 
DSAX96204A.

Example :INP:CDR:OJTF:STAT 'INF1.DataIn1',ON


Syntax

Identifier 'INF*.DataIn1'

Description This command configures the loop bandwidth of the clock recovery. For 
more details, refer the corresponding command in the Keysight Infiniium 
Oscilloscopes Programmer’s Guide.
This query returns the present setting.
This SCPI is applicable for DSO634A, DSA634A, DSOX96204A and 
DSAX96204A.

Example :INP:CDR:OJTF:SEC:LBAN 'INF1.DataIn1',12.5e6
\texttt{:INPut:CDR:OJTF:SECond:LBANdwidth:DIVisor[?]} \hfill SCPI Command Reference

\textbf{Syntax} \hfill \texttt{:INPut:CDR:OJTF:SECond:LBANdwidth:DIVisor 'identifier', \textless NRf\textgreater }
\hfill \texttt{:INPut:CDR:OJTF:SECond:LBANdwidth:DIVisor? 'identifier'

\textbf{Identifier} \hfill 'INF*.DataIn1'

\textbf{Description} \quad This command configures the loop bandwidth of the CDR by a symbol rate divisor. The resulting loop bandwidth will be the channel's symbol rate divided by the loop bandwidth divisor.

This query returns the present setting.

This SCPI is applicable for DSO634A, DSA634A, DSOX96204A and DSA96204A.

\textbf{Example} \quad \texttt{:INP:CDR:OJTF:SEC:LBAN:DIV 'INF1.DataIn1',1667}

\texttt{:INPut:CDR:OJTF:SECond:DFACtor[?]} \hfill SCPI Command Reference

\textbf{Syntax} \hfill \texttt{:INPut:CDR:OJTF:SECond:DFACtor 'identifier', \textless NRf\textgreater }
\hfill \texttt{:INPut:CDR:OJTF:SECond:DFACtor? 'identifier'

\textbf{Identifier} \hfill 'INF*.DataIn1'

\textbf{Description} \quad This command configures the damping factor of the clock recovery. For more details, refer the corresponding command in the \textit{Keysight Infiniium Oscilloscopes Programmer's Guide}.

This query returns the present setting.

This SCPI is applicable for DSO634A, DSA634A, DSOX96204A and DSA96204A.

\textbf{Example} \quad \texttt{:INP:CDR:OJTF:SEC:DFAC 'INF1.DataIn1',0.707}
:INPut:ALIGNment Subnode

This subnode has the following SCPI structure:

```
:INPut
  :ALIGNment
    :EYE
      :[AUTO]
      :ACENter
      :TCENter
      :ABORT
      :THReshold[?]
      :RESult
      :[NRZ]:[HEIGth]?
      :WIDTh?
      :[NRZ]:THReshold?
      :PAM4:HEIGht(1|2|3)?
      :PAM4:THReshold(1|2|3)?
      :DELay?
      :POLarity?
    :STATus[:VALue]?
```

This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
<th>Page</th>
</tr>
</thead>
</table>
### :INPut:ALIgnment:EYE[:AUTO]

**Syntax**

```
:INPut:ALIgnment:EYE[:AUTO] 'identifier'
```

**Input Parameters**

- `identifier`: 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'

**Description**

This command starts an auto alignment.

This SCPI is applicable for M8041A, M8051A, M8062A, M8046A, DSO634A, DSA634A, DSOX96204A and DSAX96204A.

**Example**

`:INP:ALIG:EYE 'M1.DataIn1'`

---

### :INPut:ALIgnment:EYE:ACEN

**Syntax**

```
:INPut:ALIgnment:EYE:ACEN 'identifier'
```

**Input Parameters**

- `identifier`: 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'

**Description**

This command initiates a search for the 0/1 threshold voltage midway between the two 0/1 threshold voltages with a measured BER just in excess of the BER configured by the :EYE:THReshold command. If
successful, the command leaves the 0/1 threshold at this value.

This SCPI is applicable for M8041A, M8051A, M8046A, DSO634A, DSA634A, DSOX96204A and DSAX96204A.

Example: \texttt{:INP:ALIG:EYE:ACEN ‘M1.DataIn1’}

\textbf{\texttt{:INPut:ALIgnment:EYE:TCEN}}

Syntax: \texttt{:INPut:ALIgnment:EYE:TCEN ‘identifier’}

Input: ‘identifier’: ‘M*.DataIn1’, ‘M*.DataIn2’ or ‘M*DataIn’

Parameters

Description: This command initiates a search for the value of data/clock delay that puts the active clock edge in the center of the data eye, midway between the two relative delay points with a measured BER just in excess of the BER configured by the \texttt{:EYE:THReshold} command. If successful, the command leaves the data/clock delay at this value.

This SCPI is applicable for M8041A, M8051A, M8046A, DSO634A, DSA634A, DSOX96204A and DSAX96204A.

Example: \texttt{:INP:ALIG:EYE:TCEN ‘M1.DataIn1’}

\textbf{\texttt{:INPut:ALIgnment:EYE:ABORt}}

Syntax: \texttt{:INPut:ALIgnment:EYE:ABORt ‘identifier’}

Input: ‘identifier’: ‘M*.DataIn1’, ‘M*.DataIn2’ or ‘M*DataIn’

Parameters

Description: This command interrupts and aborts an alignment.

This SCPI is applicable for M8041A, M8051A, M8062A, M8046A, DSO634A, DSA634A, DSOX96204A and DSAX96204A.

Example: \texttt{:INP:ALIG:EYE:ABOR ‘M1.DataIn1’}
**:INPut:ALIgnment:EYE:THReshold[?]**

**Syntax:** :INPut:ALIgnment:EYE:THReshold 'identifier', <NRf>

:INPut:ALIgnment:EYE:THReshold? 'identifier'

**Input Parameters**
- 'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'
- <NRf>: Set BER threshold.

**Return Range** 1E-9 to 1E-1

**Description** The command sets the BER threshold to be used in the determination of the edges of the eye. The query returns the current BER threshold value.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example** :INP:ALIG:EYE:THR 'M1.DataIn1', 1E-3

**:INPut:ALIgnment:EYE:RESult[:NRZ][:HEIGht]?**

**Syntax** :INPut:ALIgnment:EYE:RESult[:NRZ][:HEIGht]? 'identifier'

**Input Parameters**
- 'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'

**Description** This is a query command that searches for the value of data amplitude that puts the 0/1 threshold level midway between the upper and lower bounds at which the error ratio exceeds the threshold value set by the :INPut:ALIgnment:EYE:THReshold command.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example** :INP:ALIG:EYE:RES? 'M1.DataIn1'

**:INPut:ALIgnment:EYE:RESult:WIDTh?**

**Syntax** :INPut:ALIgnment:EYE:RESult:WIDTh? 'identifier'

**Input Parameters**
- 'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'

**Description** This is a query command that interrogates the eye width found by the most recent search for the value of data/clock delay that put the active edge in the center of the data eye.
This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.


`:INP:ALIGNment:EYE:RESult[:NRZ]:THRehold?

Syntax: `:INP:ALIGNment:EYE:RESult[:NRZ]:THRehold? 'identifier'`

Input: 'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'

Description: Returns the threshold detected at the last alignment run.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.


`:INP:ALIGNment:EYE:RESult:PAM4:HEIGht(1|2|3)?

Syntax: `:INP:ALIGNment:EYE:RESult:PAM4:HEIGht(1|2|3)? 'identifier'`

Input: 'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'

Description: This query returns the value of data amplitude that puts the 0/1 threshold level midway between the upper and lower bounds at which the error ratio exceeds the threshold value set by the `:INP:ALIGNment:EYE:THRehold` command.

The suffix addresses the height of the lower eye (1), middle eye (2) or upper eye (3). If the suffix is omitted, then the returned value is for the lower eye of the PAM4 signal.

This SCPI is applicable for M8046A.


`:INP:ALIGNment:EYE:RESult:PAM4:THRehold(1|2|3)?

Syntax: `:INP:ALIGNment:EYE:RESult:PAM4:THRehold(1|2|3)? 'identifier'`

Input: 'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'

Description: This query returns the threshold detected at the last alignment run. The suffix addresses the threshold of the lower eye (1), middle eye (2) or upper eye (3). If the suffix is omitted, then the returned value is for the lower eye...
of the PAM4 signal.
This SCPI is applicable for M8046A.

:INPut:ALIgnment:EYE:RESult:DELay?

Input 'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'
Parameters

Description Returns the delay detected at the last alignment run.
This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.
Example :INP:ALIG:EYE:RES:DEL? 'M1.DataIn1'

:INPut:ALIgnment:EYE:RESult:POLarity?

Input 'identifier': 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'
Parameters

Return Range NOR|INV

Description Returns the polarity of the signal detected at the last alignment run.
This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.
Example :INP:ALIG:EYE:RES:POL? 'M1.DataIn1'
This subnode reports alignment specific status information and offers an alternative way of retrieving the same status information that is available via the :STATUs:INStrument:ALIgnment Subnode on page 131.

:INPut:ALIgnment:STATus[:VALue]?

Syntax  :INPut:ALIgnment:STATus[:VALue]? 'Identifier'

Input 'identifier': 'Any error detector location and group name. E.g. 'M1.DataIn1', 'M2.DataIn', 'INF1.DataIn1'

Parameters 'M1.DataIn1', 'INF1.DataIn1'

Description This command retrieves alignment status information for an error detector or error detector group.

This command can be used to retrieve information about the alignment procedure that is working on the addressed error detector channel, or a group of error detectors. This command is reporting the status independently of the executing alignment procedure. There is no differentiation in respect to different types of alignments.

This command returns the following values:

- INPROGRESS - At least one addressed error detector is executing an alignment procedure.
- ABORTED - The alignment procedure has been aborted by the user before all aligning error detector channels finished the procedure.
- FAILED - All addressed error detector channels have finished the alignment procedure, but at least one was unable to finish successfully.
- SUCCESS - The addressed error detector channels have successfully finished the alignment procedure. Error detector channels that are addressed by the command but return UNKNOWN state, do not contribute to the result.
- UNKNOWN - None of the addressed error detectors has executed an alignment procedure yet.
NOTE

This command cannot be used in the same compound SCPI statement that is starting the alignment procedure. If this command is sent to the instrument right after the command that is initiating the alignment procedure, then the returned value is undefined and can reflect the state that was valid before the alignment. This can be overcome by reading any value from the instrument (e.g. *ESE?), except for *OPC? as this will block until the alignment procedure is finished.

Example: :INP:ALIGN:STAT? 'M1.DataIn1'
This subnode controls the equalizer settings on a data input. Depending on the product number different methods of equalizations are available.

This subnode has the following SCPI structure:

```
:INPut:EQUalization
  [:PREset][:LEVel][:STATe][:NTAPs][:NPRecursor][:TAP][:AUTO][:INCLude]
```

This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:INPut:EQUalization[:PREset]:LEVel</td>
<td>:INPut:EQUalization[:PREset]:LEVel on page 272</td>
</tr>
</tbody>
</table>
:INPut:EQUalization[:PREset]? Syntax

```
:INPut:EQUalization[:PREset] 'Identifier', <OFF | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10>
```

Input Parameters
- 'identifier': 'M*.DataIn1' or 'M*.DataIn2'
- `<OFF | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10>

Return Range
- OFF | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10

Description
This command is used to select from the following equalization presets:

For M8041A and M8051A:
- OFF - Disables equalization
- P1 - PCIe 8Gb/s -6 dB
- P2 - PCIe 8Gb/s -9 dB
- P3 - PCIe 8Gb/s -12 dB
- P4 - USB 5Gb/s -3.5 dB
- P5 - PCIe 16Gb/s -6 dB
- P6 - PCIe 16Gb/s -9 dB
- P7 - PCIe 16Gb/s -12 dB
- P8 - USB 10Gb/s 0 dB
- P9 - USB 10Gb/s -3 dB
- P10 - USB 10Gb/s -6 dB

For M8061A:
- OFF - off
- P1 - PCIe 8Gb/s -6 dB
- P2 - PCIe 8Gb/s -9 dB
- P3 - PCIe 8Gb/s -12 dB
- P4 - USB 5Gb/s -3.5 dB
For M8062A:

- P1 - Low
- P2 - Medium
- P3 - High

The availability of individual presets depends on the type of module being controlled and the calibration status of the respective module.

This SCPI is applicable for M8041A, M8051A, M8061A and M8062A.

For M8062A, the equalizer is available only for 20 Gb/s and higher. For other values, if a command is sent, it will return OFF and will notify that the value is read-only.

Example: `:INP:EQU 'M1.DataIn1',P1`

`:INPut:EUQalization[:PRESet]:LEVel[?]`

Syntax: `:INPut:EUQalization[:PRESet]:LEVel 'identifier', <NRf>`

Input Parameters:
- 'identifier': 'M*.DataIn'
- The specified number has a range from 0 – 120.

Return Range: 0 – 120

Description: This command controls the integrated and adjustable equalization of M8046A, to compensate for loss characteristic of the back channel. This feature is available for NRZ and PAM4 signals.

Equalization can be adjusted from no equalization (0) to maximum equalization (120) in equidistant steps. The available range is restricted to 0 – 55 for PAM4 signals.

This SCPI is only applicable for M8046A.

Example: `:INP:EQU:LEV 'M1.DataIn',12`

`:INP:EQU:LEV? 'M1.DataIn'`
:INPut:EQUalization:FFEQualizer Subnode

This subnode configures a Feed-Forward Equalizer on a data input. This subnode is available for real-time oscilloscope based error detector channels. It requires the existence for the Equalization license (option DEQ) on the real-time oscilloscope. If the license is not available, then equalization will not be offered in M8070A for the data input.

:INPut:EQUalization:FFEQualizer:STATe[?]

Syntax :INPut:EQUalization:FFEQualizer:STATe <identifier>,{OFF|ON|0|1}

:INPut:EQUalization:FFEQualizer:STATe? <identifier>

Identifier 'INF*.DataIn1'

Description This command enables or disables the use of the FFE algorithm on the given data input.

This query returns the present setting.

This SCPI is applicable for DSO634A, DSA634A, DSOX96204A and DSAX96204A.

Example :INP:EQU:FFEQ:STAT 'INF1.DataIn1',ON

:INPut:EQUalization:FFEQualizer:NTAPs[?]

Syntax :INPut:EQUalization:FFEQualizer:NTAPs <identifier>,<number>

:INPut:EQUalization:FFEQualizer:NTAPs? <identifier>

Identifier 'INF*.DataIn1'

Description This command sets the number of taps to be used in the FFE algorithm.

This query returns the present setting.

This SCPI is applicable for DSO634A, DSA634A, DSOX96204A and DSAX96204A.

Example :INP:EQU:FFEQ:NTAPs 'INF1.DataIn1',6
:INPut:EQUalization:FFEQualizer:NPRescessor

Syntax: :INPut:EQUalization:FFEQualizer:NPRescessor <identifier>,<number>


Identifier: 'INF*.DataIn1'

Description: This command sets the number of precursor taps to be used in the FFE algorithm.

This query returns the present setting.

This SCPI is applicable for DSO634A, DSA634A, DSOX96204A and DSAX96204A.

Example: :INP:EQU:FFEQ:NPR 'INF1.DataIn1',2

:INPut:EQUalization:FFEQualizer:TAP:AUTO


Identifier: 'INF*.DataIn1'

Description: This command starts the FFE tap optimization. Be sure to first specify the number of taps.

This query returns the present setting.

This SCPI is applicable for DSO634A, DSA634A, DSOX96204A and DSAX96204A.

Example: :INP:EQU:FFEQ:TAP:AUTO 'INF1.DataIn1'


Syntax: :INPut:EQUalization:FFEQualizer:TAP:AUTO:INCLUDe <identifier>,<0|1|OFF|ON>


Identifier: 'INF*.DataIn1'

Description: This command controls if the FFE tap optimization is done during a full sample point alignment. If this is set to OFF, then the tap coefficients must be set manually, or by executing the :INPut:EQUalization:FFEQualizer:TAP:AUTO command.
FFE tap optimization is part of the full sample point optimization by default. Use this command to disable tap optimization in scenarios where the input signal is too distorted to produce reliable tap optimization results.

This query returns the present setting.

This SCPI is applicable for DSO634A, DSA634A, DSOX96204A and DSAX96204A.

Example

```
:INP:EQU:FFEQ:TAP:AUTO:INCL 'INF1.DataIn1', ON
ON
```
SCPI Command Reference

SENSe Subsystem

The SENSe subsystem represents the analyzer's Data In port.
This subsystem has the following commands:

Table 62

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SENSe:SRATe[?]</td>
<td>:SENSe:SRATe[?] on page 276</td>
</tr>
</tbody>
</table>

:SENSe:SRATe[?]  

Syntax :SENSe:SRATe[?] <identifier>, <NRf>  
:SENSe:SRATe? <identifier>

Input Parameters  
'identifier': 'M*.DataIn', 'INF*.DataIn1'

Description  
This command controls the symbol rate for the data input. The symbol rate can only be controlled directly when the clock source is set to CDR. When Clk In or Sys Clk is selected as the clock source, then the symbol rate can only be read.

This query returns the present setting.

This SCPI is applicable for M8046A, DSO634A, DSA634A, DSOX96204A and DSAX96204A.

Example :SENS:SRAT 'M2.DataIn1',10e9
CLOCk Subsystem

The CLOCk subsystem is used to detect clock source or to set the clock frequency applied to input or output channel.

**NOTE**

M8061A has only DataOut with Clock functional block, so the commands are only supported in the DataOut location and not in the DataIn location.

This subsystem has the following SCPI structure:

```
:CLOCk
  [:SOURce][?]  
  :CRECovery:SELect[?]  
  :FREQuency [?]  
  :MULTiplier[?]  
  :TRACk
  [:STATe][?]  
```

This subsystem has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:SOURce][?]</td>
<td>:CLOCk [:SOURce][?] on page 278</td>
</tr>
<tr>
<td>:FREQuency [?]</td>
<td>:CLOCk [:SOURce] :CRECovery:SELect[?] on page 278</td>
</tr>
<tr>
<td>:FREQuency :MULTiplier[?]</td>
<td>:CLOCk :FREQuency :MULTiplier[?] on page 279</td>
</tr>
<tr>
<td>:TRACk [:STATe][?]</td>
<td>:CLOCk :TRACk [:STATe][?] on page 280</td>
</tr>
</tbody>
</table>
The availability of the different clock sources depends on the product being controlled and possible even on installed product options (e.g. CDR). The following are the different clock sources that can be selected:

- **CDR** - Clock Recovery
- **CLK** - Front panel 'CLK IN' connector
- **SYS** - Internal clock generation
- **AUXCLK** - Front panel 'AUX CLK IN' connector
- **ECRecovery** - External clock recovery. Selection is done with the command :CLOCK[:SOURce]:CRECovery:SELect. For details, see :CLOCK[:SOURce]:CRECovery:SELect[?] on page 278.

This query returns the present state.

This SCPI is applicable for M8061A, M8062A and M8046A.

Example:

```
:CLOCK:SOURce 'M3.DataIn', CLK
```

---

### :CLOCK[:SOURce]:CRECovery:SELect[?]

**Syntax**

```
:CLOCK[:SOURce]:CRECovery:SELect 'identifier', <NRf>
```

**Input Parameters**

- 'identifier': 'M*.DataOut' and 'M*.DataIn'
- `<NRf>`: CDR | CLK | SYS | AUXCLK | ECRecovery

**Return Range**

CDR | CLK | SYS | AUXCLK | ECR

**Description**

This command sets the source of the clock signal. It is supported by:

- M8061A
- M8062A
- M8046A

This SCPI is applicable for M8061A, M8062A and M8046A.
Description
This command select the external clock recovery to be used when using the clock source 'ECRecovery'.
This query returns the present state.
This SCPI is applicable for M8046A.
Example
:CLOCK:CREC:SEL 'M2.DataIn', 'DCA1.Slot1'

:CLOCK:FREQuency [?]
Syntax
CLOCK:FREQuency 'Identifier', <NRf>
CLOCK:FREQuency? 'Identifier'
Input
'identifier': 'M*.DataOut' or 'M*.DataIn'
Parameters
Return Range NRf
Description
This command sets the clock frequency applied to the output channel.
This query returns the present state.
This SCPI is applicable for M8061A, M8062A and M8046A.
Example
:CLOCK:FREQ 'M3.DataOut', 8.0e9

:CLOCK:FREQuency:MULTiplier[?]
Syntax
:CLOCK:FREQuency:MULTiplier 'Identifier', <NRf>
:CLOCK:FREQuency:MULTiplier? 'Identifier'
Input
'identifier': 'M*.DataIn'
Parameters
Return Range NRf
Description
This command sets the frequency multiplier that is applied to the externally provided clock in order to achieve the desired symbol rate.
Symbol Rate = Multiplier x Clock Frequency
This query returns the present setting.
This SCPI is applicable for M8046A.
Example
:CLOCK:FREQ:MULT 'M2.DataIn',2
:CLOCk:TRACk[:STATe]? 

Syntax: :CLOCk:TRACk[:STATe] <identifier>, <ON|OFF|1|0> 
CLOCk:FREQuency? 'Identifier'

Parameters: 'identifier': 'M*.DataIn', 'INF*.DataIn1'

Description: This command enables or disables the automatic system symbol rate tracking. When this is enabled, then the system symbol rate is used as the symbol rate of the addressed functional unit (e.g. DataIn). In this case, the required clock frequency will be determined automatically based on the selected frequency multiplier.

See:
- :CLOCk:FREQuency:MULTiplier[?]
- [:SENSe]:SRATe[?]

This query returns the present state.

This SCPI is applicable for M8046A, DSO634A, DSA634A, DSOX96204A and DSAX96204A.

Example: :CLOC:TRAC 'M2.DataIn',ON
N1000 Subsystem

The N1000 subsystem is used to control external clock recovery instruments that are themselves controlled from within the FlexDCA N1000-Series System Software.

**NOTE**

The minimum required version of the FlexDCA N1000-Series System Software is A.05.61.23.

It is possible to share one FlexDCA instance among multiple M8070A instances. However, to ensure correct access control, all M8070A instances have to be executed by the same PC.

All the commands used in this subsystem are basically identical with the SCPI commands of FlexDCA N1000-Series System Software. The only difference is that the M8070A does not use a suffix to address specific parts of the instrument, but instead always has the identifier parameter as first argument of the SCPI commands.

Detailed information about the features and SCPI commands of the supported clock recoveries can be found in the corresponding documentation:

- **N1076A Data Sheet**

- **N1076A User Guide**

- **N1010A FlexDCA Online Help**
This subsystem has the following SCPI structure:

```
:N1000
  :EMODules
    :CONNection?
    :SLOT:MODEL?
  :CRECovery
    :CLBandwidth[?]
    :CRATe [?]
    :LBWMode[?]
    :LOCKed?
    :LSELect [?]
      :AUTomatic[?]
      :AUTO[?]
    :RDIVider[?]
    :RELock
    :SOURce[?]
    :ELEVe[?]
```

This subsystem has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:N1000:CRECovery:CRATe[?]</td>
<td>:N1000:CRECovery:CRATe[?] on page 284</td>
</tr>
</tbody>
</table>
:**N1000:EMODules:SLOT:CONNect**?

**Syntax**

*:N1000:EMODules:SLOT:CONNect <identifier>, {ON | OFF | 1 | 0}*

*:N1000:EMODules:SLOT:CONNect? <identifier>*

**Input Parameters**

- **'identifier':** 'DCA*.SLOT*'
- **ON | OFF | 1 | 0**

**Return Range**

- **ON | OFF | 1 | 0**

**Description**

This command enables usage of the addressed slot by M8070A. In case of using multiple M8070A instances, make sure to enable a slot only in exactly one M8070A instance.

This query returns the present state.

This SCPI is applicable for M8046A.

**Example**

*:N1000:EMOD: SLOT:CONN 'DCA1.SLOT1',ON*

**NOTE**

This command is not the same as the similar FlexDCA command

*:EMODules:SLOT{n}:CONNect? [ON | OFF | 1 | 0]*

The above command must be sent to FlexDCA prior starting up the M8070A software instances that shall control FlexDCA!

Managing the slot usage and connection state must be done using the FlexDCA commands or user interface before starting up M8070A.
**:N1000:EMODules:SLOT:MODel?**

**Syntax**
**:N1000:EMODules:SLOT:MODel? <identifier>**

**Input Parameters**
- `'identifier': 'DCA*.SLOT*'`

**Description**
This query returns the model number of all DCA related modules which are connected to M8070A system software.

This SCPI is applicable for M8046A.

**Example**
**:N1000:EMOD:SLOT:MOD? 'DCA1.SLOT1'**

---

**:N1000:CRECovery:CLBandwidth[?]**

**Syntax**
**:N1000:CRECovery:CLBandwidth <identifier>,<bandwidth>**

**:N1000:CRECovery:CLBandwidth? <identifier>,**

**Input Parameters**
- `'identifier': 'DCA*.SLOT*'`

**Description**
This command sets the fix loop bandwidth.
This query returns the present state.
This SCPI is applicable for M8046A.

**Example**
**:N1000:CREC:CLB 'DCA1.SLOT1',10e6**

For detailed documentation, see :CRECovery{n}:CLBandwidth in the FlexDCA online help.

---

**:N1000:CRECovery:CRATe[?]**

**Syntax**
**:N1000:CRECovery:CRATe <identifier>,<data_rate>**

**:N1000:CRECovery:CRATe? <identifier>**

**Input Parameters**
- `'identifier': 'DCA*.SLOT*'`

**Description**
This command sets the clock recovery's symbol rate.
This query returns the present setting.
This SCPI is applicable for M8046A.
Example :N1000:CREC:CREAT 'DCA1.SLOT1',28e9
For detailed documentation, see :CRECovery{n}:CREATe in the FlexDCA online help.

:N1000:CRECovery:LBWMode[?]
Syntax :N1000:CRECovery:LBWMode <identifier>,{FIXed | RDEPendent}
        :N1000:CRECovery:LBWMode? <identifier>
Input Parameters
'identifier': 'DCA*.SLOT*' 
Description This command sets the loop bandwidth mode to either fixed or symbol rate dependent.
This query returns the present state.
This SCPI is applicable for M8046A.
Example :N1000:CREC:LBWM 'DCA1.SLOT1',RDEP
For detailed documentation, see :CRECovery{n}:LBWMode in the FlexDCA online help.

:N1000:CRECovery:LOCKed?
Syntax :N1000:CRECovery:LOCKed? <identifier>
Input Parameters
'identifier': 'DCA*.SLOT*' 
Description This query displays the clock recovery’s lock status.
Response: 1 (for locked) or 0 (for unlocked)
This SCPI is applicable for M8046A.
Example :N1000:CREC:LOCK? 'DCA1.SLOT1'
For detailed documentation, see :CRECovery{n}:LOCKed the FlexDCA online help.
**:N1000:CREcovery:LSElect[?]**

**Syntax**

```
:N1000:CREcovery:LSElect <identifier>, {LOOP1 | LOOP2 | LOOP3 | LOOP4}
```

```
:N1000:CREcovery:LSElect? <identifier>,
```

**Input Parameters**

- `'identifier': 'DCA*.SLOT*'`

**Description**

This command selects the type 2 loop transition frequency.
This query returns the present state.
This SCPI is applicable for M8046A.

**Example**

```
:N1000:CREC:LSEL 'DCA1.SLOT1', LOOP1
```

For detailed documentation, see :CREcovery{n}:LSElect in the FlexDCA online help.

**:N1000:CREcovery:LSElect:AUTomatic[?]**

**Syntax**

```
:N1000:CREcovery:LSElect:AUTomatic <identifier>, {ON | OFF | 1 | 0}
```

```
:N1000:CREcovery:LSElect:AUTomatic? <identifier>
```

**Input Parameters**

- `'identifier': 'DCA*.SLOT*'`
- `ON | OFF | 1 | 0`

**Description**

This command enables or disables automatic selection of the type 2 loop transition frequency.
This query returns the present state.
This SCPI is applicable for M8046A.

**Example**

```
:N1000:CREC:LSEL:AU'TO 'DCA1.SLOT1', ON
```

For detailed documentation, see :CREcovery{n}:LSElect:AUTO in the FlexDCA online help.
**:N1000:CREC:ODR?[?]**

**Syntax**

```
:N1000:CREC:ODR <identifier>, {SUB32|SUB16|SUB8|SUB4|UNITy|SUPer2|SUPer4|SUPer8}
```

**Input Parameters**

- `identifier`: 'DCA*.SLOT*'
- SUB32 | SUB16 | SUB8 | SUB4 | UNITy | SUPer2 | SUPer4 | SUPer8

**Description**

This command sets the recovered clock's output divide ratio.

This query returns the present state.

This SCPI is applicable for M8046A.

**Example**

```
:N1000:CREC:ODR 'DCA1.SLOT1',SUB2
```

For detailed documentation, see :CREC:ODR in the FlexDCA online help.

**:N1000:CREC:ODR:AUTO?[?]**

**Syntax**

```
:N1000:CREC:ODR:AUTO <identifier>, {ON | OFF | 1 | 0}
```

**Input Parameters**

- `identifier`: 'DCA*.SLOT*'

**Description**

This command enables or disables automatic selection of the recovered clock's output divide ratio.

This query returns the present state.

This SCPI is applicable for M8046A.

**Example**

```
:N1000:CREC:ODR:AUTO 'DCA1.SLOT1',OFF
```

For detailed documentation, see :CREC:ODR:AUTO in the FlexDCA online help.
**:N1000:CREC:RDIVider\[?\]**

**Syntax**

**:N1000:CREC:RDIVider\[?\] <identifier>,<ratio>
**:N1000:CREC:RDIVider? <identifier>,

**Input Parameters**

'identifier': 'DCA*.SLOT**'

**Description**

This command sets loop bandwidth symbol rate divider.

This query returns the present state.

This SCPI is applicable for M8046A.

**Example**

**:N1000:CREC:RDIV 'DCA1.SLOT1',1667

For detailed documentation, see :CREC:RDIVider in the FlexDCA online help.

**:N1000:CREC:RELock**

**Syntax**

**:N1000:CREC:RELock <identifier>

**Input Parameters**

'identifier': 'DCA*.SLOT**'

**Description**

This command initiates a re-lock of the clock recovery.

**NOTE**

The re-lock operation can take quite some time. Especially when the clock recovery cannot lock to the received signal. Therefore, the VISA timeout shall be increased to well above 10 seconds to ensure that the controlling program is waiting long enough for the re-lock operation to finish.

**Example**

**:N1000:CREC:REL 'DCA1.SLOT1';*OPC?

For detailed documentation, see :CREC:RELock in the FlexDCA online help.
### :N1000:CRECovery:SOURce[?]

**Syntax**

```
:N1000:CREC:SOUR <identifier>, {ELECTrical | EINVerted | DIFFerential | AUXilliary}
```

**Input Parameters**

- `'identifier'`: 'DCA*.SLOT*'  
- ELECTrical | EINVerted | DIFFerential | AUXilliary

**Description**

This command selects the input of the clock recovery.

This query returns the present state.

This SCPI is applicable for M8046A.

**Example**

```
:N1000:CREC:SOUR 'DCA1.SLOT1', DIFF
```

For detailed documentation, see :CRECovy{n}:SOURce in the FlexDCA online help.
FETCh Subsystem

The FETCh subsystem is used to query the error detector’s results. Commands for starting, stopping, resetting, etc. are found in the :PLUGin:ERATio subsystem.

This subsystem has the following SCPI structure:

```
:FETCh
  :BCOunter?
  :IBERate?
```

This subsystem has the following commands:

### Table 65

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:BCOunter?</td>
<td>:FETCh:BCOunter? on page 290</td>
</tr>
<tr>
<td>:IBERate?</td>
<td>:FETCh:IBERate? on page 293</td>
</tr>
</tbody>
</table>

**:FETCh:BCOunter?**

**Syntax**

`:FETCh:BCOunter? 'identifier'`

**Input Parameters**

'identifier': "M*.DataIn1", "M*.DataIn2", or "INF*.DataIn1"

**Description**

This query returns the accumulated bit counts and coding specific counts since start of accumulation period. The results are combined to an expression. An expression is comparable to a struct in a programming language. It’s a set of parameter of different types separated be a comma and enclosed by parenthesis.

The returned expression contains a timestamp, the location and a set of bit counter values.

(TimeStamp, "Location", Counted1s, Counted0s, Erroneous1s, Erroneous0s[, coding specific counters])
Timestamp: As timestamp the UNIX epoch time is used; it is defined as elapsed seconds, milliseconds etc. since 01.01.1970 UTC.

Location: It’s a string referred to the place the bit counter are located. This parameter is important if a group name is used as ‘Identifier’.

**Counted1s**: The number of received ones since start of accumulation period.

**Counted0s**: The number of received zeros since start of accumulation period.

**Erroneous1s**: The number of received erroneous ones since start of accumulation period.

**Erroneous0s**: The number of received erroneous zeros since start of accumulation period.

Additional Counters for 8b10b coding:
- ComparedSymbols
- ErroredSymbols
- IllegalSymbols
- WrongDisparity
- FillerSymbols
- ReceivedSymbols
- Frames
- ErroredFrames

Additional Counters for 128/130 and 128/132 coding:
- Blocks
- ErroredBlocks
- IllegalSyncHeaders
- FillerSymbols
- ModifiedFillerSymbols
- Frames
- ErroredFrames
- Additional Counter for 128/132 coding
- CorrectedSyncHeaders
Additional Counters for PAM4 line coding

- Compared Symbols
- Errored Symbols
- Compared Symbol0
- Errored Symbol0
- Compared Symbol1
- Errored Symbol1
- Compared Symbol2
- Errored Symbol2
- Compared Symbol3
- Errored Symbol3

If this query is sent to a ‘group’ of inputs, the response will be expanded by additional sets of counted values. The expression contains for every input belonging to a group a separate set of values starting with the location.

This query returns the present state.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8046A, DSO634A, DSA634A, DSOX96204A and DSAX96204A.

Example:
```
:FETCH:BCO? 'M1.DataIn1'
Response:
(1.366014336e9, 'M1.DataIn1', 1.2222e10, 1.112345e10, 5.1e1, 7.2e2)
```

```
:FETCH:BCO? 'Lanes'
Response: (1.366014336e9, 'M1.DataIn1', 1.2222e10, 1.112345e10, 5.1e1, 7.2e2, 'M1.DataIn2', ...., ...)
```

**NOTE**

Unix time or POSIX time, is a system for describing instances in time, defined as the number of seconds that have elapsed since midnight Coordinated Universal Time (UTC), 1. January 1970, not including leap seconds.
:FETCH:IBERate?

Syntax: :FETCH:IBERate? 'identifier'

Input Parameters:
- 'identifier': 'M*.DataIn1' or 'M*.DataIn2', 'INF*.DataIn1'

Description:
This query returns the instantaneous bit error rate measured by the addressed error detector. The response contains a timestamp, the location and the measured BER.

(TimeStamp, "Location", BER[, coding specific ratios])

Timestamp: As timestamp the Unix epoch time is used; it is defined as elapsed seconds, milliseconds etc., since 01.01.1970 UTC.

Location: It is a string referred to the place the ratios are located. This parameter is important if a group name is used as 'identifier'.

BER: The instantaneous bit error ratio.

Additional counters for 8b10b coding:
- SymbolErrorRatio
- IllegalSymbolRatio
- DisparityErrorRatio
- FillerSymbolRatio
- FrameErrorRatio

Additional Ratios for 128/130 and 128/132 coding:
- BlockErrorRatio
- FillerSymbolRatio
- FrameErrorRatio

Additional Counters for PAM4 line coding:
- Symbol Error Ratio
- Symbol0 Error Ratio
- Symbol1 Error Ratio
- Symbol2 Error Ratio
- Symbol3 Error Ratio

This query returns the present state.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8046A, DSO634A, DSA634A, DSOX96204A and DSAX96204A.
As 'Identifier' a group name is also allowed. The response will be expanded by additional sets of values per error detector included in the group.

**Example**

::FETC:IBER? 'M6.DataIn1'
Response: (1.266014336e9, 'M6.DataIn1', 1.2345e-1)

::FETC:IBER? 'Lanes'
Response: (1.266514336e9, 'M1.DataIn1', 1.2345e-1, 'M1.DataIn2', 0.0e0)
MMEMory Subsystem

All instrument states of the M8020A/M8030A instrument are stored in a "workspace". There are two defined areas in a workspace. One area is named "Factory". In this area you can find read only predefined settings and patterns. The other area is named "User". This area stores customer created settings or patterns.

The "Path" argument is a simple string that can be used to address a setting. This is a simple name that follows the Windows specification for allowed characters of a file name. Additionally, group or folder names can be specified for grouping settings. For example, "SATA/Test1" or "SATA/InitialTests/TestAtTemperature80" and "SATA/InitialTests/TestAtTemperature32" or "PCIeStartUpParameter".

This subsystem has the following SCPI structure:

```
:MMEMory
  :WORKspace
    :SET Ting
      :USER
        :CATalog?
        :STORe
        :RECall
        :DELe te
        :FACTory
          :CATalog?
          :RECall
          :IMPort
        [:USER]
          :CATalog?
          :DELe te
```
This subsystem has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:WORKspace:SETTings:USER:STORe</td>
<td>:MMEMory:WORKspace:SETTings[:USER]:STORe on page 297</td>
</tr>
<tr>
<td>:WORKspace:SETTings:USER:RECall</td>
<td>:MMEMory:WORKspace:SETTings[:USER]:RECall on page 297</td>
</tr>
<tr>
<td>:WORKspace:SETTings:FACTory:RECall</td>
<td>:MMEMory:WORKspace:SETTings:FACTory:RECall on page 298</td>
</tr>
<tr>
<td>:WORKspace:IMPort[:USER]:DELETEte</td>
<td>:MMEMory:WORKspace:IMPort[:USER]:DELETEte on page 300</td>
</tr>
</tbody>
</table>
SCPI Command Reference 5


**Syntax**  

**Description**  
This query returns a list of names currently stored in the 'User' area.

**Example**  

```
'PCI3_InitialTestSetting,DefaultSetting_1,DefaultSetting_2,
DefaultSetting_3,DefaultSetting_4,PCI3/InitialTestA, PCI3/InitialTestB,
PCI3_InitialTestSetting'
```


**Syntax**  

**Input Parameters**  
'Path': Specify name/path to store instrument state.

**Description**  
This command stores the instrument state to the specified location. The 'Path' argument of this command can be a simple string like 'TestConfigurationPCIe3'; the instrument state will then be stored into a location named 'TestConfigurationPCIe3'. In addition, it can specify a group or folder name. This command is meant for grouping settings into one folder (for example, 'PCIe1/TestConfiguration1').

**Example**  


**Syntax**  

**Input Parameters**  
'Path': Specify location of stored instrument state.

**Description**  
This command recalls/restores the instrument state from the specified location.

**Example**  
:MMEMory:WORKspace:SETTings[:USER]:DELeTe
Syntax :MMEMory:WORKspace:SETTings[:USER]:DELeTe 'Path'
Input 'Path': Specify location of stored instrument state.
Parameters
Description This command deletes the instrument state from the specified location.
Example :MMEM:WORK:SETT:DEL 'TestConfigurationSATA_1'
         :MMEM:WORK:SETT:DEL 'SATA/TestConfiguration_1'
         :MMEM:WORK:SETT:DEL 'SATA/TestConfiguration_2'

:MMEMory:WORKspace:SETTings:FACTory:CATalog?
Syntax :MMEMory:WORKspace:SETTings:FACTory:CATalog?
Description This command creates a list of "predefined" setting names located in the "Factory" area of the workspace.
Example :MMEM:WORK:SETT:FACT:CAT? 'PCIe3_InitialTestSetting'

:MMEMory:WORKspace:SETTings:FACTory:RECall
Syntax :MMEMory:WORKspace:SETTings:FACTory:RECall 'Path'
Input 'Path': Specify location of stored instrument state.
Parameters
Description This command recalls/restores the instrument state from a "factory predefined" addressed location.
Example :MMEM:WORKspace:SETTings:FACTory:RECall 'PCIe1/PCIe1_compliance_ASIC'
:MMEMory:WORKspace:IMPort[:USER]

**Syntax**

`MMEMory:WORKspace:IMPort[:USER] <ProfileType>,<"SourcePath">,<"RelativeTargetPath">`

**Input Parameters**

- `<ProfileType>`: Specify jitter sweep or spread spectrum clocking profile type or SPProfile.
- `<"SourcePath">`: Specify the absolute source path of file to be imported.
- `<"RelativeTargetPath">`: Specify a current setting or a shared (common) storage location.

**Description**

An arbitrary profile waveform file can be imported for generating jitter sweep or spread spectrum clocking. Three arguments have to be specified:

- The first one is the profile type JSPRofile | SSCProfile | SPProfile,
- second one is the absolute source path of the file to be imported,
- third one is a relative target path "current:...." addresses the current setting and "shared: ..." addresses a location in the storage that is common for all user settings.

For jitter sweep files, the file extension "...jcs" has to be used and for spread spectrum clocking the extension "...txt".

S-parameter profiles containing1- to 9-port components are supported. The file extension "...s1p...s9p" can be used.

**Example**

- Example : MMEM:WORK:IMP jspr,"c:/temp/Jittersweep.jcs","shared:Jittersweep.jcs"
- Example : MMEM:WORK:IMP jspr,"c:/temp/Jittersweep.jcs","current:Jittersweep.jcs"
- Example : MMEM:WORK:IMP jspr,"c:/temp/Jittersweep.jcs","current:PCIe3.0/Jittersweep.jcs"
- Example : MMEM:WORK:IMP ssp, "c:/temp/INCH7_7_88.s4p", "shared:INCH7_7_88.s4p"
- Example : MMEM:WORK:SSC ssc, "c:/temp/INCH7_7_88.s4p", "shared:INCH7_7_88.s4p"
:MMEMory:WORKspace:IMPort[:USER]:CATalog?

Syntax  :MMEMory:WORKspace:IMPort[:USER]:CATalog? <ProfileType>

Input   <ProfileType>: Specify jitter sweep, spread spectrum clocking and spread

Parameters spectrum profile type i.e. <JSProfile | SSCProfile | SPProfile>.

Description This query generates a list of imported files based on the profile type.


Returns:
"current:Jittersweep.jcs", "shared:Jittersweep.jcs",
"shared:Sweep.jcs", "shared:Test42.jcs"

:MMEMory:WORKspace:IMPort[:USER]:DELete

Syntax  :MMEMory:WORKspace:IMPort[:USER]:DELete

<ProfileType>, <"RelativeTargetPath">

<ProfileType>: Specify jitter sweep, spread spectrum clocking and spread
spectrum profile type i.e. <JSProfile | SSCProfile | SPProfile>.

<"RelativeTargetPath">: Specify a current setting or a shared (common) storage location.

Description Deletes an imported profile waveform file from the specified storage location. First argument is the profile type and the second argument is the relative target path of the file to be deleted.

Example :MMEM:WORK:IMP:DEL jspr, "shared:Jittersweep.jcs"
Example :MMEM:WORK:IMP:DEL jspr, "current:Jittersweep.jcs"
Example :MMEM:WORK:IMP:DEL sscp, "current:deviation.txt"
Example :MMEM:WORK:IMP:DEL sscp, "shared:deviation.txt"
Example :MMEM:WORK:IMP:DEL spp, "shared:INCH7_7_88.s4p"
ACQuisition subsystem

The acquisition subsystem controls how a real-time oscilloscope is being configured in order to capture the signal for BER measurements. This is comparable to setting up the acquisition memory of the oscilloscope, but in terms of bits or symbols, and not in analog samples.

Additionally the acquisition can be stopped in M8070A, so that the oscilloscope becomes free for manual adjustments, while still being under parametric control of M8070A.

This subsystem has the following SCPI structure:

```
:ACQuisition
  ---:BITS[?]
  ---:POINts[:ANALog]?
    ---:REConfigure
  ---:STATe[?]
  ---:SYMBols[?]
```

This subsystem has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:ACQuisition:BITS[?]</td>
<td>:ACQuisition:BITS[?] on page 302</td>
</tr>
<tr>
<td>:ACQuisition:POINts[:ANALog]?</td>
<td>:ACQuisition:POINts[:ANALog]? on page 302</td>
</tr>
<tr>
<td>:ACQuisition:STATe[?]</td>
<td>:ACQuisition:STATe[?] on page 303</td>
</tr>
<tr>
<td>:ACQuisition:SYMBols[?]</td>
<td>:ACQuisition:SYMBols[?] on page 303</td>
</tr>
</tbody>
</table>
:ACQuisition:BITS[?]
Syntax :ACquisision:BITS<identifier>,<number_of_bits>
:ACQuision:BITS? <identifier>
Identifier 'INF*.DataIn1'
Description This command configures the number of bits to capture and process per acquisition. This does directly control the smallest detectable BER value per acquisition, which corresponds to the instantaneous BER result (see :FETCh:IBERate? on page 293).
Increasing the number of bits per acquisition will lower the detectable BER, but at the same time increase the processing duration per acquisition.
This query returns the present setting.
Example :ACQ:BITS 'INF1.DataIn1',1e6

:ACQuisition:POINts[:ANALog]? [?]
Syntax :ACquisision:POINts[:ANALog]? <identifier>
Identifier 'INF*.DataIn1', 'INF*.Common'
Description This query returns the number of analog sample points that are configured into the oscilloscope. This value is for information only, and derives from the configured number of bits per acquisitions and the channel's symbol rate.
Example :ACQ:POIN? 'INF1.DataIn1'
Returns 2.96548e6

:ACQuisition:POINts:REConfigure
Syntax :ACQuision:POINts:REConfigure <identifier>
Identifier 'INF*.DataIn1'
Description This command performs a reset and full reprogramming of the oscilloscope. It effectively re-applies the M8070A settings and resets the manual changes.
Example :ACQ:POIN:REC 'INF1.DataIn1'
[:ACQQuisition:STATe]?
Syntax :ACQQuisition:STATe <identifier>, {OFF | ON | 0 | 1}
         :ACQQuisition:STATe? <identifier>
Identifier 'INF*.Common'
Description This command enables or disables the acquisition. If <identifier> addresses a location that is not a channel, then the acquisition on all channels of the oscilloscope will be controlled by this command. This query returns the present setting.
Example :ACQ:STAT 'INF1.Common',OFF

[:ACQQuisition:SYMBols]?
Syntax :ACQQuisition:SYMBols <identifier>,<number_of_UI>
         :ACQQuisition:SYMBols? <identifier>
Identifier 'INF*.DataIn1'
Description This command configures the number of symbols to capture and process per acquisition. This does directly control the smallest detectable BER value per acquisition, which corresponds to the instantaneous BER result (see :FETCh:IBERate? on page 293).
Increasing the number of symbols per acquisition will lower the detectable BER, but at the same time increase the processing duration per acquisition. In case of NRZ, this command will produce the same results as :ACQ:BITS (see :ACQQuisition:BITS? on page 302), in case of PAM4 is will configure the number of bits to twice the number of symbols.
This query returns the present setting.
Example :ACQ:SYMB 'INF1.DataIn1',1e6
PLUGin Subsystem

The M8020A/M8030A/M8040A platform supports a plugin interface used to implement certain interfaces recognized by the software and integrated into the M8070A GUI and instrument software. Error Ratio, Output Timing, Jitter Tolerance, Output Level, Eye Diagram, Parameter Sweep and Pattern Capture measurements are examples of the plugin concept.

For more information about the plugin interface, refer to PLUGin Subsystem on page 25.

This subsystem has the following SCPI structure:

```scpi
:PLUGin
   :CATalog?
   :RNODEs
   [:LIST]?
   :ERATio
   [...]
   :OTIMing
   [...]
   :OLEVel
   [...]
   :JTOLerance
   [...]
   :EDIagram
   [...]
   :PSWEep
   [...]
   :CCAPture
   [...]
```
This subsystem has the following subnodes:

Table 68

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:RNODEs[:LIST]?</td>
<td>:PLUGin:RNODEs[:LIST]? on page 305</td>
</tr>
<tr>
<td>:OTIMing</td>
<td>:PLUGin:OTIMing Subnode on page 328</td>
</tr>
<tr>
<td>:OLEVel</td>
<td>:PLUGin:OLEVel Subnode on page 359</td>
</tr>
<tr>
<td>:JTOLerance</td>
<td>:PLUGin:JTOLerance Subnode on page 392</td>
</tr>
<tr>
<td>:EDIagram</td>
<td>:PLUGin:EDIagram Subnode on page 423</td>
</tr>
<tr>
<td>:PSWEep</td>
<td>:PLUGin:PSWEep Subnode on page 449</td>
</tr>
<tr>
<td>:CCAPture</td>
<td>:PLUGin:CCAPture Subnode on page 473</td>
</tr>
</tbody>
</table>

**:PLUGin:CATalog?**

**Syntax** :PLUGin:CATalog?

**Description** This query returns the names of all the plug-ins.

**Example**


**:PLUGin:RNODEs[:LIST]?**

**Syntax** :PLUGin:RNODEs[:LIST]?

**Description** This query returns a list of SCPI plugin root nodes that can be used to address different type of measurements (plugins).

**Example**

"CCAPture", "ERATio", "JTOLerance", "OLEVel", "OTIMing", "LTXGKR", "EDIagram", "PSWEep", "OCEI", "CPHYplugin"
:PLUGin:ERATio Subnode

This is the basic measurement for the bit error ratio, symbol error ratio and frame error ratio and is integrated with the M8070A software during the startup. The measurement comes with the set of acquisition and evaluation parameters which are used to configure the measurement run. The acquisition parameters can only be configured before the measurement starts (i.e., you cannot change the acquisition parameter number while the measurement is running); however, the evaluation parameters can be configured at any time, even while the measurement is running.

Results can be retrieved using the :PLUGin:ERATio:FETCh:DATA? or :FETCh subsystem commands.

The M8040A system also supports error uatio measurement using NRZ or PAM4.
This subnode has the following SCPI structure:

```
:PLUGin
  --ERATio
    --ACQuisition
      --:ALOCation[?]
      --:AEND[?]
      --:COMPared[?]
      --:DURation[?]
      --:HISTory[?]
      --:INTerval[?]
      --:TCLevel[?]
      --:TERatio[?]
      --:TIME[?]
    --EVALuation
      --:CERatio[?]
      --:RVMode?
    --FETCh
      --:DATA?
      --:[DURation]?
      --:PFResult?
    --:CATalog?
    --:DElete
    --:NEW
    --:RESet
    --:STARt
    --:STOP
    --:SHOW:SGLegends[?]
    --:SHOW:COMmment[?]
    --:GRAPH?
    --:RUN
      --:HISTory
      --:[STATE][?]
      --:CLEar
      --:LOG?
      --:MESSage?
      --:PROGress?
      --[:STATus]?
```
This subnode has the following commands:

Table 69

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:FETCh:[DURation]?</td>
<td>:PLUGIN:ERATio:FETCh:[DURation]? on page 319</td>
</tr>
<tr>
<td>Name</td>
<td>Description under</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>:DELeTe</td>
<td>:PLUGin:ERATio:DELeTe on page 321</td>
</tr>
<tr>
<td>:NEW</td>
<td>:PLUGin:ERATio:NEW on page 321</td>
</tr>
<tr>
<td>:RESet</td>
<td>:PLUGin:ERATio:RESet on page 322</td>
</tr>
<tr>
<td>:START</td>
<td>:PLUGin:ERATio:START on page 322</td>
</tr>
<tr>
<td>:STOP</td>
<td>:PLUGin:ERATio:STOP on page 322</td>
</tr>
<tr>
<td>:RUN:[STATus]?</td>
<td>:PLUGin:ERATio:RUN:[STATus]? on page 327</td>
</tr>
</tbody>
</table>
:PLUGin:ERATio:ACQuisition:ALOCation[?]

Syntax :

:PLUGin:ERATio:ACQuisition:ALOCation 'identifier', <location-string>

:PLUGin:ERATio:ACQuisition:ALOCation? 'identifier'

Input 'identifier': Specify the measurement name.

Parameters <location-string>: Specify the location identifier or group name identifier.

Return Range <location or group name>

Description This command sets the location identifier (or group name identifier) against which the data acquisition is performed for the specified measurement name identifier. The query returns the current location or group name for the specified measurement name.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example :PLUG:ERAT:ACQ:ALOC 'MyMeasurement','M1.DataIn1'

:PLUGin:ERATio:ACQuisition:AEND[?]

Syntax :


Input 'identifier': Specify the measurement name.

Parameters <PFA|FDUR|CBIT>: Set accumulation end mode.

Return Range PFA|FDUR|CBIT

Description This command sets the accumulation to end when a pass/fail (PFA) condition has been met or end after the full duration (FDUR) or no.of bits (CBIT) of the measurement for the addressed measurement name identifier. The query returns the current accumulation end value.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example :PLUG:ERAT:ACQ:AEND 'MyMeasurement', PFA
:PLUGin:ERATio:ACQuisition:SCRiteria[?]

Syntax
:PLUGin:ERATio:ACQuisition:SCRiteria 'identifier', <PAN|CAN>

Input Parameters
'identifier': Specify the measurement name.

Return Range
PAN|CAN

Description
This command sets the accumulation stop criteria as either "Per Analyzer" or "Combined Analyzer". The combined result will appear in form of Graph, Data Tab and Calculated Results.

- Per Analyzer (PAN) - Graph, Data Tab and combined Calculated Results will not be visible.
- Combined Analyzer (CAN) - Graph, Data Tab and combined Calculated Results will be visible.

NOTE
This command is only valid when a group is selected as an analyzer (If the group is having only one location, this command is not applicable).
This command depends on "Accumulation End". It will not be not applicable if the "Accumulation End" is set as "Full Duration".

The query returns the current accumulation end value.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example
:PLUG:ERAT:ACQ:SCR 'MyMeasurement', CAN

:PLUGin:ERATio:ACQuisition:COMPared[?]

Syntax
:PLUGin:ERATio:ACQuisition:COMPared 'identifier', <NRf>
:PLUGin:ERATio:ACQuisition:COMPared? 'identifier'

Input Parameters
'identifier': Specify the measurement name.

Return Range
IE+6 to IE+18
This command sets the Number of Compared Bits for the addressed measurement name identifier. The query returns the current value. This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example: 
:PLUG:ERAT:ACQ:COMP 'MyMeasurement', IE+8

:PLUGin:ERATio:ACQuisition:DURation[?]

Syntax: 
:PLUGin:ERATio:ACQuisition:DURation 'identifier', <FTIM|IND>

Input Parameters:
'identifier': Specify the measurement name.
<FTIM|IND>: Set accumulation duration mode.

Return Range: FTIM|IND

Description: 
This command sets the accumulation duration mode to either a fixed time (FTIM) or indefinitely (IND) for the addressed measurement name identifier. If FTIM is selected, use the :PLUGin:ERATio:ACQuisition:TIME command to set the duration time. The query returns the current accumulation duration value. This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example: 
:PLUG:ERAT:ACQ:DUR 'MyMeasurement', FTIM

:PLUGin:ERATio:ACQuisition:HISTory[?]

Syntax: 
:PLUGin:ERATio:ACQuisition:HISTory 'identifier', <NRf>

Input Parameters:
'identifier': Specify the measurement name.
<NRf>: Set size of accumulation values.

Return Range: 1 to 100000

Description: 
This command sets the size of accumulation values to be kept in memory for the addressed measurement name identifier. The query returns the current history size. This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.
Example :PLUG:ERAT:ACQ:HIST 'MyMeasurement', 100000

:PLUGin:ERATio:ACQuisition:INTerval[?]

Syntax :PLUGin:ERATio:ACQuisition:INTerval 'identifier', <NRf>

Input Parameters

- 'identifier': Specify the measurement name.
- <NRf>: Set accumulation interval.

Return Range 100E-3 to 2E6

Description This command sets how often to take error ratio samples (from 0.1 to 2000000 seconds) for the addressed measurement name identifier. The query returns the current accumulation interval.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

NOTE Do not use the “Ms” unit with a value. Doing so will cause an error. Instead, use “mas” for the unit. For example, 2E6 is the same as 2 mas.

Example :PLUG:ERAT:ACQ:INT 'MyMeasurement',0.3
**:PLUGin:ERATio:ACQuisition:TCLevel[?]**

**Syntax**: 
`:PLUGin:ERATio:ACQuisition:TCLevel 'identifier', <NRf>`  
`:PLUGin:ERATio:ACQuisition:TCLevel? 'identifier'`

**Input Parameters**
- `'identifier'`: Specify the measurement name.
- `<NRf>`: Set target confidence level.

**Return Range**: 0.1% to 99.9%

**Description**: This command sets the target confidence level when accumulation end is set to pass/fail (in percent) for the addressed measurement name `identifier`. The query returns the current target confidence level.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

**Example**:  
`:PLUG:ERAT:ACQ:TCL 'MyMeasurement', 95`

**:PLUGin:ERATio:ACQuisition:TERatio[?]**

**Syntax**:  
`:PLUGin:ERATio:ACQuisition:TERatio 'identifier', <NRf>`  
`:PLUGin:ERATio:ACQuisition:TERatio? 'identifier'`

**Input Parameters**
- `'identifier'`: Specify the measurement name.
- `<NRf>`: Set target error ratio.

**Return Range**: 1E-18 to 1E-3

**Description**: This command sets the target error ratio for the addressed measurement name `identifier`. The query returns the current target error ratio value.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

**Example**:  
`:PLUG:ERAT:ACQ:TER 'MyMeasurement', 1e-09`
$:PLUGin:ERATio:ACQuisition:TIME[?]

Syntax

$:PLUGin:ERATio:ACQuisition:TIME 'identifier', <NRf>

Input Parameters

'identifier': Specify the measurement name.

<NRf>: Set accumulation fixed time.

Return Range

1 s to 31.5E6 s

Description

This command sets the accumulation fixed time, when accumulation duration is set to FTIM (fixed time), for the addressed measurement name identifier. The query returns the current accumulation fixed time value in seconds.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example

$:PLUG:ERAT:ACQ:TIME 'MyMeasurement',1000

$:PLUGin:ERATio:ACQuisition:PROEditable[?]

Syntax

$:PLUGin:ERATio:ACQuisition:PROEditable 'identifier', <ON|OFF|1|0>
$:PLUGin:ERATio:ACQuisition:PROEditable? 'identifier'

Input Parameters

'identifier': Specify the measurement name.

ON | OFF | 1 | 0

Return Range

0 or 1

Description

This command enables/disables the acquisition parameters of the Parameter Sweep measurement.

This query returns “0” when the acquisition parameters are in disabled state or the measurement is in running state. This query returns “1” when the acquisition parameters are in enabled state.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8046A and all supported real-time oscilloscopes.

NOTE

Do not use the “Ms” unit with a value. Doing so will cause an error. Instead, use “mas” for the unit. For example, 20E6 is the same as 20 mas.
Example

```
:PLUG:ERAT:ACQ:PROE 'MyMeasurement',1
:PLUG:ERAT:ACQ:PROE? 'MyMeasurement'
1
```

### :PLUG:ERAT:SHOW:CERatio?

**Syntax**

```
:PLUGin:ERATio:SHOW:CERatio 'identifier', <ERAT|EZR|EOR>
:PLUGin:ERATio:SHOW:CERatio? 'identifier'
```

**Input Parameters**

- `identifier`: Specify the measurement name.
- `<ERAT|EZR|EOR>`: Set error ratio analysis mode.

**Return Range**

ERAT|EZR|EOR

**Description**

This command sets the error ratio analysis mode to error ratio of all accumulated bits (ERAT), errored zero and counted zero bits (EZR), or errored ones and counted ones bits (EOR) for the addressed measurement name identifier. The query returns the current setting error ratio analysis mode.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example

```
:PLUG:ERAT:SHOW:CER 'MyMeasurement',EOR
```

### :PLUG:ERAT:SHOW:RVMode?

**Syntax**

```
:PLUGin:ERATio:SHOW:RVMode 'identifier', <DET|SUMM>
```

**Input Parameters**

- `identifier`: Specify the measurement name.
- `<DET|SUMM>`: Select measurement results view mode.

**Return Range**

DET|SUMM

**Description**

This command sets the measurement results view mode to detailed or summary when viewing the results using the :PLUG:ERAT:FETCH:DATA? command. This mode can be changed during a measurement or after a measurement has completed. The default is summary mode.
DET (detailed) sets the mode to return the error ratio, compared bits, errored bits, errored zero ratio, compared zeroes, errored zeroes, errored one ratio, compared ones, errored ones, etc.

SUMM (summary) sets the mode to return the error ratio, compared bits, errored bits, etc.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example :PLUG:ERAT:SHOW:RVM 'MyMeasurement', DET

:PLUGin:ERATio:FETCh:DATA?

Syntax :PLUGin:ERATio:FETCh:DATA? 'identifier'[, 'Location']

Input 'identifier': Specify the measurement name.

Parameters

[['Location']]: 'M*.DATAOUT1' or 'M*.DataOut2' (optional).

Return Range 'M*.DATAOUT1' or 'M*.DataOut2' (data results)

Description This command returns the raw data of the error ratio measurement for the addressed measurement name identifier and location identifier. A measurement run on location identifier 'M1.DataIn1', for example, would return results specific to 'M1.DataIn1' only. If the location identifier is omitted and measurement data exists from multiple locations, results will be returned for every location.

The first item in the comma separated list is the location identifier name (for example, 'M1.DataIn1'). The successive values are dependent on the coding, shown in the order below and are repeated in the same order for subsequent measured points.

Bit Error Counter

TimeStamp, ComparedOnes, ComparedZeroes, ErroredOnes, ErroredZeroes, ... 

Counter for 8b/10b Bit Coding

TimeStamp, ComparedOnes, ComparedZeroes, ErroredOnes, ErroredZeroes, ComparedSymbols, ErroredSymbols, IllegalSymbols, WrongDisparity, FillerSymbols, ReceivedSymbols, Frames, ErroredFrames, ...
Counter for 128/130 Bit Coding

TimeStamp, ComparedOnes, ComparedZeroes, ErroredOnes, ErroredZeroes, Blocks, ErroredBlocks, IllegalSynHeaders, FillerSymbols, ModifiedFillerSymbols, Frames, ErroredFrames, ...

Counter for 128/132 Bit Coding

TimeStamp, ComparedOnes, ComparedZeroes, ErroredOnes, ErroredZeroes, Blocks, ErroredBlocks, IllegalSynHeaders, CorrectedSyncHeaders, FillerSymbols, ModifiedFillerSymbols, Frames, ErroredFrames, ...

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example

The following example shows the results returned for 128/130 Bit Coding.

```
:PLUG:ERAT:FETC:DATA? 'Error Ratio 1','M1.DataIn1'
```

Return:

```
('M1.DataIn1*,1420795894.76206,499999811,500000189,0,0,7692307,0,0,0,0,0,0, ...)
```

If the location is omitted from the above example e.g. if the query is

```
:PLUG:ERAT:FETC:DATA? 'Error Ratio 1'
```

then the query returns the comma separated list of all the analyzer location group participating in the measurement.

Return:

```
('M1.DataIn1*,1420795894.76206,499999811,500000189,0,0,7692307,0,0,0,0,0,0, ...),
('M1.DataIn2*,1420795894.76206,499999811,500000189,0,0,7692307,0,0,0,0,0,0, ...)
```

The following example shows the results returned when line coding is selected as PAM4:

```
('M2.DataIn*,1511414904.34319,6064099210500,6064099210500,0,0,60
64099210496,0,1516024802624,0,1516024802624,0,1516024802624,0,1
516024802624,0,...)
```

To get the measurement results in detailed or summary mode, refer to the

```
:PLUGin:ERATio:SHOW:RVMode[?]
```

on page 316.
:PLUGin:ERATio:FETCh[:DURation]?

Syntax :PLUGin:ERATio:FETCh[:DURation]? 'identifier'[,'Location']

Input Parameters
'identifier': Specify the measurement name.

[,'Location']: Specify the location identifier (optional).

Return Range <NR3>

Description This command returns the bit error ratio duration, nOneBits, nZeroBits, nOneBitErrs and nZeroBitErrs of the specified measurement name and location identifier. A measurement run on location 'M1.DataIn1', for example, would return results specific to 'M1.DataIn1' only. If the location identifier is omitted and measurement data exists from multiple locations, results will be returned for every location.

The following example shows the results returned when line coding is selected as PAM4:

('M2.DataIn',1511414904.34319,6064099210500,6064099210500,0,0,60
64099210496,0,1516024802624,0,1516024802624,0,1516024802624,0,1
516024802624,0,...)

To get the measurement results in detailed or summary mode, refer to the :PLUG:ERAT:FETC? on page 316.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example :PLUG:ERAT:FETC? 'MyMeasurement','M1.DataIn1'

:PLUGin:ERATio:FETCh[:DURation]:PFResult?

Syntax :PLUGin:ERATio:FETCh[:DURation]:PFResult? 'identifier'[,'Location']

Input Parameters
'identifier': Specify the measurement name.

[,'Location']: Specify the location identifier (optional).

Return Range <NR3>

Description This command returns the results from the :PLUGin:ERATio:FETCh[:DURation]? query as well as four additional values of the specified measurement name and location identifier:

- ConfidenceLevelAtTargetErrorRatio
- MinimumErrorRatioAtTargetConfidenceLevel
- MaximumErrorRatioAtTargetConfidenceLevel
- PassFailedResult ("UNKNOWN", 'PASS', or 'FAIL' if the AEND parameter is set to 'PassFail')

If the AEND parameter is set to 'FullDuration', the result is the same as PLUGin:ERATio:FETCh[:DURation]? except that the four additional values are set to "Unknown" (not applicable).

A measurement run on location 'M1.DataIn1', for example, would return results specific to 'M1.DataIn1' only. If the location identifier is omitted and measurement data exists from multiple locations, results will be returned for every location.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example: PLUG:ERAT:FETC:PFR? 'MyMeasurement','M1.DataIn1'

:PLUGin:ERATio:CATalog?

Syntax: :PLUGin:ERATio:CATalog?

Description: This command returns a list of all created error ratio measurement names currently available for measuring.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example: Assume the following is a list of created error ratio measurement names:

:PLUG:ERAT:NEW 'ERAT_1'
:PLUG:ERAT:NEW 'ERAT_2'
:PLUG:ERAT:NEW 'ERAT_3'

The command and returned list would look like the following:

:PLUG:ERAT:CAT? 'ERAT_1,ERAT_2,ERAT_3'
:PLUGin:ERATio:NEW

Syntax :PLUGin:ERATio:NEW 'identifier'

Parameters

Input ‘identifier’: Specify the measurement name.

Description This command creates a new error ratio measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example The following example creates an error ratio measurement name identifier called 'MyMeasurement':

:PLUG:ERAT:NEW 'MyMeasurement'

NOTE Creating multiple plugins using this command may slow down the GUI operations which may also result delay in remote programming. To prevent the plugin from opening automatically in the GUI, it is recommended to use "0" as a parameter input in this command.

Example- :PLUG:ERAT:NEW 'MyMeasurement',0

:PLUGin:ERATio:DELete

Syntax :PLUGin:ERATio:DELete 'identifier'

Parameters

Input ‘identifier’: Specify the measurement name to delete.

Description This command deletes a previously created error ratio measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example The following example deletes an error ratio measurement addressed by the measurement name identifier called 'MyMeasurement':

:PLUG:ERAT:DEL 'MyMeasurement'
**:PLUG:ERAT:RES**

**Syntax**

`:PLUG:ERAT:RES 'identifier'`

**Input Parameters**

'identifier': Specify the measurement name to reset.

**Description**

This command resets an error ratio measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

**Example**

The following example resets an error ratio measurement addressed with the measurement name identifier called 'MyMeasurement':

`:PLUG:ERAT:RES 'MyMeasurement'

**:PLUG:ERAT:STA**

**Syntax**

`:PLUG:ERAT:STAR 'identifier'`

**Input Parameters**

'identifier': Specify the measurement name to start.

**Description**

This command starts an error ratio measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

**Example**

The following example starts an error ratio measurement addressed with the measurement name identifier called 'MyMeasurement':

`:PLUG:ERAT:STAR 'MyMeasurement'

**:PLUG:ERAT:STP**

**Syntax**

`:PLUG:ERAT:STOP 'identifier'`

**Input Parameters**

'identifier': Specify the measurement name to stop.

**Description**

This command stops an error ratio measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.
Example: The following example stops an error ratio measurement with the measurement name identifier called 'MyMeasurement':

`:PLUG:ERAT:STOP 'MyMeasurement'

`:PLUGin:ERATio:SHOW:SGLegends[?]`

Syntax: :PLUGin:ERATio:SHOW:SGLegends 'Identifier', <0|1|OFF|ON>

Input Parameters:
- Identifier: 'MyMeasurement'
- <0|1|OFF|ON>: Specify the show/hide state for graph legends.

Return Range: 0|1

Description: This command shows/hides the graph legends.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example: :PLUG:ERAT:SHOW:SGL 'MyMeasurement', 1

`:PLUGin:ERATio:SHOW:COMment[?]`

Syntax: :PLUGin:ERATio:SHOW:COMment 'Identifier', <"Comment">

Input Parameters:
- Identifier: Specify the measurement name. For example; 'MyMeasurement'
- <"Comment">: Enter the desired comment.

Description: This command is used to enter a comment for each measurement. This comment will be displayed on the Measurement History tab.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.


Eye Measurement Example
:PLUGin:ERATio:GRAPh?

Syntax: :PLUGin:ERATio:GRAPh? 'identifier'
                   [ DesiredWidth], DesiredHeight], [ CURR | WHIT], [ PNG], 1 | 0], [ TabName ]

Input Parameters
- 'identifier': Specify measurement name.
- Other optional parameters are:
  - DesiredWidth: Specify the desired width of the image.
  - DesiredHeight: Specify the desired height of the image.
  - CURR | WHIT: Specify whether the user wants to capture in current theme or wants to capture in white background.
  - PNG: Specify the format of the image. The default format is PNG.
  - 1 | 0: Specify whether to capture the graph with legends or not.
  - TabName: Specify the tab name on which the graph is supposed to be captured.

Description: This query returns data of the image captured from the graph view of the plugin in the specified format.

Example: :PLUG:ERAT:GRAP? 'MyMeasurement', 1000, 800, CURR, PNG, 1, 'Graph'

:PLUGin:ERATio:RUN:HISTory[:STATe]?[

Syntax: :PLUGin:ERATio:RUN:HISTory[:STATe] 'identifier' <0|1|ON|OFF>
                   :PLUGin:ERATio:RUN:HISTory[:STATe]? 'identifier'

Input Parameters
- 'identifier': Specify the measurement name.

Return Range: 0|1|ON|OFF

Description: This command enables/disables the storage of error ratio measurement results addressed by the measurement name identifier. This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.
Example
The following example enables storage of error ratio measurement results with the measurement name identifier called 'MyMeasurement':

:PLUG:ERAT:RUN:HIST 'MyMeasurement', 1

:PLUG:ERATio:RUN:HISTory:CLEar

Syntax
:PLUG:ERATio:RUN:HISTory:CLEar 'identifier'

Input
'identifier': Specify the measurement name.

Parameters

Description
This command deletes the storage of error ratio measurement history addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example
The following example deletes error ratio measurement history with the measurement name identifier called 'MyMeasurement':

:PLUG:ERATio:RUN:HISTory:CLEar 'MyMeasurement'

:PLUG:ERATio:RUN:LOG?

Syntax
:PLUG:ERATio:RUN:LOG? 'identifier'

Input
'identifier': Specify the measurement name.

Parameters

Description
This command returns logs for the addressed measurement.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example
The following example returns logs for the measurement name identifier called 'MyMeasurement':

:PLUG:ERATio:RUN:LOG? 'MyMeasurement'

The format of log returned is in the following way:

#XY Message Log

where X denotes the length of digits for Y, Y denotes the number of characters in the log then followed by the log message.
For example:

#10 means, there is 1 digit after 1 and there is 0 character in the log.
#2492 means, there are 3 digits after 2 and there are 492 characters in the log message.

`:PLUGin:ERATio:RUN:MESSage?`

**Syntax**: `:PLUGin:ERATio:RUN:MESSage? 'identifier'`

**Input Parameters**
- `'identifier'`: Specify the measurement name.

**Description**
This query returns a string describing the state of an error ratio measurement addressed by the measurement name identifier. Possible states include NotStarted, Running, Finished, Error, or Stopped.

This SCPI is applicable for M8041A, M8051A, M8061, M8062A and M8046A.

**Example**
The following example returns the state of an error ratio measurement with the measurement name identifier called 'MyMeasurement':

`:PLUG:ERAT:RUN:MESS? 'MyMeasurement'

Running

`:PLUGin:ERATio:RUN:PROGress?`

**Syntax**: `:PLUGin:ERATio:RUN:PROGress? 'identifier'`

**Input Parameters**
- `'identifier'`: Specify measurement name.

**Return Range**
0.0 to 1.0

**Description**
This query returns a number in the range of 0.0 to 1.0 to indicate the progress of an error ratio measurement addressed by the measurement name identifier. A 0.0 indicates that the measurement has not started and 1.0 indicates the measurement is finished.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

**Example**
The following example returns the progress of an error ratio measurement with the measurement name identifier called 'MyMeasurement':

`:PLUG:ERAT:RUN:PROG? 'MyMeasurement'

0.51
`:PLUGin:ERATio:RUN[:STATus]?`

**Syntax**  
`:PLUGin:ERATio:RUN[:STATus]? 'identifier'

**Input Parameters**  
- `'identifier'`: Specify measurement name.

**Return Range**  
0|1

**Description**  
This command returns the running status of an error ratio measurement addressed by the measurement name identifier. A 0 indicates the measurement is not running and a 1 indicates the measurement is running.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

**Example**  
The following example returns the running status of an error ratio measurement with the measurement name identifier called 'MyMeasurement':

`:PLUG:ERAT:RUN 'MyMeasurement', 1`
This is the basic measurement for BER versus the sampling point delay, which is displayed as a 'Bathtub' curve and is integrated with the M8070A software during the startup. The measurement comes with the set of acquisition and evaluation parameters which are used to configure the measurement run. The acquisition parameters can only be configured before the measurement starts (i.e., you cannot change the value of the acquisition parameter while the measurement is running); however, the evaluation parameters can be configured at any time even while the measurement is running.
This subnode has the following SCPI structure:

```
:PULGin
  :OTIMing
    :ACQuisition
    :ALOCation[?]
    :DRESolution[?]
    :FTJBer[?]
    :TLevel[?]
    :OPTimization[?]
    :COMPared[?]
    :EBEnabled[?]
    :ERRored[?]
    :PROEditable[?]
    :EVALuation
    :BERThresh[?]
    :MBJPartition[?]
    :ETJBer[?]
    :ERRType[?]
    :SHOW
    :SMPoints[?]
    :UNIT[?]
    :VSCale[?]
    :SGLegends[?]
    :COMment[?]
    :CATalog?
    :DELete
    :NEW
    :RESet
    :STARt
    :STOP
    :BREak
    :STEP
    :CONTinue
    :GRAPh?
    :RUN
    :HISTory
      [:STATE][?]
    :CLEar
    :LOG?
    :MESSage?
    :PROGress?
    [:STATus]?
  FETCH
  ...
```
This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:ACQuisition:DRESolution[?]</td>
<td>:PLUGin:OTIMing:ACQuisition:DRESolution[?] on page 332</td>
</tr>
<tr>
<td>:ACQuisition:FTJBer[?]</td>
<td>:PLUGin:OTIMing:ACQuisition:FTJBer[?] on page 333</td>
</tr>
<tr>
<td>:ACQuisition:OPTimization[?]</td>
<td>:PLUGin:OTIMing:ACQuisition:OPTimization[?] on page 334</td>
</tr>
<tr>
<td>:ACQuisition:COMPared[?]</td>
<td>:PLUGin:OTIMing:ACQuisition:COMPared[?] on page 334</td>
</tr>
<tr>
<td>:ACQuisition:EBEnabled[?]</td>
<td>:PLUGin:OTIMing:ACQuisition:EBEnabled[?] on page 335</td>
</tr>
<tr>
<td>:ACQuisition:ERRored[?]</td>
<td>:PLUGin:OTIMing:ACQuisition:ERRored[?] on page 335</td>
</tr>
<tr>
<td>:ACQuisition:PROEditable[?]</td>
<td>:PLUGin:OTIMing:ACQuisition:PROEditable[?] on page 336</td>
</tr>
<tr>
<td>:EVALuation:ETJBer[?]</td>
<td>:PLUGin:OTIMing:EVALuation:ETJBer[?] on page 337</td>
</tr>
<tr>
<td>:EVALuation:ERRType[?]</td>
<td>:PLUGin:OTIMing:EVALuation:ERRType[?] on page 338</td>
</tr>
<tr>
<td>Name</td>
<td>Description under</td>
</tr>
<tr>
<td>------</td>
<td>-------------------</td>
</tr>
<tr>
<td>:DELete</td>
<td>:PLUGin:OTIMing:DELete on page 342</td>
</tr>
<tr>
<td>:NEW</td>
<td>:PLUGin:OTIMing:NEW on page 341</td>
</tr>
<tr>
<td>:RESet</td>
<td>:PLUGin:OTIMing:RESet on page 342</td>
</tr>
<tr>
<td>:STARt</td>
<td>:PLUGin:OTIMing:STARt on page 342</td>
</tr>
<tr>
<td>:STOP</td>
<td>:PLUGin:OTIMing:STOP on page 343</td>
</tr>
<tr>
<td>:BREak</td>
<td>:PLUGin:OTIMing:BREak on page 343</td>
</tr>
<tr>
<td>:STEP</td>
<td>:PLUGin:OTIMing:STEP on page 343</td>
</tr>
<tr>
<td>:CONTinue</td>
<td>:PLUGin:OTIMing:CONTinue on page 344</td>
</tr>
<tr>
<td>:RUN[:STATus]?</td>
<td>:PLUGin:OTIMing:RUN[:STATus]? on page 347</td>
</tr>
<tr>
<td>:FETCh</td>
<td>:PLUGin:OTIMing:FETCh Subnode on page 348</td>
</tr>
</tbody>
</table>
**:PLUGin:OTIMing:ACQuisition:ALOCation[?]**

Syntax

**:PLUGin:ERATio:ACQuisition:ALOCation 'identifier', <location-string>**

**:PLUGin:ERATio:ACQuisition:ALOCation? 'identifier'**

Input Parameters

'identifier': Specify the measurement name.

<location-string>: Specify location or group name identifier.

Return Range <location or group name>

Description

This command associates the location or group name identifier with a measurement name identifier against which the data acquisition is performed. The group name identifier is defined using the :SYSTem:INSTrument:GROup:DEFine command.

The query returns the current location or group name identifier against which the measurement name identifier is configured for the data acquisition.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example

**:PLUG:OTIM:ACQ:ALOC 'MyMeasurement','M1.DataIn1'**

**:PLUGin:OTIMing:ACQuisition:DRESolution[?]**

Syntax

**:PLUGin:OTIMing:ACQuisition:DRESolution 'identifier', <NRf>**

**:PLUGin:OTIMing:ACQuisition:DRESolution? 'identifier'**

Input Parameters

'identifier': Specify the measurement name.

<NRf>: Set the sample delay resolution.

Return Range 1 mUI to 250 mUI

Description

This command sets the sample delay resolution in terms of UI and associates it with a specific measurement name identifier. The query returns the current sample delay resolution value in UI.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example

**:PLUG:OTIM:ACQ:DRES 'MyMeasurement', 0.25**
:**PLUGin:OTIMing:ACQuisition:FTJBer**

**Syntax:**
```
:PLUGin:OTIMing:ACQuisition:FTJBer 'identifier', <NRf>
:PLUGin:OTIMing:ACQuisition:FTJBer? 'identifier'
```

**Input Parameters**
- `'identifier'`: Specify the measurement name.
- `<NRf>`: Specify the BER value for the fast total jitter measurement.

**Return Range:** 1E-15 to 1E-9

**Description:**
This command specifies the BER value for the fast total jitter measurement and associates it with a specific measurement name identifier. The query returns the current value at which the fast total jitter measurement was performed.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example:**
```
:PLUG:OTIM:ACQ:FTJB 'MyMeasurement', 1E-10
```

:**PLUGin:OTIMing:ACQuisition:TCLevel**

**Syntax:**
```
:PLUGin:OTIMing:ACQuisition:TCLevel 'identifier', <NRf>
:PLUGin:OTIMing:ACQuisition:TCLevel? 'identifier'
```

**Input Parameters**
- `'identifier'`: Specify the measurement name.
- `<NRf>`: Specify the BER value for the fast total jitter measurement.

**Return Range:** 0.1% to 99.9%

**Description:**
This command sets the target confidence level when sample delay optimization is set as fast total jitter and value is in percentage for the addressed measurement name identifier. The query returns the current target confidence level.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example:**
```
:PLUG:OTIM:ACQ:TCL 'MyMeasurement', 95.1
```

.:PLUGin:OTIM:ACQ:OPTimization[?]

**Syntax**

.:PLUGin:OTIM:ACQ:OPTimization 'identifier', <FTJ|NONE>

.:PLUGin:OTIM:ACQ:OPTimization? 'identifier'

**Input Parameters**

'identifier': Specify the measurement name.

<FTJ|NONE>: Specify the sample delay optimization technique.

**Return Range**

FTJ|NONE

**Description**

This command specifies the sample delay optimization technique and associates it with a specific measurement name identifier. If NONE is selected then no optimization is performed. If FTJ is selected then the fast total jitter measurement is performed. The query returns the current value.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example**

.:PLUG:OTIM:ACQ:OPT 'MyMeasurement', FTJ

.:PLUGin:OTIM:ACQ:COMPared[?]

**Syntax**

.:PLUGin:OTIM:ACQ:COMPared 'identifier', <NRf>

.:PLUGin:OTIM:ACQ:COMPared? 'identifier'

**Input Parameters**

'identifier': Specify the measurement name.

<NRf>: Specify the number of compared bits.

**Return Range**

1E+0 to 1E+18

**Description**

This command specifies the number of compared bits used as criteria for moving to the next measurement point when the sample delay optimization is set to NONE. This command also associates this parameter with a specific measurement name identifier. The query returns the current value.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example**

.:PLUG:OTIM:ACQ:COMP 'MyMeasurement', 1E+7
:PLUGin:OTIMing:ACQuisition:EBEnabled[?]

**Syntax**

:PLUGin:OTIMing:ACQuisition:EBEnabled 'identifier', <OFF|ON|0|1>  
:PLUGin:OTIMing:ACQuisition:EBEnabled? 'identifier'

**Input Parameters**

- 'identifier': Specify the measurement name.
- <OFF|ON|0|1>: Disable/enable errored bits as criteria to move to the next measurement point.

**Return Range**

0|1

**Description**

This command enables/disables the number of errored bits. This command also associates this parameter with a specific measurement name identifier. If enabled (1 or ON), then errored bits will be used along with the number of compared bits as the criteria for moving to the next measurement point when the sample delay optimization is set to NONE. If disabled (0 or OFF), then only the number of compared bits will be used as the criteria for moving to the next measurement point while running the measurement. The query returns the current value.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example**

:PLUG:OTIM:ACQ:EBE 'MyMeasurement', 1

:PLUGin:OTIMing:ACQuisition:ERRored[?]

**Syntax**

:PLUGin:OTIMing:ACQuisition:ERRored 'identifier', <NRf>  
:PLUGin:OTIMing:ACQuisition:ERRored? 'identifier'

**Input Parameters**

- 'identifier': Specify the measurement name.
- <NRf>: Specify the number of errored bits.

**Return Range**

1E+0 to 1E+18

**Description**

This command specifies the number of errored bits. This command also associates this parameter with a specific measurement name identifier. This command can only be used when the :PLUGin:OTIMing:ACQuisition:EBEnabled command is set to 1 (enabled) and the sample delay optimization is set to NONE. This is the criteria for moving to the next measurement point while running the measurement. The query returns the current value.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example**

:PLUG:OTIM:ACQ:ERR 'MyMeasurement', 1E+4
:PLUGin:OTIM:ACQuisition:PROEditable[?]

Syntax
- :PLUGin:OTIM:ACQuisition:PROEditable 'identifier', <ON|OFF|1|0>
- :PLUGin:OTIM:ACQuisition:PROEditable? 'identifier'

Input
- 'identifier': Specify the measurement name.

Parameters
- ON | OFF | 1 | 0

Return Range
- 0 or 1

Description
This command enables/disables the acquisition parameters of the Output Timing measurement.
- This query returns "0" when the acquisition parameters are in disabled state or the measurement is in running state. This query returns "1" when the acquisition parameters are in enabled state.
- This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example
- :PLUG:OTIM:ACQ:PROE 'MyMeasurement', 1
- :PLUG:OTIM:ACQ:PROE? 'MyMeasurement', 1

:PLUGin:OTIMing:EVALuation:BERThresh[?]

Syntax
- :PLUGin:OTIMing:EVALuation:BERThresh 'identifier', <NRf>
- :PLUGin:OTIMing:EVALuation:BERThresh? 'identifier'

Input
- 'identifier': Specify the measurement name.

Parameters
- <NRf>: Specify the BER threshold value.

Return Range
- 1E-15 to 1E-1

Description
This command specifies the BER threshold value used to calculate the measurement parameters and associates it with a specific measurement name identifier. The query returns the current BER threshold value.
- This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example
- :PLUG:OTIM:ACQ:PROE 'MyMeasurement', 1E-10
:PLUGin:OTIMing:EVAluation:MBJPartition[?]
Syntax
:PLUGin:OTIMing:EVAluation:MBJPartition 'identifier', <NRf>
:PLUGin:OTIMing:EVAluation:MBJPartition? 'identifier'
Input
Parameters
'identifier': Specify the measurement name.
<NRf>: Specify the minimum BER for separating random and deterministic jitter components.
Return Range 1E-18 to 1E+0
Description
This command specifies the minimum BER value for separating the random and deterministic jitter components and associates it with a specific measurement name identifier. In order to separate the random and deterministic jitter from the bathtub, the specified BER threshold value must be greater than the value specified by this command. The range is 1E-18 to 1E+0. The query returns the minimum value specified to separate the random and deterministic jitter components. The default is 1E-9.
This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.
Example
:PLUG:OTIM:EVAL:MBJP 'MyMeasurement', 1E-7

:PLUGin:OTIMing:EVAluation:ETJBer[?]
Syntax
:PLUGin:OTIMing:EVAluation:ETJBer 'identifier', <EM6|...|EM12>
:PLUGin:OTIMing:EVAluation:ETJBer? 'identifier'
Input
Parameters
'identifier': Specify the measurement name.
<EM6|...|EM12>: Specify the residual BER for total jitter estimation calculation.
Return Range EM6 to EM12
Description
This command specifies the residual BER for total jitter estimation calculation and associates it with a specific measurement name identifier. The range is 1E-12 (EM12) to 1E-6 (EM6). The query returns the current residual BER for total jitter estimation calculation. The default is 1E-12.
This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.
Example
:PLUG:OTIM:EVAL:ETJB 'MyMeasurement', EM7
=":PLUGin:OTIMing:EVALuation:ERRType[?]

Syntax
":PLUGin:OTIMing:EVALuation:ERRType 'identifier', <AERR|ZERR|OERR>
":PLUGin:OTIMing:EVALuation:ERRType? 'identifier'

Input Parameters
'identifier': Specify the measurement name.

Return Range
AERR|ZERR|OERR

Description
This command specifies the error type to evaluate and associates it with a specific measurement name identifier. AERR allows evaluation of all errors. ZERR allows evaluation of errored zeroes and OERR allows evaluation of errored ones. The query returns the current value.

Example
":PLUG:OTIM:VAL:ERT 'MyMeasurement', ZERR"

=:PLUGin:OTIM:SHOW:SMPoints[?]

Syntax
":PLUGin:OTIMing:SHOW:SMPoints 'identifier', <OFF|ON|0|1>
":PLUGin:OTIMing:SHOW:SMPoints? 'identifier'

Input Parameters
'identifier': Specify the measurement name.

Return Range
0|1

Description
This command disables/enables viewing of the measurement points in the GUI and associates it with a specific measurement name identifier. A 0 or OFF value disables viewing and a 1 or ON value enables viewing of the measurement points. The query returns the current value.

Example
":PLUG:OTIM:SHOW:SMP 'MyMeasurement', 1"
`:PLUGin:OTIM:SHOW:UNIT[?]`


:PLUGin:OTIMing:SHOW:UNIT? 'identifier'

**Input Parameters**

- 'identifier': Specify the measurement name.
- <UINT|TIME>: Specify the timing units.

**Return Range**

UINT|TIME

**Description**

This command specifies the timing units and associates it with a specific measurement name identifier. If UINT is specified then the timing units are in UI. If TIME is specified then the timing units are in seconds. The query returns the current value.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example**

```
:PLUG:OTIM:SHOW:UNIT 'MyMeasurement', unit
```

`:PLUGin:OTIM:SHOW:VSCale[?]`

Syntax: :PLUGin:OTIMing:SHOW:VSCale 'identifier', <LOG|LIN>

:PLUGin:OTIMing:SHOW:VSCale? 'identifier'

**Input Parameters**

- 'identifier': Specify the measurement name.
- <LOG|LIN>: Specify the vertical axis scale.

**Return Range**

LOG|LIN

**Description**

This command specifies the vertical BER axis scale and associates it with a specific measurement name identifier. If LOG is specified then the BER values are in terms of a logarithmic scale. If LIN is specified then the BER values are in terms of a linear scale. The query returns the current value.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example**

```
:PLUG:OTIM:SHOW:VSC 'MyMeasurement', LOG
```
:PLUGin:OTIMing:SHOW:SGLegends?

Syntax :PLUGin:OTIMing:SHOW:SGLegends 'Identifier', <0|1|OFF|ON>
        :PLUGin:OTIMing:SHOW:SGLegends? 'Identifier'

Input Parameters
Identifier: 'MyMeasurement'
<0|1|OFF|ON>: Specify the show/hide state for graph legends.

Return Range 0|1

Description This command shows/hides the graph legends.
This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example :PLUGin:OTIMing:SHOW:SGLegends 'MyMeasurement', 1

:PLUGin:OTIMing:SHOW:COMment?

Syntax :PLUGin:OTIMing:SHOW:COMment 'Identifier', <"Comment">
        :PLUGin:OTIMing:SHOW:COMment? 'Identifier'

Input Parameters
Identifier: Specify the measurement name. For example; 'MyMeasurement'
"<"Comment">": Enter the desired comment.

Description This command is used to enter a comment for each measurement. This comment will be displayed on the Measurement History tab.
This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example :PLUG:OTIM:SHOW:COM 'MyMeasurement', "Output Timing Example"
          :PLUG:OTIM:SHOW:COM? 'MyMeasurement'
Output Timing Example

:PLUGin:OTIMing:CATalog?

Syntax :PLUGin:OTIMing:CATalog?

Description This command returns a list of all created output timing measurement names currently available for measuring.
This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.
Example Assume the following is a list of created output timing measurement names:

```
:PLUG:OTIM:NEW 'OTIM_1'
:PLUG:OTIM:NEW 'OTIM_2'
:PLUG:OTIM:NEW 'OTIM_3'
```

The command and returned list would look like the following:

```
:PLUG:OTIM:CAT?
"OTIM_1","OTIM_2","OTIM_3"
```

`:PLUGin:OTIMing:NEW

Syntax :PLUGin:OTIMing:NEW 'identifier'

Input Parameters

- 'identifier': Specify a new measurement name.

Description This command creates a new output timing measurement name identifier. This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example The following example creates an output timing measurement with the measurement name identifier called 'MyOutputTiming':

```
:PLUG:OTIM:NEW 'MyOutputTiming'
```

NOTE Creating multiple plugins using this command may slow down the GUI operations which may also result in delay in remote programming. To prevent the plugin from opening automatically in the GUI, it is recommended to use "0" as a parameter input in this command.

Example- :PLUG:OTIM:NEW 'MyOutputTiming',0
:PLUGin:OTIMing:DELeTe

Syntax :PLUGin:OTIMing:DELeTe 'identifier'

Input 'identifier': Specify the measurement name to delete.

Parameters

Description This command deletes a previously created output timing measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example The following example deletes an output timing measurement with the measurement name identifier called 'MyOutputTiming':

:PLUG:OTIM:DEL 'MyOutputTiming'

:PLUGin:OTIMing:RESet

Syntax :PLUGin:OTIMing:RESet 'identifier'

Input 'identifier': Specify the measurement name to reset.

Parameters

Description This command resets an output timing measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example The following example resets an output timing measurement addressed with the measurement name identifier called 'MyOutputTiming':

:PLUG:OTIM:RES 'MyOutputTiming'

:PLUGin:OTIMing:STARt

Syntax :PLUGin:OTIMing:STARt 'identifier'

Input 'identifier': Specify the measurement name to start.

Parameters

Description This command starts an output timing measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example The following example starts an output timing measurement with the measurement name identifier called 'MyOutputTiming':

:PLUG:OTIM:STAR 'MyOutputTiming'
**:PLUGin:OTIMing:STOP**

**Syntax**

```
:PLUGin:OTIMing:STOP 'identifier'
```

**Input Parameters**

- **identifier**: Specify the measurement name to stop.

**Description**

This command stops an output timing measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example**

The following example stops an output timing measurement with the measurement name identifier called `MyOutputTiming`:

```
:PLUG:OTIM:STOP 'MyOutputTiming'
```


:**PLUGin:OTIMing:BREak**

**Syntax**

```
:PLUGin:OTIMing:BREak 'identifier'
```

**Input Parameters**

- **identifier**: Specify the measurement name to break/pause.

**Description**

This command breaks/pauses a measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example**

The following example breaks an output timing measurement with the measurement name identifier called `MyOutputTiming`:

```
:PLUG:OTIM:BRE 'MyOutputTiming'
```


:**PLUGin:OTIMing:STEP**

**Syntax**

```
:PLUGin:OTIMing:STEP 'identifier'
```

**Input Parameters**

- **identifier**: Specify the measurement name to step.

**Description**

This command allows you to step through an output timing measurement, addressed by the measurement name identifier, that was stopped using the BREak command.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.
Example

The following example triggers a single output timing measurement step of the measurement addressed with the measurement name identifier called 'MyOutputTiming':

:PLUG:OTIM:STEP 'MyOutputTiming'

:PLUGin:OTIMing:CONTinue

Syntax

:PLUGin:OTIMing:CONTinue 'identifier'

Input Parameters

'identifier': Specify the measurement name to continue.

Description

This command allows you to continue an output timing measurement, addressed by the measurement name identifier, that was stopped using the BReak command.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example

The following example continues a halted output timing measurement addressed with the measurement name identifier called 'MyOutputTiming':

:PLUG:OTIM:CONT 'MyOutputTiming'

:PLUGin:OTIMing:GRAPh?

Syntax

:PLUGin:OTIMing:GRAPh? 'identifier'

[DesiredWidth],[DesiredHeight],[<CURR | WHIT>],[<PNG>],[1 | 0],[,'TabName']

Input Parameters

'identifier': Specify measurement name.

Other optional parameters are:

[DesiredWidth]: Specify the desired width of the image.
[DesiredHeight]: Specify the desired height of the image.
[<CURR | WHIT>]: Specify whether the user wants to capture in current theme or wants to capture in white background.
[<PNG>]: Specify the format of the image. The default format is PNG.
[1 | 0]: Specify whether to capture the graph with legends or not.
['TabName']: Specify the tab name on which the graph is supposed to be captured.
### Description
This query returns data of the image captured from the graph view of the plugin in the specified format.

### Example
`:PLUG:OTIM:GRAP? 'MyMeasurement',1000,800,CURR,PNG,1,'Graph'

### `:PLUGin:OTIMing:RUN:HISTory[:STATE][:?]

**Syntax**
`:PLUGin:OTIMing:RUN:HISTory[:STATE] 'identifier', <0|1|OFF|ON>

**Parameters**
- `'identifier'`: Specify the measurement name.

**Input Range**
0|1

**Description**
This command enables/disables the storage of output timing measurement results addressed by the measurement name identifier. This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example**
The following example enables storage of output timing measurement results with the measurement name identifier called 'MyMeasurement':

`:PLUG:OTIM:RUN:HIST 'MyMeasurement', 1`

### `:PLUGin:OTIMing:RUN:HISTory:CLEar`

**Syntax**
`:PLUGin:OTIMing:RUN:HISTory:CLEar 'identifier'

**Parameters**
- `'identifier'`: Specify the measurement name.

**Description**
This command deletes the output timing measurement history addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example**
The following example deletes the output timing measurement history with the measurement name identifier called 'MyMeasurement':

`:PLUG:OTIM:RUN:HIST:CLE 'MyMeasurement'`
:**PLUGin:OTIMing:RUN:LOG?**

**Syntax**

:PLUGin:OTIMing:RUN:LOG? 'identifier'

**Input Parameters**

'identifier': Specify the measurement name.

**Description**

This command returns logs for the addressed measurement.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example**

This command returns logs for the measurement 'MyMeasurement'.

:PLUG:OTIM:RUN:LOG? 'MyMeasurement'

The format of log returned is in the following way:

```
#XY Message Log
```

where X denotes the length of digits for Y, Y denotes the number of characters in the log then followed by the log message.

For example:

```
#10 means, there is 1 digit after 1 and there is 0 character in the log.
#2492 means, there are 3 digits after 2 and there are 492 characters in the log message.
```

:**PLUGin:OTIMing:RUN:MESSAGE?**

**Syntax**

:PLUGin:OTIMing:RUN:MESSAGE? 'identifier'

**Input Parameters**

'identifier': Specify the measurement name.

**Description**

This command returns a string describing the state of an output timing measurement addressed by the measurement name identifier. Possible states include NotStarted, Running, Suspended, Finished, Error, or Stopped.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example**

The following example returns the state of an output timing measurement with the measurement name identifier called 'MyMeasurement':

:PLUG:OTIM:RUN:MESS? 'MyMeasurement'

Running
### :PLUGin:OTIMing:RUN:PROGress?

**Syntax**
:PLUGin:OTIMing:RUN:PROGress? \('identifier'\)

**Input Parameters**
- \('identifier'\): Specify measurement name.

**Return Range**
0.0 to 1.0

**Description**
This command returns a number in the range of 0.0 to 1.0 to indicate the progress of an output timing measurement addressed by the measurement name identifier. A 0.0 indicates that the measurement has not started and 1.0 indicates the measurement is finished.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example**
The following example returns the progress of an output timing measurement with the measurement name identifier called 'MyMeasurement':

```
:PLUG:OTIM:RUN:PROG? 'MyMeasurement'
0.51
```

### :PLUGin:OTIMing:RUN[:STATus]?

**Syntax**
:PLUGin:OTIMing:RUN[:STATus]? \('identifier'\)

**Input Parameters**
- \('identifier'\): Specify measurement name.

**Return Range**
0 | 1

**Description**
This command returns the running status of an output timing measurement addressed by the measurement name identifier. A 0 indicates the measurement is not running and a 1 indicates the measurement is running.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example**
The following example returns the running status of an output timing measurement with the measurement name identifier called 'MyMeasurement':

```
:PLUG:OTIM:RUN? 'MyMeasurement'
1
```
This subnode has the following SCPI structure:

```
.PLUGIN
  .OTIMing
    .FETCh
      .EDGE
        [:LEFT]
        [:POInts]?
        ::RSQuare?
      .RIGHT
        [:POInts]?
        ::RSQuare?
    [:JITTER]
      [:TOTAL]
      ::DIJitter?
      ::ESTimated?
      ::MEAN?
      ::PTPeak?
      [:RMSQuare]?
      ::RANDom
        [:RMSQuare]?
        ::OSDelay?
        ::PMARgin?
        ::UNCertainty?
      ::DATA?
      ::DELay?
      ::FREQuency?
```
This subnode has the following commands:

### Table 71

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:EDGE[:LEFT][:POINts]?</td>
<td>:PLUGin:OTIMing:FETCh:EDGE[:LEFT][:POINts]? on page 350</td>
</tr>
<tr>
<td>:EDGE:RIGHt[:POINts]?</td>
<td>:PLUGin:OTIMing:FETCh:EDGE:RIGHt[:POINts]? on page 351</td>
</tr>
<tr>
<td>[:JITTer][:TOTal]:DJITTer?</td>
<td>:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:DJITTer? on page 352</td>
</tr>
<tr>
<td>[:JITTer][:TOTal]:ESTimated?</td>
<td>:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:ESTimated? on page 352</td>
</tr>
<tr>
<td>[:JITTer][:TOTal]:MEAN?</td>
<td>:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:MEAN? on page 353</td>
</tr>
<tr>
<td>[:JITTer][:TOTal]:PTPeak?</td>
<td>:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:PTPeak? on page 353</td>
</tr>
<tr>
<td>[:JITTer][:TOTal]:RMSquare?</td>
<td>:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:RMSquare? on page 354</td>
</tr>
<tr>
<td>[:JITTer][:TOTal]:RANDom[RMSquare]?</td>
<td>:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:RANDom[RMSquare]? on page 354</td>
</tr>
<tr>
<td>[:JITTer][:TOTal]:OSDelay?</td>
<td>:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:OSDelay? on page 355</td>
</tr>
<tr>
<td>[:JITTer][:TOTal]:PMARGIN?</td>
<td>:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:PMARGIN? on page 355</td>
</tr>
</tbody>
</table>
:PLUGin:OTIMing:FETCh:EDGE[:LEFT][:POINts]?
Syntax :PLUGin:OTIMing:FETCh:EDGE[:LEFT][:POINts]? 'identifier'[,,'Location']
Input 'identifier': Specify the measurement name.
Parameters ['Location']: Specify location (optional).
Return Range <NR3>
Description This command returns the number of points that have been measured between the BER Threshold and the Min. BER for RJ/DJ separation threshold associated with a specific measurement name identifier. It is calculated for left slope (left edge). This number has to be greater than 2 for the RJ, DJ and estimated TJ values to be applicable.
This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.
A measurement run on a location identifier (for example, 'M1.DataIn1') will return following results.
Example :PLUG:OTIM:FETC:EDGE? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh:EDGE[:LEFT]:RSQuare?
Syntax :PLUGin:OTIMing:FETCh:EDGE[:LEFT]:RSQuare? 'identifier'[,,'Location']
Input 'identifier': Specify the measurement name.
Parameters ['Location']: Specify location (optional).
Return Range <NR3>
Description This command returns the R^2 values calculated for left slope of the bathtub curve (left edge) associated with a specific measurement name identifier. They are a measure of how well the transformed points between BER Threshold and Min. BER for RJ/DJ Separation fit to the linear regression. They have to be greater than 0.75 for the RJ, DJ and estimated TJ values to be applicable.
This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.
A measurement run on a location identifier (for example, 'M1.DataIn1') will return following results.
:PLUGin:OTIMing:FETCh:EDGE:RIGHT[:POINts]?

Syntax: :PLUGin:OTIMing:FETCh:EDGE:RIGHT[:POINts]? 'identifier'[,'Location']

Input Parameters:
- `identifier`: Specify the measurement name.
- `[Location]`: Specify location (optional).

Return Range: <NR3>

Description:
This command returns the number of points that have been measured between the BER Threshold and the Min. BER for RJ/DJ separation threshold associated with a specific measurement name identifier. It is calculated for right slope (right edge). This number has to be greater than 2 for the RJ, DJ and estimated TJ values to be applicable.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following results.

Example:

:PLUGin:OTIMing:FETCh:EDGE:RIGHT:RSQuare?


Input Parameters:
- `identifier`: Specify the measurement name.
- `[Location]`: Specify location (optional).

Return Range: <NR3>

Description:
This command returns the R^2 values calculated for right slope of the bathtub curve (right edge) associated with a specific measurement name identifier. They are a measure of how well the transformed points between BER Threshold and Min. BER for RJ/DJ Separation fit to the linear regression. They have to be greater than 0.75 for the RJ, DJ and estimated TJ values to be applicable.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following results.

Example:
:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:DJITter?

Syntax: :PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:DJITter? 'identifier'[, 'Location']

Input Parameters:
- 'identifier': Specify the measurement name.
- ['Location']: Specify location (optional).

Return Range <NR3>

Description:
This command returns the total jitter component with non-Gaussian distribution (deterministic jitter) associated with a specific measurement name identifier. After transforming a contiguous range of measured points into Q space and performing a linear regression, it is calculated as the period minus the difference between the means of the two straight lines.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.

Example: :


:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:ESTimated?

Syntax: :PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:ESTimated? 'identifier'[, 'Location']

Input Parameters:
- 'identifier': Specify the measurement name.
- ['Location']: Specify location (optional).

Return Range <NR3>

Description:
This command returns an estimate of the expected jitter for very low bit error rates associated with a specific measurement name identifier. After extrapolating the measured BER curves, it is calculated as the period minus the expected width of the eye opening.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.

Example: :

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:MEAN?

Syntax
:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:MEAN? 'identifier'[, 'Location']

Input
'identifier': Specify the measurement name.

Parameters
['Location']: Specify location (optional).

Return Range <NR3>

Description
This command returns the mean value for total jitter associated with a specific measurement name identifier. This value is calculated as the weighted average of the left edge jitter histogram. This command also associates this parameter with a specific measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.

Example

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:PTPeak?

Syntax
:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:PTPeak? 'identifier'[, 'Location']

Input
'identifier': Specify the measurement name.

Parameters
['Location']: Specify location (optional).

Return Range <NR3>

Description
This command returns the peak to peak value for total jitter associated with a specific measurement name identifier. This value is calculated as the pulse period (unit interval) minus the phase margin.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.

Example
:PLUGin:OTIMing:FETCh[:JITTer][:TOTal][:RMSquare]?

Syntax  :PLUGin:OTIMing:FETCh[:JITTer][:TOTal][:RMSquare]?
         'identifier'
         ['Location']

Input  'identifier': Specify the measurement name.

Parameters  ['Location']: Specify location (optional).

Return Range  <NR3>

Description  This command returns the average of the left and right jitter histogram root mean squared values associated with a specific measurement name identifier. This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.

Example  :PLUG:OTIM:FETC? 'MyOutputTiming','M1.DataIn1'

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:RANDom[:RMSquare]?

Syntax  :PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:RANDom[:RMSquare]?
         'identifier'
         ['Location']

Input  'identifier': Specify the measurement name.

Parameters  ['Location']: Specify location (optional).

Return Range  <NR3>

Description  This command returns the total jitter component with Gaussian distribution associated with a specific measurement name identifier. After transforming a contiguous range of measured points into Q space and performing a linear regression, it is calculated as the mean of the sigmas of the two straight lines. The contiguous range is limited by the BER threshold and the minimum BER for RJ/DJ separation threshold. This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:OSDelay?

Syntax :PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:OSDelay? 'identifier'[, 'Location']

Input 'identifier': Specify the measurement name.

Parameters ['Location']: Specify location (optional).

Return Range <NR3>

Description This command returns the average of the left (Ai) and right (Bi) bathtub/BER threshold intersections (optimum sample point delay) associated with a specific measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.


:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:PMARgin?

Syntax :PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:PMARgin? 'identifier'[, 'Location']

Input 'identifier': Specify the measurement name.

Parameters ['Location']: Specify location (optional).

Return Range <NR3>

Description This command returns the period of time where the bit error rate is lower than the BER threshold (phase margin) associated with a specific measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.

:PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:UNCertainty?

Syntax    :PLUGin:OTIMing:FETCh[:JITTer][:TOTal]:UNCertainty?
          'identifier', ['Location']

Input     'identifier': Specify the measurement name.
Parameters ['Location']: Specify location (optional).

Return Range <NR3>

Description This command returns the total jitter uncertainty associated with a specific measurement name identifier. The returned value is only valid if the "Sample Delay Optimization" is set to "Fast Total Jitter" (FTJ) using the :PLUG:OTIM:ACQ:OPT command.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result.


:PLUGin:OTIMing:FETCh:DATA?

Syntax    :PLUGin:OTIMing:FETCh:DATA? 'identifier', ['Location']

Input     'identifier': Specify the measurement name.
Parameters ['Location']: Specify location (optional).

Description This command returns the raw data of the output timing measurement associated with a specific measurement name identifier.

A measurement run on a location identifier (for example, 'M1.DataIn1') will return following results.

This SCPI is applicable for M8041A, M8051A and and M8062A.

Example    :PLUG:OTIM:FETC:DATA? 'Output Timing 1','M1.DataIn1'

"M1.DataIn1",1420788572,96819,499999011,500000989,249999011,250000000,-0.6035,"UNKNOWN", ...

The first item in the comma separated list is the name of the location for which the data is being queried. Rest of the values in the comma separated list for each measurement point are Time Stamp, Compared...
One Bits, Compared Zero Bits, Errored One Bits, Errored Zero Bits, Sample Delay and Pass/Fail Results and are repeated in the same order for subsequent measured points.

In the case of Sample Delay Optimization as "None" the pass/fail result for each measurement point is returned as "UNKNOWN". However if the Fast Total Jitter Measurement is performed then the Pass/Fail results could be either "PASS", "FAIL" or "UNKNOWN"

If the location is omitted from the above example e.g. if the query is :PLUG:OTIM:FETC:DATA? 'Output Timing 1' then the query returns the comma separated list of all the analyzer location group participating in the measurement

```
(("M1.DataIn1",1420788572.96819,499999011,500000989,249999011,250000000,-0.6035,"UNKNOWN", ...),
("M1.DataIn2",1420788572.96819,499999011,500000989,249999011,250000000,-0.6035,"UNKNOWN")))
```

**:PLUGin:OTIMing:FETCh:DELay?**

**Syntax**

`:PLUGin:OTIMing:FETCh:DELay? 'identifier'`

**Input Parameters**

'identifier': Specify the measurement name.

**Description**

This command returns the currently set delay sweep in seconds. The delay sweep is set using the :INPut:DELay command. After setting the value, a data center alignment is executed followed by an output timing measurement.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example**


"M1.DataIn1",4.87924300000001E-009"
`:PLUGIN:OTIMing:FETCh:FREQuency?`

Syntax :PLUGIN:OTIMing:FETCh:FREQuency? 'identifier'

Input  'identifier': Specify the measurement name.

Parameters

Description This command returns the currently set frequency of the measurement. The frequency is set using the :SOURce:FREQuency command.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example :PLUGIN:OTIM:FETCH:FREQ? 'Output Timing 1'

(*M1.DataIn1*, 5.000000000000000E+009)
This is the basic measurement for BER versus sample threshold and is integrated with the M8070A software during the startup. The measurement comes with the set of Acquisition and Evaluation Parameters which are used to configure the measurement run. The Acquisition parameters can only be configured before the measurement starts (i.e. during the measurement run you cannot change the value of Acquisition parameter); however the Evaluation parameters can be configured any time even after the measurement has been run.

This subnode has the following SCPI structure:

```
:OLEVel
  :BREak
  :CATalog?
  :CONTinue
  :DELete
  :NEW
  :RESet
  :STARt
  :STEP
  :STOP
  :GRAPh?
  :ACQuisition
  ....
  :EVALuation
  ....
  :FETCh
  ....
  :SHOW
  ....
  :RUN
  ....
```
This subnode has the following commands and subnodes:

### Table 72

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:BREak[?]</td>
<td>:PLUGin:OLEVel:BREak on page 361</td>
</tr>
<tr>
<td>:CONTinue[?]</td>
<td>:PLUGin:OLEVel:CONTinue on page 361</td>
</tr>
<tr>
<td>:DELete[?]</td>
<td>:PLUGin:OLEVel:DELete on page 362</td>
</tr>
<tr>
<td>:NEW[?]</td>
<td>:PLUGin:OLEVel:NEW on page 362</td>
</tr>
<tr>
<td>:RESet[?]</td>
<td>:PLUGin:OLEVel:RESet on page 363</td>
</tr>
<tr>
<td>:STARt[?]</td>
<td>:PLUGin:OLEVel:STARt on page 363</td>
</tr>
<tr>
<td>:STEP[?]</td>
<td>:PLUGin:OLEVel:STEP on page 363</td>
</tr>
<tr>
<td>:STOP[?]</td>
<td>:PLUGin:OLEVel:STOP on page 364</td>
</tr>
</tbody>
</table>

**Subnodes**

- :ACQuisition: :PLUGin:OLEVel:ACQuisition Subnode on page 365
- :EVALuation     :PLUGin:OLEVel:EVALuation Subnode on page 370
- :FETCH           :PLUGin:OLEVel:FETCH Subnode on page 372
- :SHOW            :PLUGin:OLEVel:SHOW Subnode on page 385
- :RUN             :PLUGin:OLEVel:RUN Subnode on page 388
::PLUGin:OLEVel:BR Eck
Syntax :PLUGin:OLEVel:BR Eck 'identifier'
Input 'identifier': Specify the measurement name to break/pause.
Parameters Description This command breaks/pauses a measurement addressed by the measurement name identifier. This SCPI is applicable for M8041A, M8051A and M8062A.
Example The following example breaks an output level measurement with the measurement name identifier called 'MyMeasurement': ::PLUG:OLEV:BR 'MyMeasurement'

::PLUGin:OLEVel:CATalog?
Syntax :PLUGin:OLEVel:CATalog?
Description This command returns a list of all created output level measurement names currently available for measuring. This SCPI is applicable for M8041A, M8051A and M8062A.
Example Assume three output level measurements named as OLEVel_1, OLEVel_2 and OLEVel_3 are created using NEW command:
::PLUG:OLEVel:NEW 'OLEVel_1'
::PLUG:OLEVel:NEW 'OLEVel_2'
::PLUG:OLEVel:NEW 'OLEVel_3'
The command and returned list would look like the following:
::PLUG:OLEVel:CAT?
"OLEVel_1,OLEVel_2,OLEVel_3"

::PLUGin:OLEVel:CONTinue
Syntax :PLUGin:OLEVel:CONTinue 'identifier'
Input 'identifier': Specify the measurement name to continue.
Parameters Description This command allows you to continue an output level measurement, addressed by the measurement name identifier, which was stopped using the BReak command. This SCPI is applicable for M8041A, M8051A and M8062A.
Example The following example continues a halted output level measurement addressed with the measurement name identifier called 'MyMeasurement': ::PLUG:OLEV:CONT 'MyMeasurement'
**:PLUGin:OLEV:DELe**

**Syntax**

:PLUGin:OLEV:DELe 'identifier'

**Input Parameters**

'identifier': Specify the measurement name to delete.

**Description**

This command deletes a previously created output level measurement addressed by the measurement name identifier. This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**

The following example deletes an output level measurement addressed by the measurement name identifier called 'MyMeasurement':

:PLUG:OLEV:DEL 'MyMeasurement'

---

**:PLUGin:OLEV:NEW**

**Syntax**

:PLUGin:OLEV:NEW 'identifier'

**Input Parameters**

'identifier': Specify the measurement name.

**Description**

This command creates a new output level measurement name identifier. This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**

The following example creates an output level measurement name identifier called 'MyMeasurement':

:PLUG:OLEV:NEW 'MyMeasurement'

---

**NOTE**

Creating multiple plugins using this command may slow down the GUI operations which may also result delay in remote programming. To prevent the plugin from opening automatically in the GUI, it is recommended to use "0" as a parameter input in this command.

Example: :PLUG:OLEV:NEW 'MyMeasurement',0
**:PLUGin:OLEVel:RESet**

**Syntax**

`:PLUGin:OLEVel:RESet 'identifier'`

**Input Parameters**

'identifier': Specify the measurement name to reset.

**Description**

This command resets an output level measurement addressed by the measurement name identifier. This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**

The following example resets an output level measurement addressed with the measurement name identifier called 'MyMeasurement':

`:PLUG:OLEV:RES 'MyMeasurement'`

**:PLUGin:OLEVel:STARt**

**Syntax**

`:PLUGin:OLEVel:STARt 'identifier'`

**Input Parameters**

'identifier': Specify the measurement name to start.

**Description**

This command starts an output level measurement addressed by the measurement name identifier. This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**

The following example starts an output level measurement addressed with the measurement name identifier called 'MyMeasurement':

`:PLUG:OLEV:STAR 'MyMeasurement'`

**:PLUGin:OLEVel:STEP**

**Syntax**

`:PLUGin:OLEVel:STEP 'identifier'`

**Input Parameters**

'identifier': Specify the measurement name to step.

**Description**

This command allows you to step through an output level measurement, addressed by the measurement name identifier, that was stopped using the BReak command. This SCPI is applicable for M8041A, M8051A and M8062A.
Example: The following example triggers a single output level measurement step of the measurement addressed with the measurement name identifier called 'MyMeasurement':
:PLUG:OLEV:STEP 'MyMeasurement'

:PLUGin:OLEVel:STOP

Syntax: :PLUGin:OLEVel:STOP 'identifier'

Input Parameters:

- 'identifier': Specify the measurement name to stop.

Description: This command stops an output level measurement addressed by the measurement name identifier. This SCPI is applicable for M8041A, M8051A and M8062A.

Example: The following example stops an output level measurement with the measurement name identifier called 'MyMeasurement':
:PLUG:OLEV:STOP 'MyMeasurement'

:PLUGin:OLEVel:GRAPh?

Syntax: :PLUGin:OLEVel:GRAPh? 'identifier'

[DesiredWidth],[DesiredHeight],[<CURR | WHIT>],[<PNG>],[1 | 0],[‘TabName’]

Input Parameters:

- 'identifier': Specify measurement name.

Other optional parameters are:

- [DesiredWidth]: Specify the desired width of the image.
- [DesiredHeight]: Specify the desired height of the image.
- [<CURR | WHIT>]: Specify whether the user wants to capture in current theme or wants to capture in white background.
- [<PNG>]: Specify the format of the image. The default format is PNG.
- [1 | 0]: Specify whether to capture the graph with legends or not.
- [‘TabName’]: Specify the tab name on which the graph is supposed to be captured.

Description: This query returns data of the image captured from the graph view of the plugin in the specified format.

Example: :PLUG:OLEV:GRAP? 'MyMeasurement',1000,800,CURR,PNG,1,'Graph'
This subnode has the following SCPI structure:

```
:ACQuisition
  :-ALOCation[?]
  :-COMPared[?]
  :-EBEnabled[?]
  :-ERRored[?]
  :-HILevel[?]
  :-LOWLevel[?]
  :-TRESolution[?]
  :-PROEditable[?]
```

This subnode has the following commands:

### Table 73

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:ALOCation[?]</td>
<td>:PLUGin:OLEVel:ACQuisition:ALOCation[?] on page 366</td>
</tr>
<tr>
<td>:COMPared[?]</td>
<td>:PLUGin:OLEVel:ACQuisition:COMPared[?] on page 366</td>
</tr>
<tr>
<td>:EBEnabled[?]</td>
<td>:PLUGin:OLEVel:ACQuisition:EBEnabled[?] on page 367</td>
</tr>
<tr>
<td>:ERRored[?]</td>
<td>:PLUGin:OLEVel:ACQuisition:ERRored[?] on page 367</td>
</tr>
<tr>
<td>:HILevel[?]</td>
<td>:PLUGin:OLEVel:ACQuisition:HILevel[?] on page 368</td>
</tr>
<tr>
<td>:LOWLevel[?]</td>
<td>:PLUGin:OLEVel:ACQuisition:LOWLevel[?] on page 368</td>
</tr>
<tr>
<td>:TRESolution[?]</td>
<td>:PLUGin:OLEVel:ACQuisition:TRESolution[?] on page 369</td>
</tr>
<tr>
<td>:PROEditable[?]</td>
<td>:PLUGin:OLEVel:ACQuisition:PROEditable[?] on page 369</td>
</tr>
</tbody>
</table>
**:PLUGin:OLEVel:ACQuisition:ALOCation[?]**

**Syntax**

:PLUGin:OLEVel:ACQuisition:ALOCation 'Identifier', <location-string>

:PLUGin:OLEVel:ACQuisition:ALOCation? 'Identifier'

**Input Parameters**

Identifier: 'MyMeasurement'

**Description**

This command sets the Location/Location Group string against which the Data Acquisition is performed. The query returns the current Location/Location group string against which measurement is configured for the Data Acquisition.

This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**

:PLUG:OLEV:ACQ:ALOC 'MyMeasurement','M1.DataIn1'

---

**:PLUGin:OLEVel:ACQuisition:COMPared[?]**

**Syntax**

:PLUGin:OLEVel:ACQuisition:COMPared[?]'Identifier', <NRf>

**Input Parameters**

Identifier: 'MyMeasurement'

**Return Range**

1E+0 to 1E+18

**Description**

This command allows the user to specify the number of compared bits; that would be used as criteria for moving to the next measurement point. The query returns current value.

This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**

:PLUG:OLEV:ACQ:COMP 'MyMeasurement', 1E+7
:PLUGin:OLEV:ACQ:EBEnabled[?]

Syntax: :PLUGin:OLEV:ACQ:EBEnabled[?] 'Identifier', <0 | 1 | ON | OFF>

Input Parameters
Identifier: 'MyMeasurement'

Return Range: 0|1

Description: If enabled then user can specify the number of errored bits additionally with number of compared bits in order to make them criteria together to move to the next measurement point. If this command sets the value as 0 then only number of compared bits are considered to move to the next measurement point while running the measurement. The query returns current value.

This SCPI is applicable for M8041A, M8051A and M8062A.

Example: :PLUG:OLEV:ACQ:EBE 'MyMeasurement', 1

:PLUGin:OLEV:ACQ:ERRored[?]

Syntax: :PLUGin:OLEV:ACQ:ERRored[?] 'Identifier', <NRf>

Input Parameters
Identifier: 'MyMeasurement'

Return Range: 1E+0 to 1E+18

Description: This command allows the user to specify the number of errored bits. This command can only be used when the command PLUGin:OLEV:ACQ:EBEnabled is set to 1. This would be the criteria to move to the next measurement point while the measurement is running. The query returns current value.

This SCPI is applicable for M8041A, M8051A and M8062A.

Example: :PLUG:OLEV:ACQ:ERR'MyMeasurement', 1E+4
`:PLUGin:OLEVel:ACQuisition:HILevel[?]`  
**Syntax:** `:PLUGin:OLEVel:ACQuisition:HILevel[?] 'Identifier', <NRf>`  
**Input Parameters**  
- **Identifier:** 'MyMeasurement'  
**Return Range:** -5 V to 5 V  
**Description:**  
This command allows the user to specify the sample threshold high level. The query returns current value of sample threshold high level.  
This SCPI is applicable for M8041A, M8051A and M8062A.  
**Example:** `:PLUG:OLEV:ACQ:HIL 'MyMeasurement', 0.5`

`:PLUGin:OLEVel:ACQuisition:LOWLevel[?]`  
**Syntax:** `:PLUGin:OLEVel:ACQuisition:LOWLevel[?] 'Identifier', <NRf>`  
**Input Parameters**  
- **Identifier:** 'MyMeasurement'  
**Return Range:** -5 V to 5 V  
**Description:**  
This command allows the user to specify the sample threshold low level. The query returns current value of sample threshold low level.  
This SCPI is applicable for M8041A, M8051A and M8062A.  
**Example:** `:PLUG:OLEV:ACQ:LOWL 'MyMeasurement', -0.5`
`:PLUGin:OLEVel:ACQuisition:TRESolution[?]`

**Syntax**
`:PLUGin:OLEVel:ACQuisition:TRESolution[?] 'Identifier', <NRf>

**Input Parameters**
- **Identifier**: 'MyMeasurement'

**Return Range**
1E-3 to 1

**Description**
This command allows the user to specify the sample threshold resolution in terms of Volts. The query returns current value of sample threshold resolution in terms of Volts.

This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**
`:PLUG:OLEV:ACQ:TRES 'MyMeasurement', 0.001`

---

`:PLUGin:OLEVel:ACQuisition:PROEditable[?]`

**Syntax**
`:PLUGin:OLEVel:ACQuisition:PROEditable 'Identifier', <ON|OFF|1|0>

**Input Parameters**
- **Identifier**: 'MyMeasurement'

**Return Range**
0 or 1

**Description**
This command enables/disables the acquisition parameters of the Output Level measurement.

This query returns “0” when the acquisition parameters are in disabled state or the measurement is in running state. This query returns “1” when the acquisition parameters are in enabled state.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example**
`:PLUG:OLEV:ACQ:PROE 'MyMeasurement', 1

`:PLUG:OLEV:ACQ:PROE? 'MyMeasurement', 1`
:PLUGin:OLEVel:EVALuation Subnode

This subnode has the following SCPI structure:

```
:EVALuation
   :BERTThresh
   :MBQFactor
```

This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:BERTThresh[?]</td>
<td>:PLUGin:OLEVel:EVALuation:BERTThresh[?] on page 370</td>
</tr>
<tr>
<td>:MBQFactor[?]</td>
<td>:PLUGin:OLEVel:EVALuation:MBQFactor[?] on page 371</td>
</tr>
</tbody>
</table>

:PLUGin:OLEVel:EVALuation:BERTThresh[?]

Syntax  
:PLUGin:OLEVel:EVALuation:BERTThresh[?] 'Identifier', <NRf>

Input Parameters
Identifier: 'MyMeasurement'

Return Range 1E-15 to 1E-1

Description This command allows the user to specify the BER Threshold value against which the measurement parameters are calculated. The query returns current value of the BER threshold.

This SCPI is applicable for M8041A, M8051A and M8062A.

Example :PLUG:OLEV:EVAL:BERT 'MyMeasurement', 1E-10
:PLUGin:OLEVel:EVALuation:MBQFactor[?]

Syntax: :PLUGin:OLEVel:EVALuation:MBQFactor[?] 'Identifier', <NRf>

Input Parameters:
- Identifier: 'MyMeasurement'

Range: 1e-15...1.0

Description:
This command allows the user to specify the Minimum BER for Q-factor calculations. Please note that in order to be able to do Q-Factor calculations the specified BER Threshold value must be greater than the value specified by this command.

The query returns the minimum value specified to allow the Q-Factor calculations.

This SCPI is applicable for M8041A, M8051A and M8062A.

Example: :PLUG:OLEV:EVAL:MBQF 'MyMeasurement', 1E-15
This subnode has the following SCPI structure:

```
:FETCh
 --:DATA?
 --:FREQuency?
 --:LEVel
    ...
 --:NOISe
    ...
 --:QLEVel
    ...
```

This subnode has the following commands and subnodes:

Table 75

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:LEVel</td>
<td>:PLUGin:OLEVeI:FETCh:LEVel Subnode on page 374</td>
</tr>
<tr>
<td>:NOISe</td>
<td>:PLUGin:OLEVeI:FETCh:NOISe Subnode on page 377</td>
</tr>
<tr>
<td>:QLEVel</td>
<td>:PLUGin:OLEVeI:FETCh:QLEVel Subnode on page 380</td>
</tr>
</tbody>
</table>
:PLUGin:OLEVel:FETCh:DATA?

Syntax: :PLUGin:OLEVel:FETCh:DATA? 'identifier' [,'Location']

Input Parameters:
- 'identifier': Specify the measurement name.
- [,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description: Return the raw data of the output level measurement. This SCPI is applicable for M8041A, M8051A and M8062A.

Example: A measurement run on location e.g 'M1.DataIn1' would return following results.

:PLUG:OLEV:FETC:DATA? 'MyMeasurement','M1.DataIn1'
Return: ('M1.DataIn1', ...)

The first item in that comma separated list is the name of the location here "M1.DataIn1", the next values are TimeStamp, ComparedOnes, ComparedZeroes, ErroredOnes, ErroredZeroes and SampleThreshold and these 6 values will be repeated.

If the 'Location' is omitted and the measurement runs on multiple locations the response contains all results of current location.

(("M1.DataIn1", ...), ("M1.DataIn2", ...))

:PLUGin:OLEVel:FETCh:FREQuency?

Syntax: :PLUGin:OLEVel:FETCh:FREQuency? 'identifier' [,'Location']

Input Parameters:
- 'identifier': Specify the measurement name.
- [,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description: This command fetches frequency set for the whole module. It also prefixes the acquisition location.

This SCPI is applicable for M8041A, M8051A and M8062A.

Example: :plugin:olev:fetch:frequency? 'MyMeasurement'
This subnode has the following SCPI structure:

```
:LEVEL
  - :AMPLitude?
  - :HIGH?
  - :LOW?
  - :MEAN?
  - :THMargin?
```

This subnode has the following commands:

**Table 76**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
</table>
:PLUGin:OLEVel:FETCh:LEVel:AMPlitude?
Syntax :PLUGin:OLEVel:FETCh:LEVel:AMPlitude? 'identifier' [,'Location']
Input 'identifier': Specify the measurement name.
Parameters [,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).
Description This command returns the value of amplitude calculated after a run.
This SCPI is applicable for M8041A, M8051A and M8062A.
Example :PLUGin:OLEVel:FETCh:LEVel:AMPlitude? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:LEVel:HIGH?
Syntax :PLUGin:OLEVel:FETCh:LEVel:HIGH? 'identifier' [,'Location']
Input 'identifier': Specify the measurement name.
Parameters [,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).
Description This command returns the value of the high level.
This SCPI is applicable for M8041A, M8051A and M8062A.
Example :PLUGin:OLEVel:FETCh:LEVel:HIGH? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:LEVel:LOW?
Syntax :PLUGin:OLEVel:FETCh:LEVel:LOW? 'identifier' [,'Location']
Input 'identifier': Specify the measurement name.
Parameters [,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).
Description This command returns the value of the low level.
This SCPI is applicable for M8041A, M8051A and M8062A.
Example :PLUGin:OLEVel:FETCh:LEVel:LOW? 'MyMeasurement'
:PLUGin:OLEVel:FETCh:LEVel:MEAN?

Syntax  
:PLUGin:OLEVel:FETCh:LEVel:MEAN? 'identifier' ['Location']

Input  
'identifier': Specify the measurement name.

Parameters  
[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description  
This command returns the value of the mean level.

This SCPI is applicable for M8041A, M8051A and M8062A.

Example  
:PLUGin:OLEVel:FETCh:LEVel:MEAN? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:LEVel:THMargin?

Syntax  
:PLUGin:OLEVel:FETCh:LEVel:THMargin? 'identifier' ['Location']

Input  
'identifier': Specify the measurement name.

Parameters  
[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description  
This command returns the value of the threshold margin.

This SCPI is applicable for M8041A, M8051A and M8062A.

Example  
:PLUGin:OLEVel:FETCh:LEVel:THMargin? 'MyMeasurement'
This subnode has the following SCPI structure:

```
:NOISe
  --PPEak?
  --HIGH
    --STDeviation?
  --LOW
    --STDeviation?
  --SNRatio
    --PPEak?
    --RMSquare?
```

This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
</table>
:PLUGin:OLEVel:FETCh:NOISe:PPEak?

Syntax :PLUGin:OLEVel:FETCh:NOISe:PPEak? 'identifier' [, 'Location']

Input

Parameters

Description This command returns the value of the peak to peak noise level.

Example :PLUGin:OLEVel:FETCh:NOISe:PPEak? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:NOISe:HIGH:STDeviation?

Syntax :PLUGin:OLEVel:FETCh:NOISe:HIGH:STDeviation? 'identifier' [, 'Location']

Input

Parameters

Description This command returns the value of the Q High Level Standard Deviation.

Example :PLUGin:OLEVel:FETCh:NOISe:HIGH:STDeviation? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:NOISe:LOW:STDeviation?

Syntax :PLUGin:OLEVel:FETCh:NOISe:LOW:STDeviation? 'identifier' [, 'Location']

Input

Parameters

Description This command returns the value of the Q Low Level Standard Deviation.

Example :PLUGin:OLEVel:FETCh:NOISe:LOW:STDeviation? 'MyMeasurement'
### :PLUGin:OLEVel:FETCh:NOISe:SNRatio:PPEak?

**Syntax:**

```
:PLUGin:OLEVel:FETCh:NOISe:SNRatio:PPEak? 'identifier' [, 'Location']
```

**Input Parameters:**

- `'identifier'`: Specify the measurement name.
- `[,'Location']`: 'M*.DataIn1' or 'M*.DataIn2' (optional).

**Description:**

This command returns the peak to peak value of the signal to noise ratio. This SCPI is applicable for M8041A, M8051A and M8062A.

**Example:**

```
:PLUGin:OLEVel:FETCh:NOISe:SNRatio:PPEak? 'MyMeasurement'
```

### :PLUGin:OLEVel:FETCh:NOISe:SNRatio:RMSquare?

**Syntax:**

```
:PLUGin:OLEVel:FETCh:NOISe:SNRatio:RMSquare? 'identifier' [, 'Location']
```

**Input Parameters:**

- `'identifier'`: Specify the measurement name.
- `[,'Location']`: 'M*.DataIn1' or 'M*.DataIn2' (optional).

**Description:**

This command returns the value of the root mean square of signal to noise ratio. This SCPI is applicable for M8041A, M8051A and M8062A.

**Example:**

```
:PLUGin:OLEVel:FETCh:NOISe:SNRatio:RMSquare? 'MyMeasurement'
```
This subnode has the following SCPI structure:

```
:QLEVEl
  :QFActor?
  :QOThreshold?
  :QRBer?
  :HIGH?
    :POINIts?
    :RSQuare?
    :STDeviation?
  :LOW?
    :POINIts?
    :RSQuare?
    :STDeviation?
```

This subnode has the following commands:

**Table 78**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
</table>
### Name: :PLUGin:OLEVel:FETCh:QLEVel:QFActor?

**Syntax:** :PLUGin:OLEVel:FETCh:QLEVel:QFActor? 'identifier' [, 'Location']

**Input Parameters:**
- 'identifier': Specify the measurement name.
- ['Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

**Description:** This command returns the value of the Q factor.

This SCPI is applicable for M8041A, M8051A and M8062A.

**Example:** :PLUGin:OLEVel:FETCh:QLEVel:QFActor? 'MyMeasurement'

### Name: :PLUGin:OLEVel:FETCh:QLEVel:QOThreshold?

**Syntax:** :PLUGin:OLEVel:FETCh:QLEVel:QOThreshold? 'identifier' [, 'Location']

**Input Parameters:**
- 'identifier': Specify the measurement name.
- ['Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

**Description:** This command returns the value of the Q Optimum threshold.

This SCPI is applicable for M8041A, M8051A and M8062A.

**Example:** :PLUGin:OLEVel:FETCh:QLEVel:QOThreshold? 'MyMeasurement'
:PLUGin:OLEVel:FETCh:QLEVel:QRBer?

Syntax: :PLUGin:OLEVel:FETCh:QLEVel:QRBer? 'identifier' [,'Location']

Input Parameters:
- 'identifier': Specify the measurement name.
- [,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description: This command returns the value of the Q Residual BER.
This SCPI is applicable for M8041A, M8051A and M8062A.

Example: :PLUGin:OLEVel:FETCh:QLEVel:QRBer? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:QLEVel:HIGH?

Syntax: :PLUGin:OLEVel:FETCh:QLEVel:HIGH? 'identifier' [,'Location']

Input Parameters:
- 'identifier': Specify the measurement name.
- [,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description: This command returns the value of the Q High Level.
This SCPI is applicable for M8041A, M8051A and M8062A.

Example: :PLUGin:OLEVel:FETCh:QLEVel:HIGH? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:QLEVel:HIGH:POINts?


Input Parameters:
- 'identifier': Specify the measurement name.
- [,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description: This command returns the value of the Q High Level Points.
This SCPI is applicable for M8041A, M8051A and M8062A.

**:PLUGin:OLEVel:FETCh:QLEVel:HIGH:RSQuare?**

**Syntax**  
**:PLUGin:OLEVel:FETCh:QLEVel:HIGH:RSQuare? 'identifier' [,'Location']

**Input**  
'identifier': Specify the measurement name.

**Parameters**  
[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

**Description**  
This command returns the value of the Q High Level $R^2$.

This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**  
**:PLUGin:OLEVel:FETCh:QLEVel:HIGH:RSQuare? 'MyMeasurement'

**:PLUGin:OLEVel:FETCh:QLEVel:HIGH:STDDeviation?**

**Syntax**  
**:PLUGin:OLEVel:FETCh:QLEVel:HIGH:STDDeviation? 'identifier' [,'Location']

**Input**  
'identifier': Specify the measurement name.

**Parameters**  
[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

**Description**  
This command returns the value of the Q High Level Standard Deviation.

This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**  
**:PLUGin:OLEVel:FETCh:QLEVel:HIGH:STDDeviation? 'MyMeasurement'

**:PLUGin:OLEVel:FETCh:QLEVel:LOW?**

**Syntax**  
**:PLUGin:OLEVel:FETCh:QLEVel:LOW? 'identifier' [,'Location']

**Input**  
'identifier': Specify the measurement name.

**Parameters**  
[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

**Description**  
This command returns the value of the Q Low Level.

This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**  
**:PLUGin:OLEVel:FETCh:QLEVel:LOW? 'MyMeasurement'
:PLUGin:OLEVel:FETCh:QLEVel:LOW:POINts?

Syntax
:PLUGin:OLEVel:FETCh:QLEVel:LOW:POINts? 'identifier' [,'Location']

Input
'identifier': Specify the measurement name.

Parameters
[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description
This command returns the value of the Q Low Level Points.
This SCPI is applicable for M8041A, M8051A and M8062A.

Example
:PLUGin:OLEVel:FETCh:QLEVel:LOW:POINts? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:QLEVel:LOW:RSQuare?

Syntax
:PLUGin:OLEVel:FETCh:QLEVel:LOW:RSQuare? 'identifier' [,'Location']

Input
'identifier': Specify the measurement name.

Parameters
[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description
This command returns the value of the Q Low Level R².
This SCPI is applicable for M8041A, M8051A and M8062A.

Example
:PLUGin:OLEVel:FETCh:QLEVel:LOW:RSQuare? 'MyMeasurement'

:PLUGin:OLEVel:FETCh:QLEVel:LOW:STDeviation?

Syntax
:PLUGin:OLEVel:FETCh:QLEVel:LOW:STDeviation? 'identifier' [,'Location']

Input
'identifier': Specify the measurement name.

Parameters
[,'Location']: 'M*.DataIn1' or 'M*.DataIn2' (optional).

Description
This command returns the value of the Q Low Level Standard Deviation.
This SCPI is applicable for M8041A, M8051A and M8062A.

Example
:PLUGin:OLEVel:FETCh:QLEVel:LOW:STDeviation? 'MyMeasurement'
This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SMPoints[?]</td>
<td>:PLUGin:OLEVel:SHOW:SMPoints[?] on page 386</td>
</tr>
<tr>
<td>:GRAPhics[?]</td>
<td>:PLUGin:OLEVel:SHOW:GRAPhics[?] on page 386</td>
</tr>
<tr>
<td>:SGLegends[?]</td>
<td>:PLUGin:OLEVel:SHOW:SGLegends[?] on page 387</td>
</tr>
<tr>
<td>:COMment[?]</td>
<td>:PLUGin:OLEVel:SHOW:COMment[?] on page 387</td>
</tr>
</tbody>
</table>
:PLUGin:OLEVel:SHOW:SMPoints[?]
Syntax: :PLUGin:OLEVel:SHOW:SMPoints[?] 'Identifier', <0|1|ON|OFF>
Input Parameters
Identifier: 'MyMeasurement'
Description: This command allows the user to control the GUI if he/she wants to visualize the measurement points of the measurement run. Value of 0 indicates that user does not want to visualize the measurement points and 1 indicates that user wants to visualize the measurement points. The query returns current value.
This SCPI is applicable for M8041A, M8051A and M8062A.
Example: :PLUG:OLEV:SHOW:SMP 'MyMeasurement', 1

:PLUGin:OLEVel:SHOW:GRAPhics[?]
Syntax: :PLUGin:OLEVel:SHOW:GRAPhics[?] 'Identifier', <BVTH | QFBV>
Input Parameters
Identifier: 'MyMeasurement'
Description: The command allows the user to specify the Graphics Type to Visualize. If BVTH is used then BER versus Threshold graphics is visualized and the Horizontal Axis Scale (see :PLUGin:OLEVel:SHOW:HSCale[?]) can be visualized in terms of Logarithmic or Linear Scale. If QFBV is used then QBER versus Threshold graphics is visualized and the horizontal axis scale is always linear. The query returns current value.
This SCPI is applicable for M8041A, M8051A and M8062A.
Example: :PLUG:OLEV:SHOW:GRAP 'MyMeasurement', BVTH

:PLUGin:OLEVel:SHOW:HSCale[?]
Syntax: :PLUGin:OLEVel:SHOW:HSCale[?] 'Identifier', <LOG | LIN>
Input Parameters
Identifier: 'MyMeasurement'
Description: The command allows the user to specify the horizontal axis scale (i.e. BER scale) if BER versus Threshold graphics is visualized (see :PLUGin:OLEVel:SHOW:GRAPhics[?]), however if QBER versus threshold
graphics is visualized then the Horizontal Axis Scale is always linear. If LOG is used then the BER values are in terms of Logarithmic scale and if the LIN is used then the BER values are in terms of linear scale. The query returns current value.

This SCPI is applicable for M8041A, M8051A and M8062A.

Example :PLUGin:OLEVel:SHOW:HSC 'MyMeasurement', LOG

:PLUGin:OLEVel:SHOW:SGLegends[?]

Syntax :PLUGin:OLEVel:SHOW:SGLegends 'Identifier', <0|1|OFF|ON>

Input Parameters Identifier: 'MyMeasurement'

Return Range 0|1

Description This command shows/hides the graph legends.

This SCPI is applicable for M8041A, M8051A and M8062A.

Example :PLUGin:OLEVel:SHOW:SGLegends 'MyMeasurement', 1

:PLUGin:OLEVel:SHOW:COMment[?]

Syntax :PLUGin:OLEVel:SHOW:COMment 'Identifier', "<Comment>"

Input Parameters Identifier: Specify the measurement name. For example; 'MyMeasurement'

"<Comment>": Enter the desired comment.

Description This command is used to enter a comment for each measurement. This comment will be displayed on the Measurement History tab.

This SCPI is applicable for M8041A, M8051A and M8062A.

Example :PLUG:OLEVel:SHOW:COMment 'MyMeasurement', "Output Level Example"
This subnode has the following SCPI structure:

```plaintext
:RUN
  :MESSage?
  :PROGress?
  :STATus?
  :HISTORY
    [:STATE]?
    :CLEar
    :LOG?
```

This subnode has the following commands:

**Table 80**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:HISTORY[:STATE]?</td>
<td>:PLUGIN:OLEVel:RUN:HISTORY[:STATE]? on page 390</td>
</tr>
</tbody>
</table>
### :PLUGin:OLEVel:RUN:MESSage?

**Syntax**

```
:PLUGin:OLEVel:RUN:MESSage? 'Identifier'
```

**Input Parameters**

- **Identifier**: 'MyMeasurement'

**Description**

This command returns a string describing the state of an output level measurement addressed by the measurement name identifier. Possible states include NotStarted, Running, Suspended, Finished, Error, or Stopped.

This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**

```
:PLUGin:OLEVel:RUN:MESSage? 'MyMeasurement'
```

### :PLUGin:OLEVel:RUN:PROGress?

**Syntax**

```
:PLUGin:OLEVel:RUN:PROGress? 'Identifier'
```

**Input Parameters**

- **Identifier**: 'MyMeasurement'

**Return Range**

0 to 1

**Description**

This command returns a number in the range of 0.0 to 1.0 to indicate the progress of an output level measurement addressed by the measurement name identifier. A 0.0 indicates that the measurement has not started and 1.0 indicates the measurement is finished.

This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**

```
```

### :PLUGin:OLEVel:RUN:STATus?

**Syntax**

```
:PLUGin:OLEVel:RUN:STATus? 'Identifier'
```

**Input Parameters**

- **Identifier**: 'MyMeasurement'

**Return Range**

0|1

**Description**

This command returns the running status of an output level measurement addressed by the measurement name identifier. A 0 indicates the measurement is not running and a 1 indicates the measurement is running.
This SCPI is applicable for M8041A, M8051A and M8062A.

Example

```plaintext
:PLUG:OLEV:RUN:STATus? 'MyMeasurement'

:PLUGin:OLEVel:RUN:HISTory[:STATe][?]?
```

Syntax :PLUGin:OLEVel:RUN:HISTory[:STATe] 'Identifier', <0|1|OFF|ON>

Input Parameters

Identifier: 'MyMeasurement'

Return Range 0|1

Description This command enables/disables the storage of output level measurement results addressed by the measurement name identifier. This query returns the current setting. This SCPI is applicable for M8041A, M8051A and M8062A.

Example

```plaintext
:PLUGin:OLEVel:RUN:HISTory[:STATe] 'MyMeasurement',1

:PLUGin:OLEVel:RUN:HISTory:CLEar
```

Syntax :PLUGin:OLEVel:RUN:HISTory:CLEar 'Identifier'

Input Parameters

Identifier: 'MyMeasurement'

Description This command deletes the output level measurement history addressed by the measurement name identifier. This SCPI is applicable for M8041A, M8051A and M8062A.

Example

```plaintext
:PLUGin:OLEVel:RUN:HISTory:CLEar 'MyMeasurement'
```
:PLUGin:OLEVel:RUN:LOG?

Syntax: :PLUGin:OLEVel:RUN:LOG? 'Identifier'

Input Parameters:
Identifier: 'MyMeasurement'

Description:
This command returns logs for the addressed measurement.
This SCPI is applicable for M8041A, M8051A and M8062A.

Example: :PLUGin:OLEVel:RUN:LOG? 'MyMeasurement'

Output:
The format of log returned is in the following way:

#XY Message Log

where X denotes the length of digits for Y, Y denotes the number of characters in the log then followed by the log message.

For example:
#10 means, there is 1 digit after 1 and there is 0 character in the log.
#2492 means, there are 3 digits after 2 and there are 492 characters in the log message.
The jitter tolerance commands are used to determine the capacity of a serial data receiver to tolerate jitter. There are two modes: compliance or characterization. Jitter frequency is set at multiple test points. At each point, the jitter amplitude is modulated using various search algorithms to determine the level of jitter which causes the device to fail. The M8040A system supports also jitter tolerance measurement using NRZ or PAM4.

This subnode has the following SCPI structure:

```scpi
:PLUGin:JTOLerance Subnode

The jitter tolerance commands are used to determine the capacity of a serial data receiver to tolerate jitter. There are two modes: compliance or characterization. Jitter frequency is set at multiple test points. At each point, the jitter amplitude is modulated using various search algorithms to determine the level of jitter which causes the device to fail. The M8040A system supports also jitter tolerance measurement using NRZ or PAM4.

This subnode has the following SCPI structure:
This subnode has the following commands:

### Table 81

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:BSETup:TBERatio[?]</td>
<td>:PLUGin:JTOLeance:BSETup:TBERatio[?] on page 397</td>
</tr>
<tr>
<td>:BSETup:CLEVel[?]</td>
<td>:PLUGin:JTOLeance:BSETup:CLEVel[?] on page 397</td>
</tr>
<tr>
<td>:BSETup:FRTime[?]</td>
<td>:PLUGin:JTOLeance:BSETup:FRTime[?] on page 398</td>
</tr>
<tr>
<td>:BSETup:ARTime[?]</td>
<td>:PLUGin:JTOLeance:BSETup:ARTime[?] on page 398</td>
</tr>
<tr>
<td>Name</td>
<td>Description under</td>
</tr>
<tr>
<td>------</td>
<td>-------------------</td>
</tr>
<tr>
<td>.MSETup:TEMPLatE:FILE:FRESult</td>
<td>:PLUGin:JTOLerance:.MSETup:TEMPLatE:FILE:FRESult on page 400</td>
</tr>
<tr>
<td>.MSETup:FREQuency:STARt[?]</td>
<td>:PLUGin:JTOLerance:.MSETup:FREQuency:STARt[?] on page 402</td>
</tr>
<tr>
<td>.MSETup:MTPoints[?]</td>
<td>:PLUGin:JTOLerance:.MSETup:MTPoints[?] on page 403</td>
</tr>
<tr>
<td>.MSETup:NPoInts[?]</td>
<td>:PLUGin:JTOLerance:.MSETup:NPoInts[?] on page 403</td>
</tr>
<tr>
<td>.MSETup:MODE[?]</td>
<td>:PLUGin:JTOLerance:.MSETup:MODE[?] on page 403</td>
</tr>
<tr>
<td>.MSETup:CMARgin[?]</td>
<td>:PLUGin:JTOLerance:.MSETup:CMARgin[?] on page 404</td>
</tr>
<tr>
<td>.MSETup:ALGorithm[?]</td>
<td>:PLUGin:JTOLerance:.MSETup:ALGorithm[?] on page 405</td>
</tr>
<tr>
<td>.MSETup:BINary:SSIZe[?]</td>
<td>:PLUGin:JTOLerance:.MSETup:BINary:SSIZe[?] on page 408</td>
</tr>
<tr>
<td>.MSETup:LINear:SSIZe[?]</td>
<td>:PLUGin:JTOLerance:.MSETup:LINear:SSIZe[?] on page 408</td>
</tr>
<tr>
<td>.MSETup:LOG:SSIZe[?]</td>
<td>:PLUGin:JTOLerance:.MSETup:LOG:SSIZe[?] on page 409</td>
</tr>
<tr>
<td>.MSETup:CLAuto[?]</td>
<td>:PLUGin:JTOLerance:.MSETup:CLAuto[?] on page 410</td>
</tr>
<tr>
<td>.GRAPh?</td>
<td>:PLUGin:JTOLerance:.GRAPh? on page 410</td>
</tr>
<tr>
<td>.GRAPh:TLIMits[?]</td>
<td>:PLUGin:JTOLerance:.GRAPh:TLIMits[?] on page 411</td>
</tr>
<tr>
<td>.GRAPh:TPOInts[?]</td>
<td>:PLUGin:JTOLerance:.GRAPh:TPOInts[?] on page 411</td>
</tr>
<tr>
<td>.GRAPh:CLIMits[?]</td>
<td>:PLUGin:JTOLerance:.GRAPh:CLIMits[?] on page 412</td>
</tr>
<tr>
<td>.GRAPh:APOInts[?]</td>
<td>:PLUGin:JTOLerance:.GRAPh:APOInts[?] on page 412</td>
</tr>
<tr>
<td>.FETCh:DATA?</td>
<td>:PLUGin:JTOLerance:.FETCh:DATA[?]:ALL on page 412</td>
</tr>
<tr>
<td>Name</td>
<td>Description under</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>:BREak</td>
<td>PLUGin:JTOLerance:BREak on page 414</td>
</tr>
<tr>
<td>:CONTinue</td>
<td>PLUGin:JTOLerance:CONTinue on page 415</td>
</tr>
<tr>
<td>:NEW</td>
<td>PLUGin:JTOLerance:NEW on page 416</td>
</tr>
<tr>
<td>:DELete</td>
<td>PLUGin:JTOLerance:DELete on page 416</td>
</tr>
<tr>
<td>:RESet</td>
<td>PLUGin:JTOLerance:RESet on page 417</td>
</tr>
<tr>
<td>:START</td>
<td>PLUGin:JTOLerance:START on page 417</td>
</tr>
<tr>
<td>:STOP</td>
<td>PLUGin:JTOLerance:STOP on page 417</td>
</tr>
<tr>
<td>:STEP</td>
<td>PLUGin:JTOLerance:STEP on page 418</td>
</tr>
<tr>
<td>:SHOW:COMMENT[?]</td>
<td>PLUGin:JTOLerance:SHOW:COMMENT[?] on page 421</td>
</tr>
<tr>
<td>:ACQuisition:PROEdita[-]</td>
<td>PLUGin:JTOLerance:ACQuisition:PROEdita[-] on page 422</td>
</tr>
</tbody>
</table>
:PLUGin:JTOlerance:INSTruments:GENerator[?]

Syntax
:PLUGin:JTOlerance:INSTruments:GENerator 'identifier',<location-string>

Input
'identifier': Specify the measurement name.

Parameters
<location-string>: Specify the location identifier or group name identifier.

Return Range <location or group name>

Description
This command is used to select the pattern generator module/channel for the jitter tolerance measurement. The location-string refers to either a location name or a group name identifier associated with a measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example
:PLUG:JTOL:INST:GEN 'MyMeasurement','M1.DataOut1'

:PLUGin:JTOlerance:INSTruments:ANALyzer[?]

Syntax
:PLUGin:JTOlerance:INSTruments:ANALyzer 'identifier',<location-string>

Input
'identifier': Specify the measurement name.

Parameters
<location-string>: Specify the location identifier or group name identifier.

Return Range <location or group name>

Description
This command is used to select the analyzer (error detector) module/channel for the jitter tolerance measurement. The location-string refers to either a location name or a group name identifier associated with a measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example
:PLUG:JTOL:INST:ANAL 'MyMeasurement','M1.DataIn1'
:PLUGin:JTOLerance:BSETup:TBERatio[?]

Syntax
:PLUGin:JTOLerance:BSETup:TBERatio 'identifier', <NRf>
:PLUGin:JTOLerance:BSETup:TBERatio? 'identifier'

Input Parameters
'identifier': Specify the measurement name.

Input Parameters
<NRF>: Set the target BER.

Return Range 1E-15 to 1E-1

Description
This command is used to set the target BER of the measurement specified by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example
:PLUG:JTOL:BSET:TBER 'MyMeasurement', 1E-15

:PLUGin:JTOLerance:BSETup:CLEVel[?]

Syntax
:PLUGin:JTOLerance:BSETup:CLEVel 'identifier', <NRf>
:PLUGin:JTOLerance:BSETup:CLEVel? 'identifier'

Input Parameters
'identifier': Specify the measurement name.

Input Parameters
<NRF>: Set the percent confidence interval.

Return Range 0.1 to 99.9

Description
This command is used to set the percent confidence interval of the measurement specified by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example
:PLUG:JTOL:BSET:CLEV 'MyMeasurement', 95%
:PLUGin:JTOLerance:BSETup:FRTime[?]

Syntax

:PLUGin:JTOLerance:BSETup:FRTime 'identifier', <NRf>
:PLUGin:JTOLerance:BSETup:FRTime? 'identifier'

Input Parameters

'identifier': Specify the measurement name.

<NRF>: Set the frequency relax time.

Return Range 0 ms to 60 s

Description

This command is used to set the frequency relax time after a change in jitter modulation frequency for the measurement specified by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example

:PLUG:JTOL:BSET:FRT 'MyMeasurement',30s

:PLUGin:JTOLerance:BSETup:ARTime[?]

Syntax

:PLUGin:JTOLerance:BSETup:ARTime 'identifier', <NRf>
:PLUGin:JTOLerance:BSETup:ARTime? 'identifier'

Input Parameters

'identifier': Specify the measurement name.

<NRF>: Set the amplitude relax time.

Return Range 0 ms to 60 s

Description

This command is used to set the amplitude relax time after a change in jitter modulation amplitude for the measurement specified by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example

:PLUG:JTOL:BSET:ART 'MyMeasurement',30s
:PLUGin:JTOlarance:MSETup:TEMPlate:FILE

Syntax :PLUGin:JTOlarance:MSETup:TEMPlate:FILE 'identifier', <filepath-string>

Input Parameters

'identifier': Specify the measurement name.

<filepath-string>: Specify path/filename of template to load.

Return Range Path/filename of current measurement template.

Description This command is used to specify the path/filename of the measurement template to load for the measurement specified by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example :PLUGin:JTOlarance:MSETup:TEMPlate:FILE "MyMeasurement","Shared/USB_1.jtt"

:PLUGin:JTOlarance:MSETup:TEMPlate:FILE:DEFine

Syntax :PLUGin:JTOlarance:MSETup:TEMPlate:FILE:DEFine "Identifier","templateName","templatePoints"

Input Parameters

'identifier': Specify the measurement name.

<filepath-string>: Specify path/filename of template to load.

Return Range Path/filename of current measurement template.

Description This command creates a template file and selects it immediately in the given jitter tolerance measurement.

The jitter amplitude values are automatically limited to a minimum of 0.1mUI.

templateName: Specifies the name and location of the template file. Structure is "Shared|Current:name.jtt" alternatively "Shared|Current/name.jtt"

Examples:

"Shared:template1.jtt"

"Shared/template1.jtt"

"Current:template2.jtt"

"Current/template2.jtt"
Template Points

A list of tuples, containing values for frequency, min limit, compliance value and max limit with the following structure:

(tuple, tuple, ...)

- Each tuple has the following structure:

  (frequencyHz, minLimitUI, complianceLevelUI, maxLimitUI)

Example: \(((100, 500, 750, 1000), (1000, 1, 10, 500), (500, 0.05, 1, 1))\)

This defines the following points:

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Minimum Limit</th>
<th>Compliance Value</th>
<th>Maximum Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 Hz</td>
<td>500 UI</td>
<td>750 UI</td>
<td>1000 UI</td>
</tr>
<tr>
<td>100 kHz</td>
<td>1 UI</td>
<td>10 UI</td>
<td>500 UI</td>
</tr>
<tr>
<td>500 MHz</td>
<td>1 mUI</td>
<td>1 UI</td>
<td>1 UI</td>
</tr>
</tbody>
</table>

The frequency value are clipped to a resolution of 1Hz and the jitter amplitudes are clipped to a minimum value of 0.1mUI.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example:

```plaintext
:PLUGin:JTOLerance:MSETup:TEMPlate:FILE:DEFine "jitter tolerance 1", "Shared:template.jtt", "((100,500,750,1000), (1000,1,10,500), (50000, 0.05, 1, 1000))"
```

**:PLUGin:JTOLerance:MSETup:TEMPlate:FILE:FRESult**

Syntax:

```plaintext
:PLUGin:JTOLerance:MSETup:TEMPlate:FILE:FRESult "Identifier", "templateName", [[[minMarginPct], minMarginUI], maxMarginPct], maxMarginUI]
```

Description:

This command creates a template file and immediately selects it in the given jitter tolerance measurement.

The results of the last measurement run define points in the template file.

If a jitter frequency has not resulted in a passed point then the template will be set to:
If the jitter frequency has resulted in a passed point, then the template will be set to:

<table>
<thead>
<tr>
<th>Minimum</th>
<th>The smaller value of</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Maximum passed jitter amplitude) * minMarginPct/100</td>
<td></td>
</tr>
<tr>
<td>(Maximum passed jitter amplitude) – abs(minMarginUI)</td>
<td></td>
</tr>
</tbody>
</table>

| Confidence Level | Maximum passed jitter amplitude |

<table>
<thead>
<tr>
<th>Maximum</th>
<th>The larger value of</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Maximum passed jitter amplitude) * maxMarginPct/100</td>
<td></td>
</tr>
<tr>
<td>(Maximum passed jitter amplitude) + abs(maxMarginUI)</td>
<td></td>
</tr>
</tbody>
</table>

Defaults and value ranges for the optional arguments are:

<table>
<thead>
<tr>
<th></th>
<th>Default</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>minMarginPct</td>
<td>50</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>minMarginUI</td>
<td>0.25</td>
<td>0</td>
<td>10000</td>
</tr>
<tr>
<td>maxMarginPct</td>
<td>110</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>maxMarginUI</td>
<td>0.25</td>
<td>0</td>
<td>10000</td>
</tr>
</tbody>
</table>

The resulting limit values will be clipped to a minimum of 0.1 mUI.
:PLUGin:JTOLerance:MSETup:FREQuency:STARt

Syntax  :PLUGin:JTOLerance:MSETup:FREQuency:STARt 'identifier', <NRf>
        :PLUGin:JTOLerance:MSETup:FREQuency:STARt? 'identifier'

Parameters
'identifier': Specify the measurement name.
<Nrf>: Set the start frequency.

Return Range 100 Hz to 500 MHz

Description This command is used to set the start (minimum) modulation frequency for
the measurement specified by the measurement name identifier. The
default is 100 kHz.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and
M8046A.

Example :PLUG:JTOL:MSET:FREQ:STAR 'MyMeasurement', 10kHz

:PLUGin:JTOLerance:MSETup:FREQuency:STOp

Syntax  :PLUGin:JTOLerance:MSETup:FREQuency:STOp 'identifier', <NRf>

Parameters
'identifier': Specify the measurement name.
<Nrf>: Set the stop frequency.

Return Range 100 Hz to 500 MHz

Description This command is used to set the stop (maximum) modulation frequency for
the measurement specified by the measurement name identifier. The
default is 100 MHz.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and
M8046A.

Example :PLUG:JTOL:MSET:FREQ:STO 'MyMeasurement', 100MHz
### :PLUGin:JTOLerance:MSETup:MTPoints[?]

**Syntax:**
```
:PLUGin:JTOLerance:MSETup:MTPoints[?] 'identifier', <NRf>
```

**Input Parameters**
- `'identifier'`: Specify the measurement name.

**Return Range:**
0 | 1 OFF | ON

**Description:**
This command selects either template points or specified range.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

**Example:**
```
:PLUGin:JTOLerance:MSETup:MTPoints 'MyMeasurement'
```

### :PLUGin:JTOLerance:MSETup:NPOints[?]

**Syntax:**
```
:PLUGin:JTOLerance:MSETup:NPOints 'identifier', <NRf>
```

**Input Parameters**
- `'identifier'`: Specify the measurement name.
- `<NRf>`: Set the number of measurement points.

**Return Range:**
1 to 100

**Description:**
This command is used to set the number of measurement points for the measurement specified by the measurement name identifier.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

**Example:**
```
:PLUG:JTOL:MSET:NPO 'MyMeasurement', 50
```

### :PLUGin:JTOLerance:MSETup:MODE[?]

**Syntax:**
```
:PLUGin:JTOLerance:MSETup:MODE 'identifier', <COMPliance|CHARacterize>
```

**Input Parameters**
- `'identifier'`: Specify the measurement name.
- `<COMPliance|CHARacterize>`: Specify the measurement mode.

**Return Range:**
COMPliance|CHARacterize
This command is used to specify compliance or characterization as the measurement mode for the measurement specified by the measurement name identifier. Compliance is used for checking that a device passes the minimum margin required for compliance to a standard. Characterization tests a device for tolerance to jitter to establish the pass/fail amplitude at each measurement point.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example:

```plaintext
:PLUG:JTOL:MSET:MODE 'MyMeasurement', CHAR
```

### :PLUGin:JTOLerance:MSETup:CMARgin[?]

**Syntax**: 

```plaintext
:PLUGin:JTOLerance:MSETup:CMARgin 'identifier', <NRf>
:PLUGin:JTOLerance:MSETup:CMARgin? 'identifier'
```

**Input Parameters**

- `'identifier'`: Specify the measurement name.
- `<NRf>`: Specify the compliance margin.

**Return Range**: -75% to 1000%

**Description**: This command is used to specify the compliance margin for the measurement specified by the measurement name identifier. The default is 0%.

Compliance is used for checking that a device passes the minimum margin required for compliance to a standard. Characterization tests a device for tolerance to jitter to establish the pass/fail amplitude at each measurement point.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example:

```plaintext
:PLUG:JTOL:MSET:CMAR 'MyMeasurement', 10
```
:PLUGin:JTOLerance:MSETup:ALGorithm[?]

Syntax :PLUGin:JTOLerance:MSETup:ALGorithm 'identifier',
<BINary|BDLinear|BULinear|DLINear|DLOGarithmic|ULINear|
ULOGarithmic|ULL|ABIN|AdaptiveBINary|CUSTom>
:PLUGin:JTOLerance:MSETup:ALGorithm? 'identifier'

Input Parameters

'identifier': Specify the measurement name.

Return Range BINary|BDLinear|BULinear|DLINear|DLOGarithmic|ULINear|ULOGarithmic|ULL|ABIN|AdaptiveBINary|CUSTom

Description This command is used to specify the measurement algorithm during characterization measurements for the measurement specified by the measurement name identifier.

**BINary**: The Binary algorithm uses a variable step size to find the highest passing SJ amplitude at each SJ frequency in the template. Starting at the maximum SJ amplitude determined by the upper limit in the template file or the instrument limit, whichever is lower, followed by the minimum SJ amplitude determined by the lower limit in the template file, the Binary algorithm then sets subsequent SJ amplitudes at the logarithmic midpoint between the highest passing and lowest failing SJ amplitudes. A test point is determined to be a passing point if the BER measured is below the BER threshold set by the user.

**BDLinear**: The Binary + Down Linear algorithm is the same as the Binary algorithm followed by the Down Linear algorithm. Refer to the descriptions for Binary and Down Linear.

**BULinear**: The Binary + Up Linear algorithm is the same as the Binary algorithm followed by the Up Linear algorithm. Refer to the descriptions for Binary and Up Linear.

**DLINear**: The Down Linear search algorithm starts at the maximum jitter value determined by the template.

If a BER measurement has errors above the measurement threshold, the jitter amplitude is adjusted lower by the linear step size.

This algorithm exits when the BER is measured below the measurement threshold BER, or if the last amplitude measurement point was at the minimum jitter value in the template.
The jitter tolerance result at each jitter modulation frequency is the highest jitter amplitude at which the measurement was made below the threshold BER. However, if the device is intolerant of even the minimum jitter value, the result is not valid and is not plotted.

**DLOGarithmic:** The Down Logarithmic search algorithm starts at the maximum jitter value determined by the template.

If a BER measurement has errors above the measurement threshold, the jitter amplitude is adjusted lower by the coefficient calculated from the logarithmic step size. For example, if the coefficient is 10%, then the next amplitude equals the previous amplitude minus 10% of the previous amplitude.

This algorithm exits when the BER is measured below the measurement threshold BER, or if the last amplitude measurement point was at the minimum jitter value in the template.

The jitter tolerance result at each jitter modulation frequency is the highest jitter amplitude at which the measurement was made below the threshold BER. However, if the device is intolerant of even the minimum jitter value, the result is not valid and is not plotted.

**ULINear:** The Up Linear search algorithm starts at the minimum jitter value determined by the template.

If a BER measurement has errors below the measurement threshold, the jitter amplitude is adjusted higher by the linear step size.

This algorithm exits when BER is measured above the measurement threshold BER, or if the last amplitude measurement point was at the maximum jitter value in the template.

The jitter tolerance result at each jitter modulation frequency is the highest jitter amplitude at which the measurement was made below the threshold BER. However, if the device is intolerant of even the minimum jitter value, the result is not valid and is not plotted.

**ULOGarithmic:** The Up Logarithmic search algorithm starts at the minimum jitter value determined by the template.

If a BER measurement has errors below the measurement threshold, the jitter amplitude is adjusted higher by the coefficient calculated from the logarithmic step size. For example, if the coefficient is 10%, then the next amplitude equals the previous amplitude plus 10% of the previous amplitude.
This algorithm exits when the BER is measured above the measurement threshold BER, or if the last amplitude measurement point was at the maximum jitter value determined by the template.

**Up Log + Linear:** The Up Log + Linear search algorithm performs the same algorithm as Up Logarithmic, but in addition, it returns to the last passing amplitude and steps linearly up until it reaches a fail point. The algorithm continues to increase the SJ amplitude until failure, even if it surpassed the original failed point.

The jitter tolerance result at each jitter modulation frequency is the highest jitter amplitude at which the measurement was made below the threshold BER. However, if the device is intolerant of even the minimum jitter value, the result is not valid and is not plotted.

**Adaptive Binary:** The Adaptive Binary search algorithm is a mix of the Up-Logarithmic and Binary and works best when the jitter frequency is being swept from high to low frequencies. It will increase/decrease the jitter amplitude using the logarithmic step size until it finds a pass/fail transition. Then reduce the step size until it becomes less than the defined binary step size. The minimum value of the active template and the maximum passed amplitude of the previous jitter frequency define the initial jitter amplitude.

**Custom:** The Custom search algorithm requires the DUT Control Interface license (M8070A-1TP, M8070A-1NP or M8070A-1TT) and allow to define the jitter amplitude by implementing 3 functions in the DUT Control Interface.

The tutorial script 'JTOL-Custom-Algorithm-with-Virtual-DUT.py' demonstrates how to implement the algorithm correctly. This script is a part of the M8070A installation.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example: :PLUG:JTOL:MSET:ALG 'MyMeasurement', BIN
.:PLUGin:JTOLerance:MSETup:BIANary:SSIIZe[?]

Syntax

.:PLUGin:JTOLerance:MSETup:BIANary:SSIIZe 'identifier', <NRf>
.:PLUGin:JTOLerance:MSETup:BIANary:SSIIZe? 'identifier'

Input

'identifier': Specify the measurement name.

Parameters

<NRf>: Set the step size.

Return Range 2 mUI to 5 UI

Description

This command is used to set the step size for the binary algorithm for the measurement specified by the measurement name identifier. The step size defines the exit criteria for the algorithm. The Binary search algorithm stops once its step size falls below this user defined Step Size.

The Binary Step Size applies to the Binary algorithm and the binary portions of the Binary + Down Linear and Binary + Up Linear algorithms.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example

.:PLUG:JTOL:MSET:BIN:SSIIZ 'MyMeasurement', 100mUI

.:PLUGin:JTOLerance:MSETup:LIINear:SSIIZe[?]

Syntax

.:PLUGin:JTOLerance:MSETup:LIINear:SSIIZe 'identifier', <NRf>
.:PLUGin:JTOLerance:MSETup:LIINear:SSIIZe? 'identifier'

Input

'identifier': Specify the measurement name.

Parameters

<NRf>: Set the step size.

Return Range 5 mUI to 100 UI

Description

This command is used to set the step size for each step of the Up Linear and Down Linear algorithms for the measurement specified by the measurement name identifier.

When Down Linear is enabled as the search algorithm, the measurement starts from the maximum (which depends on the jitter frequency). A step size of 100 mUI, for example, may result in a sequence of 100 UI, 99.9 UI, 99.8 UI and so on. The test for one frequency stops when the BER limit is met or zero amplitude is reached.

When Up Linear is enabled as the search algorithm, the measurement starts from the minimum. A step size of 100 mUI, for example, will result in
a sequence of 0.1 UI, 0.2 UI, 0.3 UI and so on. The test for one frequency stops when the BER limit is exceeded or the maximum amplitude is reached.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example :PLUG:JTOL:MSET:LIN:SSIZe 'MyMeasurement', 100mUI

:PLUGin:JTOLerance:MSETup:LOG:SSIZe[?]

Syntax :PLUGin:JTOLerance:MSETup:LOG:SSIZe 'identifier', <NRf>

:PLUGin:JTOLerance:MSETup:LOG:SSIZe? 'identifier'

Input Parameters

'identifier': Specify the measurement name.

<NRf>: Set the step size.

Return Range 0.1% to 50%

Description This command is used to set the step size for each step of the Up Logarithmic and Down Logarithmic algorithms for the measurement specified by the measurement name identifier.

When Down Logarithmic is enabled as the search algorithm, the measurement starts from the maximum (which depends on the jitter frequency). A percentage of 50%, for example, may result in a sequence of 100 UI, 50 UI, 25 UI and so on. The test for one frequency stops when the BER limit is met or the specified minimum amplitude is reached.

When Up Logarithmic is enabled as the search algorithm, the measurement starts from the specified minimum amplitude. A percentage of 50%, for example, may result in a sequence of 0.1 UI, 0.15 UI, 0.23 UI and so on. The test for one frequency stops when the BER limit is crossed or the maximum amplitude (which depends on the jitter frequency) is reached.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example :PLUG:JTOL:MSET:LOG:SSIZe 'MyMeasurement', 25
:PLUGin:JTOLerance:MSETup:CLAuto[?]

Syntax
:PLUGin:JTOLerance:MSETup:CLAuto 'identifier', <ON|OFF|1|0>
:PLUGin:JTOLerance:MSETup:CLAuto? 'identifier'

Input Parameters
'identifier': 'Specify the measurement name.
<ON|OFF|1|0>: Enable/disable auto loop bandwidth.

Return Range 1|0

Description
This command sets the external CDR loop bandwidth selection during a Jitter Tolerance measurement. If AUTO state is ON, will set the CDR LBW based on the jitter frequency. If AUTO state is OFF, CDR LBW is not altered during the measurement. The query returns the current auto mode state.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example
:PLUG:JTOL:MSET:CLA 'MyMeasurement',ON

:PLUGin:JTOLerance:GRAPh?

Syntax
:PLUGin:JTOLerance:GRAPh? 'identifier'

[DesiredWidth],[DesiredHeight],[<CURR | WHIT>],[<PNG>],[1 | 0],[,'TabName']

Input Parameters
'identifier': Specify measurement name.
Other optional parameters are:
[DesiredWidth]: Specify the desired width of the image.
[DesiredHeight]: Specify the desired height of the image.
[<CURR | WHIT>]: Specify whether the user wants to capture in current theme or wants to capture in white background.
[<PNG>]: Specify the format of the image. The default format is PNG.
[1 | 0]: Specify whether to capture the graph with legends or not.
['TabName']: Specify the tab name on which the graph is supposed to be captured.

Description
This query returns data of the image captured from the graph view of the plugin in the specified format.

Example
:PLUG:JTOL:GRAP? 'MyMeasurement',1000,800,CURR,PNG,1,'Graph'
### :PLUGin:JTOLerance:GRAPh:TLIMits

**Syntax:**
```
PLUGin:JTOLerance:GRAPh:TLIMits 'identifier', <ON|OFF|1|0>
PLUGin:JTOLerance:GRAPh:TLIMits? 'identifier'
```

**Input Parameters:***
- `'identifier'`: Specify the measurement name.
- `<ON|OFF|1|0>`: Enable/disable template limit lines.

**Return Range:** 1|0

**Description:**
This command enables/disables the template limits in the graph for the measurement specified by the measurement name identifier. These limits show the search range amplitude (UI) during characterization defined using the template editor in the M8070A software interface.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

**Example:**
```
:PLUG:JTOL:GRAP:TLIM 'MyMeasurement', ON
```

### :PLUGin:JTOLerance:GRAPh:TPOints

**Syntax:**
```
PLUGin:JTOLerance:GRAPh:TPOints 'identifier', <ON|OFF|1|0>
PLUGin:JTOLerance:GRAPh:TPOints? 'identifier'
```

**Input Parameters:***
- `'identifier'`: Specify the measurement name.
- `<ON|OFF|1|0>`: Enable/disable template test points.

**Return Range:** 1|0

**Description:**
This command enables/disables the template measurement points in the graph for the measurement specified by the measurement name identifier. These test points are defined using the template editor in the M8070A software interface.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

**Example:**
```
:PLUG:JTOL:GRAP:TPO 'MyMeasurement', ON
```
::PLUGin:JTOLerance:GRAPH:CLIMIts[?]
Syntax  ::PLUGin:JTOLerance:GRAPH:CLIMIts 'identifier', <ON|OFF|1|0>
        ::PLUGin:JTOLerance:GRAPH:CLIMIts? 'identifier'
Input   'identifier': Specify the measurement name.
Parameters <ON|OFF|1|0>: Enable/disable compliance limits.
Return Range 1|0
Description This command enables/disables the compliance limits in the graph for the measurement specified by the measurement name identifier. These limits are defined using the template editor in the M8070A software interface.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.
Example  ::PLUGin:JTOLerance:GRAPH:CLIM 'MyMeasurement', ON

::PLUGin:JTOLerance:GRAPH:APOints[?]
Syntax  ::PLUGin:JTOLerance:GRAPH:APOints 'identifier', {ON|OFF|1|0}
        ::PLUGin:JTOLerance:GRAPH:APOints? 'identifier'
Input   'identifier': Specify measurement name.
Parameters  ON | OFF | 1 | 0
Description This command enables or disables the show all measured points option on the graph.
This query returns the present settings.
Example  ::PLUGin:JTOLerance:GRAPH:APOints 'Jitter Tolerance 1', ON

::PLUGin:JTOLerance:FETCh:DATA[:ALL]?
Syntax  ::PLUGin:JTOLerance:FETCh:DATA? 'identifier'
Input   'identifier': Specify the measurement name.
Parameters
Description This command returns the jitter tolerance measurement results addressed by the measurement name identifier.
"M1.DataIn1",100000,0.01,3000000000,0,0,"PASS"

The first item in the comma separated list is the name of the location which is "M1.DataIn1". The next values are Frequency, Amplitude, Number of Bits, Number of Errors, BER and Pass/Fail Results and are repeated in the same order for subsequent measured points.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example :PLUG:JTOL:FETC:DATA? 'MyMeasurement'

:PLUGin:JTOLerance:FETCh:DATA:MAXPass?


Input 'identifier': Specify the measurement name.

Parameters

Description This query returns the maximum passed amplitude information for each measured jitter frequency. However, if a jitter frequency has not achieved a passed result, then the minimum failed amplitude information is being returned for the specific jitter frequency.

The response is build up as a list of location and frequency results enclosed in '(' and ')'. The frequency results are reporting the jitter frequency in Hz and a list of jitter amplitude records enclosed in '(' and ')'.

The jitter amplitude records reports the jitter amplitude in UI, number of compared bits, number of errors, bit error ratio and the PASS/FAIL indication as a comma separated list enclosed by '(' and ')'.

The response format is identical to the SCPI command :PLUGin:JTOLerance:FETCh:DATA:PFINterval with the difference that only one amplitude record is reported per jitter frequency.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example :PLUGin:JTOLerance:FETCh:DATA:MAXPass? 'jitter tolerance 1'

Response (*M2.DataIn*,(100,((100,3290274304,0,0,"PASS")),(500000000,((0.11326875,3554441216,0,0,"PASS")))))

The tutorial script ‘JTOL-parse-results.py’ demonstrates how to parse the response for further processing.
:PLUGin:JTOlerance:FETCh:DATA:PFINterval?

Syntax
:PLUGin:JTOlerance:FETCh:DATA:PFINterval? 'Identifier'

Input
'identifier': Specify the measurement name.

Parameters

Description
This query returns the maximum passed and minimum failed amplitude information for each measured jitter frequency. However, if a jitter frequency has not achieved a passed result, then the minimum failed amplitude information is being returned twice and if there is no failed jitter amplitude, then the maximum passed amplitude is being returned twice.

The response is build up as a list of location and frequency results enclosed in (‘‘ and ’’). The frequency results are reporting the jitter frequency in Hz and a list of jitter amplitude records enclosed in (’ ‘ and ’’).

The jitter amplitude records reports the jitter amplitude in UI, number of compared bits, number of errors, bit error ratio and the PASS/FAIL indication as a comma separated list enclosed by (‘‘ and ’’).

The response format is identical to the SCPI command :PLUGin:JTOlerance:FETCh:DATA:MAXPass with the difference that 2 amplitude records are reported per jitter frequency.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example
:PLUGin:JTOlerance:FETCh:DATA:PFINterval? 'jitter tolerance 1'

Response
(M2.DataIn,(100,((100,3290274304,0,0,"PASS"),(100,3290274304,0,0,"PASS")),(500000000,((0.11326875,3554441216,0,0,"PASS"),(0.151025,39507456,11.278428456643728E-07,"FAIL"))))

The tutorial script 'JTOL-parse-results.py' demonstrates how to parse the response for further processing.

:PLUGin:JTOlerance:BREak

Syntax
:PLUGin:JTOlerance:BREak 'identifier'

Input
'identifier': Specify the measurement name to break/pause.

Parameters

Description
This command breaks/pauses a measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and
Example  The following example breaks a jitter tolerance measurement with the measurement name identifier called 'MyMeasurement':

:\PLUG:JTOL:BRE 'MyMeasurement'

:\PLUGin:JTOLerance:CATalog?

Syntax  :\PLUGin:JTOLerance:CATalog?

Description  This command returns a list of all created jitter tolerance measurement names currently available for measuring.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example  Assume the following is a list of created jitter tolerance measurement names:

:\PLUG:JTOL:NEW 'JTOL_1'
:\PLUG:JTOL:NEW 'JTOL_2'
:\PLUG:JTOL:NEW 'JTOL_3'

The command and returned list would look like the following:

:\PLUG:JTOL:CAT?
"JTOL_1,JTOL_2,JTOL_3"

:\PLUGin:JTOLerance:CONTinue

Syntax  :\PLUGin:JTOLerance:CONTinue 'identifier'

Input  'identifier': Specify the measurement name to continue.

Parameters

Description  This command allows you to continue a jitter tolerance measurement, addressed by the measurement name identifier, that was stopped using the BREak command.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example  The following example continues a halted jitter tolerance measurement addressed with the measurement name identifier called 'MyMeasurement':

:\PLUG:JTOL:CONT 'MyMeasurement'
:PLUGin:JTOLerance:NEW

Syntax  
:PLUGin:JTOLerance:NEW 'identifier'

Input

Parameters

'identifier': Specify the measurement name.

Description

This command creates a new jitter tolerance measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example

The following example creates a jitter tolerance measurement name identifier called 'MyMeasurement':

:PLUG:JTOL:NEW 'MyMeasurement'

NOTE

Creating multiple plugins using this command may slow down the GUI operations which may also result delay in remote programming. To prevent the plugin from opening automatically in the GUI, it is recommended to use "0" as a parameter input in this command.

Example-  :PLUG:JTOL:NEW 'MyMeasurement',0

:PLUGin:JTOLerance:DELeate

Syntax  
:PLUGin:JTOLerance:DELeate 'identifier'

Input

Parameters

'identifier': Specify the measurement name to delete.

Description

This command deletes a previously created jitter tolerance measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example

The following example deletes a jitter tolerance measurement addressed by the measurement name identifier called 'MyMeasurement':

:PLUG:JTOL:DEL 'MyMeasurement'
:PLUGin:JTOLerance:RESet

Syntax: :PLUGin:JTOLerance:RESet 'identifier'

Input Parameters:

'identifier': Specify the measurement name to reset.

Description: This command resets a jitter tolerance measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example: The following example resets a jitter tolerance measurement addressed with the measurement name identifier called 'MyMeasurement':

:PLUG:JTOL:RES 'MyMeasurement'

:PLUGin:JTOLerance:STARt

Syntax: :PLUGin:JTOLerance:STARt 'identifier'

Input Parameters:

'identifier': Specify the measurement name to start.

Description: This command starts a jitter tolerance measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example: The following example starts a jitter tolerance measurement addressed with the measurement name identifier called 'MyMeasurement':

:PLUG:JTOL:STAR 'MyMeasurement'

:PLUGin:JTOLerance:STOP

Syntax: :PLUGin:JTOLerance:STOP 'identifier'

Input Parameters:

'identifier': Specify the measurement name to stop.

Description: This command stops a jitter tolerance measurement addressed by the measurement name identifier.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example
The following example stops a jitter tolerance measurement with the measurement name identifier called 'MyMeasurement':

`:PLUG:JTOL:STOP 'MyMeasurement'`

**:PLUGin:JTOLerance:STEP**

Syntax
`:PLUGin:JTOLerance:STEP 'identifier'`

Input Parameters
`'identifier'`: Specify the measurement name to step.

Description
This command allows you to step through a jitter tolerance measurement, addressed by the measurement name identifier, that was stopped using the BReak command.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example
The following example triggers a single jitter tolerance measurement step of the measurement addressed with the measurement name identifier called 'MyMeasurement':

`:PLUG:JTOL:STEP 'MyMeasurement'`

**:PLUGin:JTOLerance:RUN:HISTory[:STATe][?]**

Syntax
`:PLUGin:JTOLerance:RUN:HISTory[:STATe] 'identifier',<0|1>

`:PLUGin:JTOLerance:RUN:HISTory[:STATe]? 'identifier'`

Input Parameters
`'identifier'`: Specify the measurement name.

Return Range
`0|1`

Description
This command enables/disables the storage of jitter tolerance measurement results addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.
Example

The following example enables storage of jitter tolerance measurement results with the measurement name identifier called 'MyMeasurement':

:PLUG:JTOL:RUN:HIST 'MyMeasurement', 1

:PLUGin:JTOLerance:RUN:HISTory:CLEar

Syntax
:PLUGin:JTOLerance:RUN:HISTory:CLEar 'identifier'

Input
'identifier': Specify the measurement name.

Parameters

Description
This command deletes the stored jitter tolerance measurement history addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example

The following example deletes the stored jitter tolerance measurement history with the measurement name identifier called 'MyMeasurement':

:PLUG:JTOL:RUN:HIST:CLE 'MyMeasurement'

:PLUGin:JTOLerance:RUN:LOG?

Syntax
:PLUGin:JTOLerance:RUN:LOG? 'identifier'

Input
'identifier': Specify the measurement name.

Parameters

Description
This command returns logs for the addressed measurement.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example

The following example returns logs for the measurement name identifier called 'MyMeasurement':

:PLUG:JTOL:RUN:LOG? 'MyMeasurement'

The format of log returned is in the following way:

#XY Message Log
where X denotes the length of digits for Y, Y denotes the number of characters in the log then followed by the log message.
For example:

#10 means, there is 1 digit after 1 and there is 0 character in the log.
#2492 means, there are 3 digits after 2 and there are 492 characters in the log message.

:PLUGin:JTOLerance:RUN:MESSage?

Syntax  :PLUGin:JTOLerance:RUN:MESSage? 'identifier'

Input  'identifier': Specify the measurement name.

Parameters

Description This command returns a string describing the state of an jitter tolerance measurement addressed by the measurement name identifier. Possible states include NotStarted, Running, Suspended, Finished, Error, or Stopped.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example The following example returns the state of a jitter tolerance measurement with the measurement name identifier called 'MyMeasurement':

:PLUG:JTOL:RUN:MESS? 'MyMeasurement'
Running

:PLUGin:JTOLerance:RUN:PROGress?


Input  'identifier': Specify measurement name.

Parameters

Return Range 0.0 to 1.0

Description This command returns a number in the range of 0.0 to 1.0 to indicate the progress of a jitter tolerance measurement addressed by the measurement name identifier. A 0.0 indicates that the measurement has not started and 1.0 indicates the measurement is finished.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.
Example The following example returns the progress of a jitter tolerance measurement with the measurement name identifier called 'MyMeasurement':
  -PLUG.JTOL:RUN:PROG? 'MyMeasurement'
  0.51

:PLUGin:JTOLerance:RUN[:STATus]?

Syntax :PLUGin:JTOLerance:RUN[:STATus]? 'identifier'

Input Parameters
  'identifier': Specify measurement name.

Return Range 0|1

Description This command returns the running status of a jitter tolerance measurement addressed by the measurement name identifier. A 0 indicates the measurement is not running and a 1 indicates the measurement is running.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example The following example returns the running status of a jitter tolerance measurement with the measurement name identifier called 'MyMeasurement':
  -PLUG.JTOL:RUN? 'MyMeasurement'
  1

:PLUGin:JTOLerance:SHOW:COMment[?]

Syntax :PLUGin:JTOLerance:SHOW:COMment 'identifier', <"Comment">
  -PLUGin:JTOLerance:SHOW:COMment? 'identifier'

Input Parameters
  'identifier': Specify measurement name. For example; 'MyMeasurement'
  <"Comment">: Enter the desired comment.

Description This command is used to enter a comment for each measurement. This comment will be displayed on the Measurement History tab.

This query returns the current settings.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example

:PLUG:JTOL:SHOW:COM 'MyMeasurement', "Jitter Tolerance Example"

:PLUG:JTOL:SHOW:COM? 'MyMeasurement'

Jitter Tolerance Example

:PLUGin:JTOLerance:ACQuisition:PROEditable[?]

Syntax

:PLUGin:JTOLerance:ACQuisition:PROEditable 'identifier', <ON|OFF|1|0>

:PLUGin:JTOLerance:ACQuisition:PROEditable? 'identifier'

Input Parameters

'identifier': Specify measurement name. For example; 'MyMeasurement'

ON | OFF | 1 | 0

Description

This command enables/disables the BER, Instruments & Measurement setups parameters of the Jitter Tolerance measurement.

This query returns "0" when the acquisition parameters are in disabled state or the measurement is in running state. This query returns "1" when the acquisition parameters are in enabled state.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A and M8046A.

Example

:PLUG:JTOL:ACQ:PROE 'MyMeasurement',1

:PLUG:JTOL:ACQ:PROE? 'MyMeasurement'

1
The M8070A system software provides quick design analysis with the eye diagram capability. The measurement allows a quick check for the DUT’s signal output and determines the signal quality. The eye contour lines display the measured eye at a deeper BER level, for accurate results. The eye diagram generates a three dimensional graph of the bit error rate (BER). This measurement helps in determining and analyzing the quality of the DUT’s signal output. The eye diagram results comprise of voltage(y), time(x) and BER(z). The measurement comes with the set of Acquisition and Evaluation Parameters which are used to configure the measurement run. The Acquisition parameters can only be configured before the measurement starts (i.e. during the measurement run you cannot change the value of Acquisition parameter); however the Evaluation parameters can be configured any time even after the measurement has been run.

This subnode has the following SCPI structure:

`:EDiagram
  - `BREak
  - `CATalog?
  - `CONTinue
  - `DELete
  - `NEW
  - `RESet
  - `STARt
  - `STEP
  - `STOP
  - `ACQuisition
    - ...
  - `EVALuation
    - ...
  - `FETCH[:RESult]
    - ...
  - `RUN
    - ...
  - `SHOW
    - ...

`:PLUGin:EDiagram Subnode
This subnode has the following commands and subnodes:

### Table 82

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:BREak</td>
<td>:PLUGin:EDiagram:BREak on page 425</td>
</tr>
<tr>
<td>:CONTinue</td>
<td>:PLUGin:EDiagram:CONTinue on page 425</td>
</tr>
<tr>
<td>:DELete</td>
<td>:PLUGin:EDiagram:DELete on page 426</td>
</tr>
<tr>
<td>:NEW</td>
<td>:PLUGin:EDiagram:NEW on page 426</td>
</tr>
<tr>
<td>:RESet</td>
<td>:PLUGin:EDiagram:RESet on page 427</td>
</tr>
<tr>
<td>:START</td>
<td>:PLUGin:EDiagram:START on page 427</td>
</tr>
<tr>
<td>:STEP</td>
<td>:PLUGin:EDiagram:STEP on page 427</td>
</tr>
<tr>
<td>:STOP[</td>
<td>:PLUGin:EDiagram:STOP on page 428</td>
</tr>
</tbody>
</table>

**Subnodes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:EVALuation</td>
<td>:PLUGin:EDiagram:EVALuation Subnode on page 433</td>
</tr>
<tr>
<td>:RUN</td>
<td>:PLUGin:EDiagram:RUN Subnode on page 441</td>
</tr>
<tr>
<td>:SHOW</td>
<td>:PLUGin:EDiagram:SHOW Subnode on page 444</td>
</tr>
</tbody>
</table>
:PLUGin:EDiagram:BREak
Syntax :PLUGin:EDiagram:BREak 'identifier'
Input 'identifier': Specify the measurement name to break/pause.
Parameters
Description This command breaks/pauses a measurement addressed by the measurement name identifier.
This SCPI is applicable for M8041A, M8051A and M8062A.
Example The following example breaks an eye diagram measurement with the measurement name identifier called 'MyMeasurement':
:PLUG:EDI:BRE 'MyMeasurement'

:PLUGin:EDiagram:CATalog?
Syntax :PLUGin:EDiagram:CATalog?
Description This command returns a list of all created eye diagram measurement names currently available for measuring.
This SCPI is applicable for M8041A, M8051A and M8062A.
Example Assume three eye diagram measurements named as EDiagram_1, EDiagram_2 and EDiagram_3 are created using NEW command:
:PLUG:EDI:NEW 'EDIagram_1'
:PLUG:EDI:NEW 'EDIagram_2'
:PLUG:EDI:NEW 'EDIagram_3'
The command and returned list would look like the following:
:PLUG:EDI:CAT?
"EDIagram_1", "EDIagram_2", "EDIagram_3"

:PLUGin:EDiagram:CONTinue
Syntax :PLUGin:EDiagram:CONTinue 'identifier'
Input 'identifier': Specify the measurement name to continue.
Parameters
Description This command allows you to continue an eye diagram measurement, addressed by the measurement name identifier, which was stopped using the BREak command.
This SCPI is applicable for M8041A, M8051A and M8062A.
Example The following example continues a halted eye diagram measurement addressed with the measurement name identifier called 'MyMeasurement':
:PLUG:EDI:CONT 'MyMeasurement'
:PLUGin:EDIagram:DELete

Syntax: :PLUGin:EDIagram:DELete 'identifier'

Input:
- 'identifier': Specify the measurement name to delete.

Parameters:

Description: This command deletes a previously created eye diagram measurement addressed by the measurement name identifier. This SCPI is applicable for M8041A, M8051A and M8062A.

Example: The following example deletes an eye diagram measurement addressed by the measurement name identifier called 'MyMeasurement':

:PLUG:EDI:DEL 'MyMeasurement'

:PLUGin:EDIagram:NEW

Syntax: :PLUGin:EDIagram:NEW 'identifier'

Input:
- 'identifier': Specify the measurement name.

Parameters:

Description: This command creates a new eye diagram measurement name identifier. This SCPI is applicable for M8041A, M8051A and M8062A.

Example: The following example creates an eye diagram measurement name identifier called 'MyMeasurement':

:PLUG:EDI:NEW 'MyMeasurement'

NOTE

Creating multiple plugins using this command may slow down the GUI operations which may also result delay in remote programming. To prevent the plugin from opening automatically in the GUI, it is recommended to use "0" as a parameter input in this command.

Example: :PLUG:EDI:NEW 'MyMeasurement',0
:PLUGin:EDIagram:RESet

Syntax  
:PLUGin:EDIagram:RESet 'identifier'

Input
'identifier': Specify the measurement name to reset.

Parameters

Description  
This command resets an eye diagram measurement addressed by the measurement name identifier. This SCPI is applicable for M8041A, M8051A and M8062A.

Example  
The following example resets an eye diagram measurement addressed with the measurement name identifier called 'MyMeasurement':
:PLUG:EDI:RES 'MyMeasurement'

:PLUGin:EDIagram:STARt

Syntax  
:PLUGin:EDIagram:STARt 'identifier'

Input
'identifier': Specify the measurement name to start.

Parameters

Description  
This command starts an eye diagram measurement addressed by the measurement name identifier. This SCPI is applicable for M8041A, M8051A and M8062A.

Example  
The following example starts an eye diagram measurement addressed with the measurement name identifier called 'MyMeasurement':
:PLUG:EDI:STAR 'MyMeasurement'

:PLUGin:EDIagram:STEP

Syntax  
:PLUGin:EDIagram:STEP 'identifier'

Input
'identifier': Specify the measurement name to step.

Parameters

Description  
This command allows you to step through an eye diagram measurement, addressed by the measurement name identifier, that was stopped using the BREak command. This SCPI is applicable for M8041A, M8051A and M8062A.
Example The following example triggers a single eye diagram measurement step of the measurement addressed with the measurement name identifier called 'MyMeasurement':
:PLUG:EDI:STEP 'MyMeasurement'

:PLUGin:EDIagram:STOP

Syntax :PLUGin:EDIagram:STOP 'identifier'

Input 'identifier': Specify the measurement name to stop.

Parameters

Description This command stops an eye diagram measurement addressed by the measurement name identifier. This SCPI is applicable for M8041A, M8051A and M8062A.

Example The following example stops an eye diagram measurement with the measurement name identifier called 'MyMeasurement':
:PLUG:EDI:STOP 'MyMeasurement'
This subnode has the following SCPI structure:

```
:ACQuisition
  - :ALOCation[?]
  - :NEYEs[?]
  - :PERSistence[?]
  - :TIME[?]
  - :PROEditable[?]
```

This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:ALOCation[?]</td>
<td>:PLUGin:EDIagram:ACQuisition:ALOCation[?] on page 430</td>
</tr>
<tr>
<td>:NEYEs[?]</td>
<td>:PLUGin:EDIagram:ACQuisition:NEYEs[?] on page 430</td>
</tr>
<tr>
<td>:PERSistence[?]</td>
<td>:PLUGin:EDIagram:ACQuisition:PERSistence[?] on page 431</td>
</tr>
<tr>
<td>:PROEditable[?]</td>
<td>:PLUGin:EDIagram:ACQuisition:PROEditable[?] on page 432</td>
</tr>
</tbody>
</table>
:PLUGin:EDlagram:ACQuisition:ALOCation[?]

Syntax: :PLUGin:EDlagram:ACQuisition:ALOCation 'Identifier', <location-string>
       :PLUGin:EDlagram:ACQuisition:ALOCation? 'Identifier'

Input Parameters
Identifier: 'ParameterSweep1'
<location-string>: 'M*.DataIn*'

Description
This command sets the location/location group string against which the data acquisition is performed.
This query returns the current location/location group string against which measurement is configured for the data acquisition.
This SCPI is applicable for M8041A, M8051A and M8062A.

Example: :PLUG:EDI:ACQ:ALOC 'MyMeasurement','M1.DataIn1'

:PLUGin:EDlagram:ACQuisition:NEYEs[?]

Syntax: :PLUGin:EDlagram:ACQuisition:NEYEs 'Identifier', TDZ | ODF
       :PLUGin:EDlagram:ACQuisition:NEYEs? 'Identifier',

Input Parameters
Identifier: 'MyMeasurement'

Range
TDZ | ODF

Description
This command sets the number eyes to be displayed in UI. It can be either 1.5 or 2.0.
The query returns current value.
This SCPI is applicable for M8041A, M8051A and M8062A.

Example: :PLUG:EDI:ACQ:NEYEs 'MyMeasurement', TDZ
**:PLUGin:EDiagram:ACQuisition:PERSistence[*]?

**Syntax**
**:PLUGin:EDiagram:ACQuisition:PERSistence 'Identifier', <INF | FTIM>
**:PLUGin:EDiagram:ACQuisition:PERSistence? 'Identifier',

**Input Parameters**
Identifier: 'MyMeasurement'

**Return Range**
INF | FTIM

**Description**
This command sets the criteria for persistence. Persistence could be infinite or fixed time.
The query returns the current setting.
This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**
**:PLUG:EDI:ACQ:PERS 'MyMeasurement',INF

**:PLUGin:EDiagram:ACQuisition:TIME[*]?

**Syntax**
**:PLUGin:EDiagram:ACQuisition:TIME 'Identifier', <NRf>
**:PLUGin:EDiagram:ACQuisition:TIME? 'Identifier'

**Input Parameters**
Identifier: 'MyMeasurement'

**Return Range**
50s to 216ks

**Description**
This command allows the user to specify the persistence time.
The query returns current setting.
This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**
**:PLUG:EDI:ACQ:TIME 'MyMeasurement', 80
### :PLUGin:EDiagram:ACquisition:PROEditable[?]

**Syntax**

`:PLUGin:EDiagram:ACquisition:PROEditable 'identifier', <ON|OFF|1|0>`

`:PLUGin:EDiagram:ACquisition:PROEditable? 'identifier'`

**Input Parameters**

- `'identifier'`: Specify the measurement name.
- `ON | OFF | 1 | 0`

**Return Range**

0 or 1

**Description**

This command enables/disables the acquisition parameters of the Eye Diagram measurement.

This query returns “0” when the acquisition parameters are in disabled state or the measurement is in running state. This query returns “1” when the acquisition parameters are in enabled state.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example**

`:PLUG:EDI:ACQ:PROE 'MyMeasurement', 1`

`:PLUG:EDI:ACQ:PROE? 'MyMeasurement', 1`
:PLUGin:EDiagram:EVALuation Subnode

This subnode has the following SCPI structure:

```
:EVALuation
   - :BERThresh[?]
   - :BERThresh:THReshold[?]
   - :TTIME[?]
```

This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
</table>

:PLUGin:EDiagram:EVALuation:BERThresh[?]

**Syntax**

```
:PLUGin:EDiagram:EVALuation:BERThresh 'Identifier', <ZERR | BTHR>
```

**Input Parameters**

- Identifier: 'MyMeasurement'

**Return Range**

ZERR | BTHR

**Description**

This command tells how to calculate measurement results, such as, eye height, eye width and JPPeak either using '0 errors' (ZERO) or a specific 'BER Threshold' (THReshold). In the latter case, numeric value of the
The threshold is specified with the `:PLUG:EDIagram:EVALuation:BERThresh:THReshold[?]` command on page 434.

The query returns either 0 errors (ZERR) or BER threshold (BTHR).

This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**: `:PLUG:EDI:EVAL:BERT 'MyMeasurement', BTHR`

### :PLUGin:EDIagram:EVALuation:BERThresh:THReshold[?]

**Syntax**: `:PLUGin:EDIagram:EVALuation:BERThreshold 'Identifier', <NRf>`

`:PLUGin:EDIagram:EVALuation:BERThreshold? 'Identifier'`

**Input Parameters**

- **Identifier**: 'MyMeasurement'

- **Range**: 1E-12 to 1E-1

- **Description**: This command allows the user to specify BER Threshold value.

- **Example**: `:PLUG:EDI:EVAL:BERT 'MyMeasurement', 1E-10`

### :PLUGin:EDIagram:EVALuation:TTIMe[?]

**Syntax**: `:PLUGin:EDIagram:EVALuation:TTIMe 'Identifier', <TEIG | TNIN>`

`:PLUGin:EDIagram:EVALuation:TTIMe? 'Identifier'`

**Input Parameters**

- **Identifier**: 'MyMeasurement'

- **Range**: TEIG | TNIN

- **Description**: This command allows the user to specify the transition time. It can be either 10/90 or 20/80.

- **Example**: `:PLUG:EDI:EVAL:TTIM 'MyMeasurement', TNIN`
This subnode has the following SCPI structure:

```plaintext
:FETCh
 [:RESult]
  :AMPLitude?
  :CVOLtage?
  :DCDistortion?
  :FALLtime?
  :HEIGHt?
  :HILevel?
  :JPPeak?
  :JRMSquare?
  :LOLevel?
  :RISetime?
  :SCOunt?
  :SNRatio?
  :WIDTh?
```

This subnode has the following commands and subnodes:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:CVOLtage?</td>
<td>:PLUGin:EDiagram:FETCh[:RESult]:CVOLtage? on page 437</td>
</tr>
<tr>
<td>:FALLtime?</td>
<td>:PLUGin:EDiagram:FETCh[:RESult]:FALLtime? on page 437</td>
</tr>
</tbody>
</table>
SCPI Command Reference

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:HEIGHT?</td>
<td>:PLUGin:EDIagram:FETCh[:RESULT]:HEIGHT? on page 438</td>
</tr>
<tr>
<td>:HILevel?</td>
<td>:PLUGin:EDIagram:FETCh[:RESULT]:HILevel? on page 438</td>
</tr>
<tr>
<td>:JPPeak?</td>
<td>:PLUGin:EDIagram:FETCh[:RESULT]:JPPeak? on page 438</td>
</tr>
<tr>
<td>:JRMSSquare?</td>
<td>:PLUGin:EDIagram:FETCh[:RESULT]:JRMSSquare? on page 439</td>
</tr>
<tr>
<td>:LOLevel?</td>
<td>:PLUGin:EDIagram:FETCh[:RESULT]:LOLevel? on page 439</td>
</tr>
</tbody>
</table>

:PLUGin:EDIagram:FETCh[:RESULT]:AMPLitude?

**Syntax:**
:PLUGin:EDIagram:FETCh[:RESULT]:AMPLitude? 'identifier'

**Input Parameters:**
'identifier': Specify the measurement name.

**Description:**
This command returns the eye amplitude which is the difference between the mean logic 1 level values and the mean logic 0 level values in a histogram of an eye diagram.

This SCPI is applicable for M8041A, M8051A and M8062A.

**Example:**
:PLUG:EDI:FETC:AMPL? 'MyMeasurement'
**:PLUGin:EDIagram:FETCh[:RESult]:CVOLtage?**

**Syntax**

`:PLUGin:EDIagram:FETCh[:RESult]:CVOLtage? 'identifier'`

**Input Parameters**

- `'identifier'`: Specify the measurement name.

**Description**

This command returns the cross voltage which is the measurement of the crossing point level in relation to the logic 1 level and logic 0 level. This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**

`:PLUG:EDI:FETC:CVOL? 'MyMeasurement'`

---

**:PLUGin:EDIagram:FETCh[:RESult]:DCDistortion?**

**Syntax**

`:PLUGin:EDIagram:FETCh[:RESult]:DCDistortion? 'identifier'`

**Input Parameters**

- `'identifier'`: Specify the measurement name.

**Description**

This command returns the value of DC Distortion calculated after a run. This value is the difference between the period of a 1 bit and a 0 bit. This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**

`:PLUG:EDI:FETC:DCD? 'MyMeasurement'`

---

**:PLUGin:EDIagram:FETCh[:RESult]:FALLtime?**

**Syntax**

`:PLUGin:EDIagram:FETCh[:RESult]:FALLtime? 'identifier'`

**Input Parameters**

- `'identifier'`: Specify the measurement name.

**Description**

This command returns the value of the fall time. Fall time is a measurement of the mean transition time of the data on the downward slope of an eye diagram. This transition time is either 2080 or 1090. See :PLUGin:EDIagram:EVALuation:TTIMe[?] on page 434. This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**

`:PLUG:EDI:FETC:FALL? 'MyMeasurement'`
Syntax: :PLUGin:EDiagram:FETCH[:RESULT]:HEIGHT?

Parameters:
- 'identifier': Specify the measurement name.

Description: This command returns the value of the eye height which is a measurement of the vertical opening of an eye diagram. This opening is affected by the BER threshold that is set by the BER threshold command. See :PLUGin:EDiagram:EVALuation:BERThresh[?] on page 433.

This SCPI is applicable for M8041A, M8051A and M8062A.

Example: :PLUG:EDI:FETC:HEIG? 'MyMeasurement'

Syntax: :PLUGin:EDiagram:FETCH[:RESULT]:HILEvel?

Parameters:
- 'identifier': Specify the measurement name.

Description: This command returns the measurement of the mean value of the logical 1 in the eye diagram. This is directly affected by the values set for Eye Boundaries. One Level is a measure of the mean value of the logical 1 of an eye diagram.

This SCPI is applicable for M8041A, M8051A and M8062A.

Example: :PLUG:EDI:FETC:HIL? 'MyMeasurement'

Syntax: :PLUGin:EDiagram:FETCH[:RESULT]:JPPeak?

Parameters:
- 'identifier': Specify the measurement name.

Description: This command returns Jitter P-P value. It is the full width of the eye diagram at the eye crossing point. This is affected by the BER threshold that is set by the BER threshold command. See :PLUGin:EDiagram:EVALuation:BERThresh:THReshold[?] on page 434.

This SCPI is applicable for M8041A, M8051A and M8062A.

Example: :PLUG:EDI:FETC:JPP? 'MyMeasurement'
```plaintext
:PLUGin:EDiagram:FETCh[:RESult]:JRMSquare?

Syntax    :PLUGin:EDiagram:FETCh[:RESult]:JRMSquare? 'identifier'
Input     'identifier': Specify the measurement name.
Parameters
Description This command returns the Jitter Root-Mean Square value. It is the standard deviation of the normal distribution of random jitter. It is dependent on the BER Threshold.

This SCPI is applicable for M8041A, M8051A and M8062A.

Example   :PLUG:EDI:FETC:JRMS? 'MyMeasurement'

:PLUGin:EDiagram:FETCh[:RESult]:LOLevel?

Syntax    :PLUGin:EDiagram:FETCh[:RESult]:LOLevel? 'identifier'
Input     'identifier': Specify the measurement name.
Parameters
Description This command returns the measurement of the mean value of the logical 1 in the eye diagram. This is directly affected by the values set for eye boundaries.

This SCPI is applicable for M8041A, M8051A and M8062A.

Example   :PLUG:EDI:FETC:LOL? 'MyMeasurement'

:PLUGin:EDiagram:FETCh[:RESult]:RISetime?

Syntax    :PLUGin:EDiagram:FETCh[:RESult]:RISetime? 'identifier'
Input     'identifier': Specify the measurement name.
Parameters
Description This command returns rise time which is a measurement of the mean transition time of the data on the upward slope of an eye diagram. This transition time is either 20/80 or 10/90. See :PLUGin:EDiagram:EVALuation:TTIme[?] on page 434

This SCPI is applicable for M8041A, M8051A and M8062A.

Example   :PLUG:EDI:FETC:RIS? 'MyMeasurement'
```
:PLUGin:EDiagram:FETCh[:RESult]:SCOunt?
Syntax :PLUGin:EDiagram:FETCh[:RESult]:SCOunt? 'identifier'
Input 'identifier': Specify the measurement name.
Parameters
Description This command returns the sample count.
This SCPI is applicable for M8041A, M8051A and M8062A.
Example :PLUG:EDI:FETC:SCO? 'MyMeasurement'

:PLUGin:EDiagram:FETCh[:RESult]:SNRatio?
Syntax :PLUGin:EDiagram:FETCh[:RESult]:SNRatio? 'identifier'
Input 'identifier': Specify the measurement name.
Parameters
Description This command returns signal-to-noise ratio which is a measurement of the signal difference between 1 level and 0 level in relation to the rms value of 1-level noise + rms value of 0-level noise.
This SCPI is applicable for M8041A, M8051A and M8062A.
Example :PLUG:EDI:FETC:SNR? 'MyMeasurement'

:PLUGin:EDiagram:FETCh[:RESult]:WIDTh?
Syntax :PLUGin:EDiagram:FETCh[:RESult]:WIDTh? 'identifier'
Input 'identifier': Specify the measurement name.
Parameters
Description This command returns the calculated eye width which is horizontal measurement of the eye opening at a specified BER Threshold. It can be either at the eye crossing point (CROSSing) or custom defined (CUSTom).
This SCPI is applicable for M8041A, M8051A and M8062A.
Example :PLUG:EDI:FETC:WIDT? 'MyMeasurement'
This subnode has the following SCPI structure:

```
:RUN
  :MESSage?
  :PROGress?
  :STATus?
  :HISTORY
     [:STATE][?]
     :CLEar
     :LOG?
```

This subnode has the following commands:

### Table 86

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
</table>
:PLUGin:EDIagram:RUN:MESSage?
Syntax :PLUGin:EDIagram:RUN:MESSage? 'Identifier'
Input Identifier: 'MyMeasurement'
Parameters
Description This command returns a string describing the state of an eye diagram measurement addressed by the measurement name identifier. Possible states include Not Started, Running, Suspended, Finished, Error, or Stopped.
This SCPI is applicable for M8041A, M8051A and M8062A.
Example :PLUG:EDI:RUN:MESS? 'MyMeasurement'

:PLUGin:EDIagram:RUN:PROGress?
Syntax :PLUGin:EDIagram:RUN:PROGress? 'Identifier'
Input Identifier: 'MyMeasurement'
Parameters
Return Range 0 to 1
Description This command returns a number in the range of 0.0 to 1.0 to indicate the progress of an eye diagram measurement addressed by the measurement name identifier. 0.0 indicates that the measurement has not started and 1.0 indicates the measurement is finished.
This SCPI is applicable for M8041A, M8051A and M8062A.
Example :PLUG:EDI:RUN:PROG? 'MyMeasurement'

:PLUGin:EDIagram:RUN:STATus?
Syntax :PLUGin:EDIagram:RUN:STATus? 'Identifier'
Input Identifier: 'MyMeasurement'
Parameters
Return Range 0|1
Description This command returns the running status of an eye diagram measurement addressed by the measurement name identifier. 0 indicates the measurement is not running and a 1 indicates the measurement is running.
This SCPI is applicable for M8041A, M8051A and M8062A.
Example :PLUG:EDI:RUN:STAT? 'MyMeasurement'
:**PLUGin:EDIagram:RUN:HISTory[:STATE][?]

Syntax: :PLUGin:EDIagram:RUN:HISTory[:STATE] 'Identifier', <0|1|ON|OFF>

:PLUGin:EDIagram:RUN:HISTory[:STATE]? 'Identifier'

Input Parameters:
- Identifier: 'MyMeasurement'

Return Range: 0|1

Description:
This command enables/disables the storage of eye diagram measurement results addressed by the measurement name identifier.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A and M8062A.

Example:
:PLUG:EDI:RUN:HIST 'MyMeasurement', 1

:**PLUGin:EDIagram:RUN:HISTory:CLEar

Syntax: :PLUGin:EDIagram:RUN:HISTory:CLEar 'Identifier'

Input Parameters:
- Identifier: 'MyMeasurement'

Description:
This command deletes the eye diagram measurement history addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A and M8062A.

Example:
:PLUG:EDI:RUN:HIST:CLE 'MyMeasurement'

:**PLUGin:EDIagram:RUN:LOG?


Input Parameters:
- Identifier: 'MyMeasurement'

Description:
This command returns logs for the addressed measurement.

This SCPI is applicable for M8041A, M8051A and M8062A.

Example:
:PLUG:EDI:RUN:LOG? 'MyMeasurement'

The format of log returned is in the following way:

#XY Message Log

where X denotes the length of digits for Y, Y denotes the number of
characters in the log then followed by the log message.
For example:
#10 means, there is 1 digit after 1 and there is 0 character in the log.
#2492 means, there are 3 digits after 2 and there are 492 characters in the log message.

:PLUGin:EDIagram:SHOW Subnode

This subnode has the following SCPI structure:

```
:SHOW
  :CONTour[?]
  :LEGend[?]
  :UNIT[?]
  :WAVEform[?]
  :SGRaph[?]
  :THReshold[?]
  :COMment[?]
```

This subnode has the following commands:

Table 87

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:CONTour</td>
<td>:PLUGin:EDIagram:SHOW:CONTour(?) on page 445</td>
</tr>
<tr>
<td>:LEGend</td>
<td>:PLUGin:EDIagram:SHOW:CONTour:LEGend(?) on page 445</td>
</tr>
<tr>
<td>:UNIT</td>
<td>:PLUGin:EDIagram:SHOW:UNIT(?) on page 446</td>
</tr>
<tr>
<td>:WAVEform</td>
<td>:PLUGin:EDIagram:SHOW:WAVEform(?) on page 446</td>
</tr>
</tbody>
</table>
:PLUGin:EDIagram:SHOW:CONTour

Syntax :PLUGin:EDIagram:SHOW:CONTour 'Identifier', <0|1|ON|OFF>
        :PLUGin:EDIagram:SHOW:CONTour? 'Identifier'

Input Parameters
Identifier: 'MyMeasurement'

Return Range 0|1

Description
This command shows/hides the contour lines on the graph.
This query returns the current setting.
This SCPI is applicable for M8041A, M8051A and M8062A.

Example :PLUG:EDI:SHOW:CONT 'MyMeasurement', 1

:PLUGin:EDIagram:SHOW:CONTour:LEGend

Syntax :PLUGin:EDIagram:SHOW:CONTour:LEGend 'Identifier', <0|1|ON|OFF>
        :PLUGin:EDIagram:SHOW:CONTour:LEGend? 'Identifier'

Input Parameters
Identifier: 'MyMeasurement'

Return Range 0|1

Description
The command shows/hides contour legends.
This query returns the current setting.
This SCPI is applicable for M8041A, M8051A and M8062A.

Example :PLUG:EDI:SHOW:CONT:LEG 'MyMeasurement', 1
**:PLUGin:EDiagram:SHOW:UNIT[?]**

**Syntax**

**:PLUGin:EDiagram:SHOW:UNIT 'Identifier', <UNIT | TIME>**

**:PLUGin:EDiagram:SHOW:UNIT? 'Identifier',**

**Input Parameters**

Identifier: 'MyMeasurement'

**Return Range**

UNIT | TIME

**Description**

The command sets the display unit. It can be either in terms of seconds or UNIT.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**

**:PLUG:EDI:SHOW:UNIT 'MyMeasurement', UNIT**

**:PLUGin:EDiagram:SHOW:WAVeform[?]**

**Syntax**

**:PLUGin:EDiagram:SHOW:WAVeform 'Identifier', <0|1|OFF|ON>**

**:PLUGin:EDiagram:SHOW:WAVeform? 'Identifier',**

**Input Parameters**

Identifier: 'MyMeasurement'

<0|1|OFF|ON>: Show/hide waveform

**Return Range**

0|1

**Description**

This command shows/hides the waveform. When this feature is enabled, the waveform is displayed together with the contour lines. Disabling this feature will only display contour lines, if available.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A and M8062A.

**Example**

**:PLUG:EDI:SHOW:WAV 'MyMeasurement', 1**
:PLUG:EDiagram:SHOW:WAVEform:SGRaph[?]

Syntax  
:PLUG:EDiagram:SHOW:WAVEform:SGRaph 'Identifier', <0|1|OFF|ON>
:PLUG:EDiagram:SHOW:WAVEform:SGRaph? 'Identifier'

Input Parameters
Identifier: 'MyMeasurement'
<0|1|OFF|ON>: Smooth waveform graphics.

Return Range 0|1

Description This command smooths out the waveform graphics. When this feature is enabled, the waveform is displayed together with the contour lines. Disabling this feature will only display contour lines, if available.

This query returns the current setting.
This SCPI is applicable for M8041A, M8051A and M8062A.

Example :PLUG:EDI:SHOW:WAV:SGR 'MyMeasurement', 1

:PLUG:EDiagram:SHOW:WAVEform:THReshold[?]

Syntax  
:PLUG:EDiagram:SHOW:WAVEform:THReshold 'Identifier', <NR3>
:PLUG:EDiagram:SHOW:WAVEform:THReshold? 'Identifier'

Input Parameters
Identifier: 'MyMeasurement'

Return Range 1E-6 to 1E-1

Description This command sets the waveform BER threshold value.
This query returns the current setting.
This SCPI is applicable for M8041A, M8051A and M8062A

Example :PLUG:EDI:SHOW:WAV:THR 'MyMeasurement', 1E-3
Syntax:

`:PLUGin:EDiagram:SHOW:COMment[?]`

Input Parameters:

- `Identifier`: Specify the plugin name. For e.g. 'MyMeasurement'
- `<"Commnet">`: Enter the desired comment.

Description:

This command is used to enter a comment for each measurement. This comment will be displayed on the Measurement History tab. This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example:

`:PLUG:EDI:SHOW:COM 'MyMeasurement', "Eye Measurement Example"

`:PLUG:EDI:SHOW:COM? 'MyMeasurement'

Eye Measurement Example
The Parameter Sweep plugin allows you to sweep the parameters of the selected location/location group to the desired analyzer in order to calculate BER. This is helpful in scenarios when a user wants to pick a property to sweep within a specified range with specified steps and measure BER with respect to that sweep. Using this plugin, a user can sweep the parameters using the acquisition parameters provided in the Parameter window and then run the measurement to calculate BER.

This subnode has the following SCPI structure:

```
:PSWEep
  :CATalog?
  :DELete
  :FETCh:DATA?
  :GRAPh?
  :NEW
  :RESet
  :STARt
  :BREak
  :STOP
  :CONTinue
  :STEP
  :ACQuisition
  ...
  :RUN
  ...
  :SHOW
  ...
```
This subnode has the following commands and subnodes:

Table 88

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRAPh?</td>
<td>:PLUGin:PSWEep:GRAPh? on page 452</td>
</tr>
<tr>
<td>:NEW</td>
<td>:PLUGin:PSWEep:NEW on page 453</td>
</tr>
<tr>
<td>:RESet</td>
<td>:PLUGin:PSWEep:RESet on page 454</td>
</tr>
<tr>
<td>:START</td>
<td>:PLUGin:PSWEep:START on page 454</td>
</tr>
<tr>
<td>:BREak</td>
<td>:PLUGin:PSWEep:BREak on page 454</td>
</tr>
<tr>
<td>:STOP</td>
<td>:PLUGin:PSWEep:STOP on page 455</td>
</tr>
<tr>
<td>:CONTinue</td>
<td>:PLUGin:PSWEep:CONTinue on page 455</td>
</tr>
<tr>
<td>:STEP</td>
<td>:PLUGin:PSWEep:STEP on page 456</td>
</tr>
<tr>
<td>Subnodes</td>
<td></td>
</tr>
<tr>
<td>:RUN</td>
<td>:PLUGin:PSWEep:RUN Subnode on page 466</td>
</tr>
<tr>
<td>:SHOW</td>
<td>:PLUGin:PSWEep:SHOW Subnode on page 470</td>
</tr>
</tbody>
</table>

:PLUGin:PSWEep:CATalog?

Syntax :PLUGin:PSWEep:CATalog?

Description This command returns a list of all created parameter sweep names currently available for measuring.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

Example Assume three parameter sweep measurements named as: 'ParameterSweep1', 'ParameterSweep2' and 'ParameterSweep3' are
created using NEW command:
:PLUG:PSWE:NEW 'ParameterSweep1'
:PLUG:PSWE:NEW 'ParameterSweep2'
:PLUG:PSWE:NEW 'ParameterSweep3'
The command and returned list would look like the following:
:PLUG:PSWE:CAT?
"ParameterSweep1","ParameterSweep2","ParameterSweep3"

:PLUGin:PSWEep:DELeTe
Syntax  :PLUGin:PSWEep:DELeTe 'identifier'
Input Parameters
'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'
Description This command deletes a previously created parameter sweep measurement addressed by the measurement name identifier.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.
Example The following example deletes a parameter sweep measurement addressed by the measurement name identifier called 'Parameter Sweep 1':
:PLUG:PSWE:DELe 'Parameter Sweep 1'

:PLUGin:PSWEep:FETCh:DATA?
Syntax  :PLUGin:PSWEep:FETCh:DATA? 'identifier', ['Location']
Input Parameters
'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'
'Location': Specify location (optional) in case of location group
Description This command returns the raw data of the parameter sweep measurement associated with a specific measurement name identifier.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.
Example :PLUG:PSWE:FETC:DATA? 'ParameterSweep1', 'M1.DataIn1'
A measurement run on a location identifier (for example, 'M1.DataIn1') will return following result:
The first item in the comma separated list is the name of the location for which the data is being queried. Rest of the values in the comma separated list for each measurement point are Sweep Value, Error Ratio, Compared Bits, Errored Bits, Errored Zero Ratio, Compared Zeros, Errored Zeros, Error One Ratio, Compared Ones and Errored Ones are repeated in the same order for subsequent measured points.

If the location is omitted from the above example e.g. if the query is :PLUG:PSWE:FETC:DATA? 'ParameterSweep1' then the query returns the comma separated list of all the analyzer location group participating in the measurement.

**:PLUGin:PSWEep:GRAPh?**

**Syntax** :PLUGin:PSWEep:GRAPh? 'identifier',[DesiredWidth],[DesiredHeight],[<CURR | WHIT>],[<PNG>],[1 | 0],[TabName]

**Input Parameters**
- 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'
- Other optional parameters are:
  - [DesiredWidth]: Specify the desired width of the image.
  - [DesiredHeight]: Specify the desired height of the image.
  - [<CURR | WHIT>]: Specify whether the user wants to capture in current theme or wants to capture in white background.
  - [<PNG>]: Specify the format of the image. The default format is PNG.
  - [1 | 0]: Specify whether to capture the graph with legends or not.
  - ['TabName']: Specify the tab name on which the graph is supposed to be captured.

**Description** This query returns data of the image captured from the graph view of the plugin in the specified format.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

**Example** :PLUG:PSWE:GRAP? 'Parameter Sweep 1',1000,800,CURR,PNG,1,'Graph'
**:PLUGin:PSWEep:NEW**

**Syntax**  
`:PLUGin:PSWEep:NEW 'identifier'`

**Input Parameters**

- **'Identifier':** Specify the plugin name. For e.g. 'ParameterSweep1'

**Description**

This command creates a new parameter sweep measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

**Example**

The following example creates a parameter sweep measurement name identifier called 'ParameterSweep1':

`:PLUG:PSWE:NEW 'ParameterSweep1'`

**NOTE**

Creating multiple plugins using this command may slow down the GUI operations which may also result delay in remote programming. To prevent the plugin from opening automatically in the GUI, it is recommended to use "0" as a parameter input in this command.

**Example** - :PLUG:PSWE:NEW 'ParameterSweep1',0
**:PLUGin:PSWEep:RESet**

**Syntax:** :PLUGin:PSWEep:RESet 'identifier'

**Input Parameters:**
- **Identifier**: Specify the plugin name. For e.g. 'ParameterSweep1'

**Description:** This command resets a parameter sweep measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

**Example:** The following example resets a parameter sweep measurement addressed with the measurement name identifier called 'ParameterSweep1':

:PLUG:PSWE:RES 'ParameterSweep1'

**:PLUGin:PSWEep:STARt**

**Syntax:** :PLUGin:PSWEep:STARt 'identifier'

**Input Parameters:**
- **Identifier**: Specify the plugin name. For e.g. 'ParameterSweep1'

**Description:** This command starts a parameter sweep measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

**Example:** The following example starts a parameter sweep measurement addressed with the measurement name identifier called 'ParameterSweep1':

:PLUG:PSWE:STAR 'ParameterSweep1'

**:PLUGin:PSWEep:BREak**

**Syntax:** :PLUGin:PSWEep:BREak 'identifier'

**Input Parameters:**
- **Identifier**: Specify the measurement name to break/pause. For e.g. 'ParameterSweep1'

**Description:** This command breaks/pauses a measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.
Example: The following example breaks a Parameter Sweep measurement with the measurement name identifier called 'ParameterSweep1':

`:PLUG:PSWE:BRE 'ParameterSweep1'

`:PLUGin:PSWEep:STOP

Syntax: :PLUGin:PSWEep:STOP 'identifier'

Input Parameters:

'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Description: This command stops a parameter sweep measurement addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

Example: The following example stops a parameter sweep measurement with the measurement name identifier called 'ParameterSweep1':

`:PLUG:PSWE:STOP 'ParameterSweep1'

`:PLUGin:PSWEep:CONTinue

Syntax: :PLUGin:PSWEep:CONTinue 'identifier'

Input Parameters:

'Identifier': Specify the measurement name to continue. For e.g. 'ParameterSweep1'

Description: This command allows you to continue a Parameter Sweep measurement, addressed by the measurement name identifier, which was stopped using the BREak command.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

Example: The following example continues a parameter sweep measurement with the measurement name identifier called 'ParameterSweep1':

`:PLUG:PSWE: CONT 'ParameterSweep1'
:PLUGin:PSWEep:STEP

Syntax
:PLUGin:PSWEep:STEP 'identifier'

Input
'Identifier': Specify the measurement name to step. For e.g. 'ParameterSweep1'

Parameters

Description
This command allows you to step through a Parameter Sweep measurement, addressed by the measurement name identifier, that was stopped using the BREak command.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

Example
The following example steps a parameter sweep measurement with the measurement name identifier called 'ParameterSweep1':

:PLUG:PSWE:STEP 'ParameterSweep1'

:PLUGin:PSWEep:ACQuisition Subnode

This subnode has the following SCPI structure:

:ACQuisition
  :ALOCation[?]
  :COMPared[?]
  :SFRom[?]
  :SLOCation[?]
  :SMODE[?]
  :SRTime[?]
  :STEP[?]
  :STO[?]
  :SWParameter[?]
  :EBEnabled[?]
  :ERRored[?]
  :TBEratio[?]
  :TCLevel[?]
  :PREBer[?]
  :SAENd[?]
  :PROEditable[?]
This subnode has the following commands:

**Table 89**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SLOCation[?]</td>
<td>:PLUGin:PSWEEP:ACQuisition:SLOCation[?] on page 458</td>
</tr>
<tr>
<td>:SFRom[?]</td>
<td>:PLUGin:PSWEEP:ACQuisition:SFRom[?] on page 460</td>
</tr>
<tr>
<td>:STO[?]</td>
<td>:PLUGin:PSWEEP:ACQuisition:STO[?] on page 460</td>
</tr>
<tr>
<td>:STEP[?]</td>
<td>:PLUGin:PSWEEP:ACQuisition:STEP[?] on page 461</td>
</tr>
<tr>
<td>:COMPared[?]</td>
<td>:PLUGin:PSWEEP:ACQuisition:COMPared[?] on page 461</td>
</tr>
<tr>
<td>:ERRored[?]</td>
<td>:PLUGin:PSWEEP:ACQuisition:ERRored[?] on page 462</td>
</tr>
<tr>
<td>:PREBer[?]</td>
<td>:PLUGin:PSWEEP:ACQuisition:PREBer[?] on page 464</td>
</tr>
<tr>
<td>:PROEditable[?]</td>
<td>:PLUGin:PSWEEP:ACQuisition:PROEditable[?] on page 465</td>
</tr>
</tbody>
</table>
PLUGin:PSWeep:ACQisition:ALOCation?

Syntax:
- `:PLUGin:PSWeep:ACQisition:ALOCation 'Identifier', <location-string>`
- `:PLUGin:PSWeep:ACQisition:ALOCation? 'Identifier'`

Input Parameters:
- `Identifier`: Specify the plugin name. For e.g. 'ParameterSweep1'
- `<location-string>`: 'M*.DataIn*'

Description:
This command sets the location or location group string against which the data acquisition is performed.

This query returns the current location or location group string against which measurement is configured for the data acquisition.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

Example:
`:PLUG:PSWE:ACQ:ALOC 'ParameterSweep1', 'M1.DataIn1'

PLUGin:PSWeep:ACQisition:SLOCation?

Syntax:
- `:PLUGin:PSWeep:ACQisition:SLOCation 'Identifier', <location-string>`
- `:PLUGin:PSWeep:ACQisition:SLOCation? 'Identifier'`

Input Parameters:
- `Identifier`: Specify the plugin name. For e.g. 'ParameterSweep1'
- `<location-string>`: Specify the location/location group string against which the sweep property needs to be performed. For e.g. 'M1.DataIn1'

Description:
This command sets the location/location group string against which the sweep property needs to be performed.

This query returns the current sweep location/location group string from which property is configured for the sweep.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

Example:
`:PLUG:PSWE:ACQ:SLOC 'ParameterSweep1', 'M1.DataIn1'
:PLUGin:PSWep:ACQuisition:SWPArameter[?]

Syntax  
:PLUGin:PSWep:ACQuisition:SWPArameter 'Identifier', <Sweep-Parameter>
:PLUGin:PSWep:ACQuisition:SWPArameter? 'Identifier'

Input Parameters

'Swee}\-Parameter': Specify the plugin name. For e.g. 'ParameterSweep1'

Description
This command sets the sweep parameter to be swept amongst the list of other parameters present in current location or group.

This query returns the current sweep parameter set on which sweep can be performed.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

Example
:PLUG:PSWE:ACQ:SWPA 'ParameterSweep1', 'Delay'

:PLUGin:PSWep:ACQuisition:SMODe[?]

Syntax  
:PLUGin:PSWep:ACQuisition:SMODe 'Identifier', <ABSolute | RELative>
:PLUGin:PSWep:ACQuisition:SMODe? 'Identifier'

Input Parameters

ABS | REL

Description
This command sets the sweep mode. The choices are 'ABSolute' and 'RELative'.

In the 'ABSolute' mode, the property will sweep from the value specified in the 'Sweep From' parameter to a value specified in the 'Sweep To' parameter.

In the 'RELative' mode, the property will be sweep from a value relative to the current value specified in the 'Sweep From' parameter to a value relative to the current value specified in the 'Sweep To' parameter.

This query returns the current sweep mode.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

Example
:PLUG:PSWE:ACQ:SMOD 'ParameterSweep1', ABS
**:PLUGin:PSWEep:ACQuisition:SRTTime**[

**Syntax**
:PLUGin:PSWEep:ACQuisition:SRTTime 'Identifier', <NRf>
:PLUGin:PSWEep:ACQuisition:SRTTime? 'Identifier'

**Input Parameters**
'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

**Description**
This command sets the relaxing time between the sweep before reading the BER. The relaxing time can be between 0 sec to 60 sec.
This query returns the current set relaxing time.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

**Example**
:PLUG:PSWE:ACQ:SRT 'ParameterSweep1', 1E-2

**:PLUGin:PSWEep:ACQuisition:SFRom**[

**Syntax**
:PLUGin:PSWEep:ACQuisition:SFRom 'Identifier', <NRf>
:PLUGin:PSWEep:ACQuisition:SFRom? 'Identifier'

**Input Parameters**
'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

**Description**
This command sets the value from where the sweep needs to be performed for that particular parameter.
This query returns the current sweep value.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

**Example**
:PLUG:PSWE:ACQ:SFR 'ParameterSweep1', 1

**:PLUGin:PSWEep:ACQuisition:STO**[

**Syntax**
:PLUGin:PSWEep:ACQuisition:STO 'Identifier', <NRf>
:PLUGin:PSWEep:ACQuisition:STO? 'Identifier'

**Input Parameters**
'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'
**Description**
This command sets the value till the sweep needs to be performed for that particular parameter. This query returns the current sweep value till the sweep is performed.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

**Example**
:PLUG:PSWE:ACQ:STO 'ParameterSweep1', 1

**:PLUGin:PSWep:ACquisition:STEP[?]**

**Syntax**
:PLUGin:PSWep:ACquisition:STEP 'Identifier', <NRf>
:PLUGin:PSWep:ACquisition:STEP? 'Identifier'

**Input Parameters**
'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

**Description**
This command sets the value as for how many steps user wants to sweep the parameter between the current 'Sweep From' and 'Sweep To' range. This query returns the current number of steps set for the acquisition.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

**Example**
:PLUG:PSWE:ACQ:STEP 'ParameterSweep1', 100

**:PLUGin:PSWep:ACquisition:COMPared[?]**

**Syntax**
:PLUGin:PSWep:ACquisition:COMPared 'Identifier', <NRf>
:PLUGin:PSWep:ACquisition:COMPared? 'Identifier',

**Input Parameters**
'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'
<location-string>: 'M*.DataIn*''

**Description**
This command allows the user to specify the number of compared bits; that would be used as criteria for moving to the next measurement point in case when the sample delay optimization is 'NONE'.
This query returns the present setting.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

**Example**
:PLUG:PSWE:ACQ:COMP 'ParameterSweep1', 1E+7
:PLUGin:PSWEep:ACQuisition:EBEnabled[?]

Syntax :PLUGin:PSWEep:ACQuisition:EBEnabled 'Identifier', ON|OFF|1|0
 :PLUGin:PSWEep:ACQuisition:EBEnabled? 'Identifier'

Input Parameters

'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Description This command enabled/disables no. of errors feature. When this feature is enabled, user can specify additional number of errored bits to consider while moving to the next measurement points.

This query returns the present setting.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

Example :PLUG:PSWE:ACQ:EBE 'ParameterSweep1', ON
 :PLUG:PSWE:ACQ:EBE? 'ParameterSweep1'
 1

:PLUGin:PSWEep:ACQuisition:ERRored[?]

Syntax :PLUGin:PSWEep:ACQuisition:ERRored 'Identifier', <NRf>
 :PLUGin:PSWEep:ACQuisition:ERRored? 'Identifier'

Input Parameters

'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Description This command allows the user to specify the number of errored bits; that would be used as criteria for moving to the next measurement point in case when the sample delay optimization is 'NONE'.

This query returns the present setting.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

Example :PLUG:PSWE:ACQ: ERR 'ParameterSweep1', 1E+5
 :PLUG:PSWE:ACQ: ERR? 'ParameterSweep1'
 1E+5
:PLUGin:PSWEep:ACQuisition:TBERatio[?]

Syntax:

:PLUGin:PSWEep:ACQuisition:TBERatio 'identifier', <NRf>

:PLUGin:PSWEep:ACQuisition:TBERatio? 'identifier'

Input Parameters:

'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

<NRF>: Set the target BER.

Return Range: 1E-18 to 1E-3

Description:

This command sets the target error ratio of the accumulated results. This option is available when accumulation end is set to pass/fail.

This query returns the present setting.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

Example:

:PLUG:PSWE:ACQ:TBER 'ParameterSweep1', 1E-9

:PLUG:PSWE:ACQ:TBER? 'ParameterSweep1'

1E-9

:PLUGin:PSWEep:ACQuisition:TCLevel[?]

Syntax:

:PLUGin:PSWEep:ACQuisition:TCLevel 'identifier', <NRf>

:PLUGin:PSWEep:ACQuisition:TCLevel? 'identifier'

Input Parameters:

'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

<NRF>: Set target confidence level.

Return Range: 0.1% to 99.9%

Description:

This command sets the target confidence level of the accumulated results. This option is available when accumulation end is set to pass/fail.

This query returns the current target confidence level.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

Example:

:PLUG:PSWE:ACQ:TCL 'ParameterSweep1', 95

:PLUG:PSWE:ACQ:TCL? 'ParameterSweep1'

95
[:PLUGin:PSWEep:ACQuisition:PREBer?] Syntax

[:PLUGin:PSWEep:ACQuisition:PREBer 'identifier',
<NONE|DAL|SDOP|TOP|FSP>]

[:PLUGin:PSWEep:ACQuisition:PREBer? 'identifier']

Input Parameters

'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

NONE|DAL|SDOP|TOP|FSP

Return Value

NONE|DAL|SDOP|TOP|FSP

Description

This command sets the pre BER alignment criteria before it sweeps to next measurement point. The choices are:

• None - This parameter does not initiate any pre BER alignment.

• Data Alignment (DAL) - This parameter initiates pattern resynchronization. For details, refer to :DATA:SYNC[:ONCE] on page 529.

• Sample Delay Optimization (SDOP) - This parameter initiates search for the value of data/clock delay that puts the active clock edge in the center of the data eye, midway between the two relative delay points with a measured BER. For details, refer to :INPut:ALIGnment:EYE:TCEN on page 264.

• Threshold Optimization (TOP) - This parameter initiates search for the 0/1 threshold voltage midway between the two 0/1 threshold voltages with a measured BER. For details, refer to :INPut:ALIGnment:EYE:ACEN on page 263.

• Full Sample Point Optimization (FSP) - This parameter initiates auto alignment. For details, refer to :INPut:ALIGnment:EYE[:AUTO] on page 263.

This query returns the present setting.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

Example

[:PLUG:PSWE:ACQ:PREB 'ParameterSweep1', DAL]

[:PLUG:PSWE:ACQ:PREB? 'ParameterSweep1']

DAL
PLUGin:PSWEep:ACQuisition:SAENd

**Syntax:**
```
:PLUGin:PSWEep:ACQuisition:SAENd 'Identifier', <PFA|NOB>
:PLUGin:PSWEep:ACQuisition:SAENd? 'Identifier',
```

**Input Parameters**
- `'Identifier'`: Specify the plugin name. For e.g. 'ParameterSweep1'
- `<PFA | NOB>`

**Description**
This command sets the steps accumulation end parameter as Pass/Fail (PFA) or Number of Bits (NOB).
This query returns the present setting.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

**Example**
```
:PLUG:PSWE:ACQ:SAEN 'ParameterSweep1', PFA
:PLUG:PSWE:ACQ:SAEN? 'ParameterSweep1'
PFA
```

PLUGin:PSWEep:ACQuisition:PROEditable

**Syntax:**
```
:PLUGin:PSWEep:ACQuisition:PROEditable 'Identifier', <ON|OFF|1|0>
:PLUGin:PSWEep:ACQuisition:PROEditable? 'Identifier',
```

**Input Parameters**
- `'Identifier'`: Specify the plugin name. For e.g. 'ParameterSweep1'
- `<ON | OFF | 1 | 0>`

**Return Range**
0 or 1

**Description**
This command enables/disables the acquisition parameters of the Parameter Sweep measurement.
This query returns “0” when the acquisition parameters are in disabled state or the measurement is in running state. This query returns “1” when the acquisition parameters are in enabled state.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

**Example**
```
:PLUG:PSWE:ACQ:PROE 'ParameterSweep1', 1
:PLUG:PSWE:ACQ:PROE? 'ParameterSweep1'
1
```
This subnode has the following SCPI structure:

```
:RUN
  --:MESSage?
  --:PROGress?
  --:STATus?
  --:HISTORY
    [[:STATE][?]]
    --:CLEar
    --:LOG?
```

This subnode has the following commands:

**Table 90**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
</table>
:PLUGin:PSWep:RUN:MESSage?
Syntax :PLUGin:PSWep:RUN:MESSage? 'Identifier'
Input 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'
Parameters
Description This command returns a string describing the state of the parameter sweep addressed by the measurement name identifier. Possible states include; Not Started, Running, Suspended, Finished, Error, or Stopped. This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.
Example :PLUG:PSWE:RUN:MESS? 'ParameterSweep1'

:PLUGin:PSWep:RUN:PROGress?
Syntax :PLUGin:PSWep:RUN:PROGress? 'Identifier'
Input 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'
Parameters
Return Range 0 to 1
Description This command returns a number in the range of 0.0 to 1.0 to indicate the progress of a parameter sweep measurement addressed by the measurement name identifier. 0.0 indicates that the measurement has not started and 1.0 indicates the measurement is finished.
This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.
Example :PLUG:PSWE:RUN:PROG? 'ParameterSweep1'

:PLUGin:PSWep:RUN:STATus?
Syntax :PLUGin:PSWep:RUN:STATus? 'Identifier'
Input 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'
Parameters
Return Range 0|1
Description This command returns the running status of a parameter sweep measurement addressed by the measurement name identifier. 0 indicates
the measurement is not running and a 1 indicates the measurement is running.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

Example :PLUG:PSWE:RUN:STAT? 'ParameterSweep1'

:PLUGin:PSWEep:RUN:HISTory[:STATe][?]

Syntax :PLUGin:PSWEep:RUN:HISTory[:STATe] 'Identifier', <0|1|ON|OFF> 
:PLUGin:PSWEep:RUN:HISTory[:STATe]? 'Identifier'

Input Parameters

'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Return Range 0|1

Description This command enables/disables the storage of parameter sweep measurement results addressed by the measurement name identifier.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

Example :PLUG:PSWE:RUN:HIST 'ParameterSweep1', 1

:PLUGin:PSWEep:RUN:HISTory:CLEar

Syntax :PLUGin:PSWEep:RUN:HISTory:CLEar 'Identifier'

Input Parameters

'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Description This command deletes the parameter sweep measurement history addressed by the measurement name identifier.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

Example :PLUG:PSWE:RUN:HIST:CLE 'ParameterSweep1'
:PLUGin:PSWEep:RUN:LOG?


Input Parameters:
- 'Identifier': Specify the plugin name. For e.g. 'ParameterSweep1'

Description: This command returns logs for the addressed measurement.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

Example: :PLUG:PSWE:RUN:LOG? 'ParameterSweep1'

The format of log returned is in the following way:

#XY Message Log

where X denotes the length of digits for Y, Y denotes the number of characters in the log then followed by the log message.

For example:

#10 means, there is 1 digit after 1 and there is 0 character in the log.

#2492 means, there are 3 digits after 2 and there are 492 characters in the log message.
This subnode has the following commands:

**Table 91**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SGLegends</td>
<td>:PLUGin:PSWEep:SHOW:SGLegends[?] on page 470</td>
</tr>
<tr>
<td>:VSCale</td>
<td>:PLUGin:PSWEep:SHOW:VSCale[?] on page 471</td>
</tr>
<tr>
<td>COMment</td>
<td>:PLUGin:PSWEep:SHOW:COMment[?] on page 472</td>
</tr>
</tbody>
</table>

`:PLUGin:PSWEep:SHOW:SGLegends[?]`

**Syntax**

`:PLUGin:PSWEep:SHOW:SGLegends 'Identifier', <0|1|ON|OFF>`

Input Parameters:
- `Identifier`: Specify the plugin name. For e.g. `ParameterSweep1`
- `<0|1|OFF|ON>`: Specify the show/hide state for graph legends.

**Return Range**

`0|1`

**Description**

This command shows/hides the graph legends.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

**Example**

`:PLUG:PSWE:SHOW:SGL 'ParameterSweep1', 1`
**:PLUGin:PSWEep:SHOW:SMPoints[?]**

**Syntax**: 
- `:PLUGin:PSWEep:SHOW:SMPoints 'Identifier', <0|1|ON|OFF>`
- `:PLUGin:PSWEep:SHOW:SMPoints? 'Identifier'`

**Input Parameters**
- **Identifier**: Specify the plugin name. For e.g. 'ParameterSweep1'
- `<OFF|ON|0|1>`: Disable/enable viewing the measurement points.

**Return Range**: 0|1

**Description**: This command disables/enables viewing of the measurement points in the GUI and associates it with a specific measurement name identifier. A 0 or OFF value disables viewing and a 1 or ON value enables viewing of the measurement points.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

**Example**: `:PLUG:PSWE:SHOW:SMP 'ParameterSweep1', 1`

**:PLUGin:PSWEep:SHOW:VSCale[?]**

**Syntax**: 
- `:PLUGin:PSWEep:SHOW:VSCale 'Identifier', <LOGarithmic | LINear>`

**Input Parameters**
- **Identifier**: Specify the plugin name. For e.g. 'ParameterSweep1'

**Return Range**: LOG | LIN

**Description**: This command specifies the vertical BER axis scale and associates it with a specific measurement name identifier. If LOGarithmic is specified then the BER values are in terms of a logarithmic scale. If LIN is specified then the BER values are in terms of a linear scale.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

**Example**: `:PLUG:PSWE:SHOW:VSC 'ParameterSweep1', LOG`
**:PLUGin:PSWEep:SHOW:COMment[?]**

**Syntax**

`:PLUGin:PSWEep:SHOW:COMment 'Identifier', <"Commnet">`

`:PLUGin:PSWEep:SHOW:COMment? 'Identifier',

**Input Parameters**

Identifier: Specify the plugin name. For e.g. 'ParameterSweep1'

<"Commnet"> - Enter the desired comment.

**Description**

This command is used to enter a comment for each measurement. This comment will be displayed on the Measurement History tab.

This query returns the current setting.

This SCPI is applicable for M8041A, M8051A, M8061A, M8062A, M8045A and M8046A.

**Example**

`:PLUG:PSWE:SHOW:COM 'ParameterSweep1', "Measurement with no. of bits"

`:PLUG:PSWE:SHOW:COM? 'ParameterSweep1'

Measurement with no. of bits
:PLUGin:CCAPture Subnode

The M8020A/M8030A Analyzer captures the data received from the device under test. The captured data bits are displayed in the pattern capture pane in binary or 8b/10b symbol coding. The received data is compared with the expected data and the errored bits/symbols are highlighted. The captured data can be saved for post processing. The maximum capture memory is 2 Gb. However, it also depends on the holdoff length which represents the amount of symbols in which the trigger events will be ignored.

This subnode has the following SCPI structure:

```
:CCAPture
  :CATalog?
  :DELete
  :NEW
  :RESet
  :STARt
  :STOP
  :SAVE
  :TRIGger
    :IMMediate
    [:ONCE]
  :SHOW
    :BIT
    :PATTern
    :VIEW
  :SYMBol
    :PATTern
    :VIEW
  :FETCh
    ...
  :ACQuisition
    ...
  :RUN
    ...
```
This subnode has the following commands and subnodes:

Table 92

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELete[?]</td>
<td>:PLUGin:CCAPture:DELete on page 476</td>
</tr>
<tr>
<td>NEW[?]</td>
<td>:PLUGin:CCAPture:NEW on page 476</td>
</tr>
<tr>
<td>RESet[?]</td>
<td>:PLUGin:JTOlerance:RESet on page 417</td>
</tr>
<tr>
<td>STARt[?]</td>
<td>:PLUGin:CCAPture:STARt on page 477</td>
</tr>
<tr>
<td>STOP[?]</td>
<td>:PLUGin:CCAPture:STOP on page 477</td>
</tr>
<tr>
<td>SAVE[?]</td>
<td>:PLUGin:CCAPture:SAVE on page 477</td>
</tr>
</tbody>
</table>

Subnodes

<table>
<thead>
<tr>
<th>Subnode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FETCH</td>
<td>:PLUGin:CCAPture:FETCh Subnode on page 480</td>
</tr>
<tr>
<td>ACQuisition</td>
<td>:PLUGin:CCAPture:ACQuisition Subnode on page 485</td>
</tr>
<tr>
<td>RUN</td>
<td>:PLUGin:CCAPture:RUN Subnode on page 490</td>
</tr>
</tbody>
</table>
### :PLUGin:CCAPture:CATalog?

<table>
<thead>
<tr>
<th>Syntax</th>
<th>:PLUGin:CCAPtue:CATalog?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>This command returns a list of all created compare and capture measurement names currently available for measuring. This SCPI is applicable for M8041A and M8051A.</td>
</tr>
<tr>
<td>Example</td>
<td>Assume the following is a list of created compare and capture measurement names:</td>
</tr>
</tbody>
</table>

```
:PLUG:CCAP:NEW 'CCAP_1'
:PLUG:CCAP:NEW 'CCAP_2'
:PLUG:CCAP:NEW 'CCAP_3'
```

The command and returned list would look like the following:

```
:PLUG:CCAP:CAT?
"CCAP_1,CCAP_2,CCAP_3"
```
.:PLUGin:CCAPture:DELeTe
Syntax :PLUGin:CCAPture:DELeTe 'identifier'
Input  'identifier': Specify the measurement name to delete.
Parameters
Description This command deletes a previously created compare and capture measurement addressed by the measurement name identifier. This SCPI is applicable for M8041A and M8051A.
Example The following example deletes a compare and capture measurement addressed by the measurement name identifier called 'MyMeasurement':
.:PLUG:CCAP:DEL 'MyMeasurement'

.:PLUGin:CCAPture:NEW
Syntax :PLUGin:CCAPture:NEW 'identifier'
Input  'identifier': Specify the measurement name.
Parameters
Description This command creates a new compare and capture measurement name identifier. This SCPI is applicable for M8041A and M8051A.
Example The following example creates a compare and capture measurement name identifier called 'MyMeasurement':
.:PLUG:CCAP:NEW 'MyMeasurement'

.:PLUGin:CCAPture:RESet
Syntax :PLUGin:CCAPture:RESet 'identifier'
Input  'identifier': Specify the measurement name to reset.
Parameters
Description This command resets a compare and capture measurement addressed by the measurement name identifier. This SCPI is applicable for M8041A and M8051A.
Example The following example resets a compare and capture measurement addressed with the measurement name identifier called 'MyMeasurement':
.:PLUG:CCAP:RES 'MyMeasurement'
:PLUGin:CCAPtue:STARt

Syntax  
:PLUGin:CCAPtue:STARt 'identifier'

Input  
'identifier': Specify the measurement name to start.

Parameters

Description  
This command starts a compare and capture measurement addressed by the measurement name identifier. This SCPI is applicable for M8041A and M8051A.

Example  
The following example starts a compare and capture measurement addressed with the measurement name identifier called 'MyMeasurement':

:PLUG:CCAP:STAR 'MyMeasurement'

:PLUGin:CCAPtue:STOP

Syntax  
:PLUGin:CCAPtue:STOP 'identifier'

Input  
'identifier': Specify the measurement name to stop.

Parameters

Description  
This command stops the capturing of data and data is not uploaded into the measurement 'plug-in'. This SCPI is applicable for M8041A and M8051A.

Example  
The following example stops a compare and capture measurement with the measurement name identifier called 'MyMeasurement':

:PLUG:CCAP:STOP 'MyMeasurement'

:PLUGin:CCAPtue:SAVE

Syntax  
:PLUGin:CCAPtue:SAVE 'identifier','Location','pattern-path' [overwrite-flag]

Input  
'identifier': Specify the measurement name to save.

Parameters  
'Location': 'M*.DataIn1' or 'M*.DataIn2'

'pattern-path': Patterns can be saved as data and are stored in three different locations:

1 Local to current setting ("current:")

2 Shared between settings ("shared:")

3 Factory supplied standard patterns ("factory:"). These patterns are read only and cannot be modified.
[:<overwrite-flag>]: ON | OFF | 0 | 1
Optional flag is a boolean parameter which specifies if an existing pattern file should be overwritten or not.

Description
The currently uploaded pattern is stored into the specified file on the hard disk.
This SCPI is applicable for M8041A and M8051A.

Example
The following example saves a compare and capture measurement with the measurement name identifier called 'MyMeasurement':

:PLUGin:CCAPture:TRIGger:IMMediate[:ONCE]

Syntax :PLUGin:CCAPture:TRIGger:IMMediate[:ONCE]

Input Parameters
'identifier': Specify the measurement name.
'Location': 'M*.DataIn1' or 'M*.DataIn2'

Description
This command simulates a stop event. Compare and capture is stopped and the recorded data is uploaded into the measurement.
This SCPI is applicable for M8041A and M8051A.

Example
PLUGin:CCAPture:TRIGger:IMMediate:ONCE 'MyMeasurement'

:PLUGin:CCAPture:SHOW:BIT[PATTern]][:VIEW][:?]


Input Parameters
'identifier': Specify the measurement name.
'Mode': BIN|HEX

Return Range BIN|HEX

Description
This command selects the bit pattern view mode viz. BIN|HEX.
This SCPI is applicable for M8041A and M8051A.

Example
:PLUGin:CCAPture:SHOW:SYMBol[:PATTern][:VIEW][?]

Syntax:

Input:
'identifier': Specify the measurement name.

Parameters:
'Mode': BIN|HEX|SYMBol

Return Range:
BIN|HEX|SYMBol

Description:
This command selects 8/10 bit pattern view mode viz. BIN|HEX|SYMBol.
This SCPI is applicable for M8041A and M8051A.

Example:
This subnode has the following SCPI structure:

```
:FETCh
  - :ERRor
  - :COUNt
  - :PATTern
    - :DATA
      - :HOLDoff?
      - :DEPTh?
    - :ERRor
    - :SYMBol
      - :FIRSt?
      - :LAST?
  - RUN
  - [:STATus]?
```
This subnode has the following commands:

**Table 93**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:RUN[:STATus]?</td>
<td>:PLUGin:CCAPture:FETCh:RUN[:STATus]? on page 484</td>
</tr>
</tbody>
</table>

:PLUGin:CCAPture:FETCh:ERRor:COUNT?

**Syntax**
```
:PLUGin:CCAPture:FETCh:ERRor:COUNT? 'identifier' [, 'Location']
```

**Input Parameters**
- 'identifier': Specify the measurement name.
- ['Location']: 'M*.DataIn1' or 'M*.DataIn2'

**Description**
This command gives the error count of the captured data. If simply run without specifying the location, then it returns the results for the current run.

This SCPI is applicable for M8041A and M8051A.

**Example**
```
:PLUGin:CCAPture:FETCh:ERRor:COUNT? 'MyMeasurement', 'M1.DataIn1'
```
`:PLUGin:CCAPture:FETCh:PATTern:[DATA]:HOLDoff?`

**Syntax**
```
:PLUGin:CCAPture:FETCh:PATTern:[DATA]:HOLDoff? 'identifier' ['Location']
```

**Input**
- 'identifier': Specify the measurement name.
- ['Location']: 'M*.DataIn1' or 'M*.DataIn2'

**Description**
This command returns the actual holdoff i.e. number of data bits captured before the trigger event. Returned value of this query is only valid when capture measurement is finished.

This SCPI is applicable for M8041A and M8051A.

**Example**
```
```

**Return**
```
("M1.DataIn1", ...)
```

`:PLUGin:CCAPture:FETCh:PATTern:[DATA]:DEPTh?`

**Syntax**
```
:PLUGin:CCAPture:FETCh:PATTern:[DATA]:DEPTh? 'identifier' ['Location']
```

**Input**
- 'identifier': Specify the measurement name
- ['Location']: 'M*.DataIn1' or 'M*.DataIn2'

**Description**
This command returns the total number of data bits captured. Returned value of this query is only valid when capture measurement is finished.

This SCPI is applicable for M8041A and M8051A.

**Example**
```
```

**Return**
```
("M1.DataIn1", ...)
```
:PLUGin:CCAPture:FETCh:PATTern:ERRor:SYMBol:FIRSt?

Syntax
:PLUGin:CCAPture:FETCh:PATTern:ERRor:SYMBol:FIRSt?
   'identifier' [,'Location']

Input
'identifier': Specify the measurement name.

Parameters
[,"Location']: 'M*.DataIn1' or 'M*.DataIn2'

Description
This command fetches the first error symbol.

Example
:PLUGin:CCAPture:FETCh:PATTern:ERRor:SYMBol:FIRSt?
   'MyMeasurement','M1.DataIn1'

Return: ("M1.DataIn1", ...)

:PLUGin:CCAPture:FETCh:PATTern:ERRor:SYMBol:LAST?

Syntax

Input
'identifier': Specify the measurement name.

Parameters
[,"Location']: 'M*.DataIn1' or 'M*.DataIn2'

Description
This command fetches the last error symbol.

Example
:PLUGin:CCAPture:FETCh:PATTern:ERRor:SYMBol:LAST?
   'MyMeasurement','M1.DataIn1'

Return: ("M1.DataIn1", ...)
:PLUGin:CCAPture:FETCh:RUN[:STATus]?

Syntax :PLUGin:CCAPture:FETCh:RUN[:STATus]? 'identifier' [, 'Location']

Parameters
- 'identifier': Specify the measurement name.
- ['Location']: 'M*.DataIn1' or 'M*.DataIn2'

Description This command returns the capture measurement status for each analyzer location. Returned value can be any of the following:

- **STOP**: Capture measurement is stopped or not currently running. No valid data is available in capture memory.
- **STAR**: Capture measurement is starting. No valid data is available in capture memory.
- **ARM**: Capture measurement is started (Armed) and waiting for trigger. No valid data is available in capture memory.
- **TRIG**: Capture logic is triggered and data is being captured. Valid data will be available soon in capture memory.
- **UPL**: Data capturing is done and data is being uploaded and analyzed. Valid data is available in captured memory.
- **FIN**: Capture measurement is finished.

This SCPI is applicable for M8041A and M8051A.


Return: ('M1.DataIn1', ...)


This subnode has the following SCPI structure:

```
:ACQUision
  --:ALOCation[?]
  --:DEPTh[?]
  --:HOLDoff[?]
  --:SOURce[?]
  --:CIMemory[?]
  --:CAPOnly[?]
  --:PROEditable[?]
```

This subnode has the following commands:

**Table 94**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:DEPTh[?]</td>
<td>:PLUGin:CCAPture:ACQuisition:DEPTh[?] on page 486</td>
</tr>
<tr>
<td>:HOLDoff[?]</td>
<td>:PLUGin:CCAPture:ACQuisition:HOLDoff[?] on page 486</td>
</tr>
<tr>
<td>:SOURce[?]</td>
<td>:PLUGin:CCAPture:ACQuisition:SOURce[?] on page 487</td>
</tr>
<tr>
<td>:SLOPe[?]</td>
<td>:PLUGin:CCAPture:ACQuisition:SLOPe[?] on page 487</td>
</tr>
<tr>
<td>:CIMemory[?]</td>
<td>:PLUGin:CCAPture:ACQuisition:CIMemory[?] on page 488</td>
</tr>
<tr>
<td>:PROEditable[?]</td>
<td>:PLUGin:CCAPture:ACQuisition:PROEditable[?] on page 489</td>
</tr>
</tbody>
</table>
:PLUGIn:CCAPture:ACQuisition:ALOCation[?]

Syntax  :PLUGIn:CCAPture:ACQuisition:LOCation 'identifier','Location'
         :PLUGIn:CCAPture:ACQuisition:LOCation? 'identifier'

Input  'identifier': Specify the measurement name.

Parameters

[,'Location']': 'M*.DataIn1' or 'M*.DataIn2'

Description Specified the location the compare & capture measurement cooperates with. Examples for a location are 'M1.DataIn1', 'M3.DataIn2',... or a name of an existing group of Datalns.

This SCPI is applicable for M8041A, M8051A, M8045A and M8046A.

Example  PLUGIn:CCAPture:ACQuisition:ALOCation 'My Measurement', 'M1.DataIn1'

:PLUGIn:CCAPture:ACQuisition:DEPTh[?]

Syntax  :PLUGIn:CCAPture:ACQuisition:DEPTh 'identifier',<NRf>
         :PLUGIn:CCAPture:ACQuisition:DEPTh? 'identifier'

Input  'identifier': Specify the measurement name.

Parameters

Range  0 to 2147483648

Description This command configures how long the capture logic will write the capture memory after detecting the trigger event.

This SCPI is applicable for M8041A, M8051A, M8045A and M8046A.

Example  :PLUGIn:CCAPture:ACQuisition:DEPTh 'My Measurement', 1

:PLUGIn:CCAPture:ACQuisition:HOLDoff[?]

Syntax  :PLUGIn:CCAPture:ACQuisition:HOLDoff 'identifier',<NRf>
         :PLUGIn:CCAPture:ACQuisition:HOLDoff? 'identifier'

Input  'identifier': Specify the measurement name.

Parameters

Range  0 to 2147483648

Description Specifies the amount of bits the trigger event is ignored after starting capturing of data.
This SCPI is applicable for M8041A, M8051A, M8045A and M8046A.

Example :PLUGin:CCAPture:ACQuisition:HOLDoff 'My Measurement', 1


Input 'identifier': Specify the measurement name.

Parameters <CINA | CINB | ERRor | IMM>

Return Range CINA | CINB | ERRor | IMM

Description Specifies the event for triggering the compare & capture measurement. CTRL IN A or CRTL IN B or an ERRor occurred can be chosen to trigger the capture recording. IMM can be used to capture data without any trigger.

This SCPI is applicable for M8041A, M8051A, M8045A and M8046A.

Example :PLUGin:CCAPture:ACQuisition:SOURce 'My Measurement',CINA


Input 'identifier': Specify the measurement name.

Parameters <POS | NEG>

Return Range POS | NEG

Description If CINA or CINB is selected for triggering the capture process this command specifies the slope the trigger react with.

This SCPI is applicable for M8041A, M8051A, M8045A and M8046A.

Example :PLUGin:CCAPture:ACQuisition:SLOPe 'My Measurement', POS
:PLUG:CCAP:ACQ:CIM

Syntax :PLUG:CCAP:ACQ:CIM 'identifier', [ON|OFF][1|0]

Parameters
- 'identifier': Specify the measurement name.
- [ON|OFF][1|0]

Input Parameters
- [ON|OFF][1|0]

Return Range 1 | 0

Description This commands allows the user to capture the data in memory.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example :PLUG:CCAP:ACQ:CIM 'My Measurement', ON
:PLUG:CCAP:ACQ:CIM? 'My Measurement'
1

:PLUG:CCAP:ACQ:CAPO

Syntax :PLUG:CCAP:ACQ:CAPO 'identifier', [ON|OFF][1|0]

Parameters
- 'identifier': Specify the measurement name.
- [ON|OFF][1|0]

Return Range 1 | 0

Description This commands allows the user to capture data without any comparison.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example :PLUG:CCAP:ACQ:CAPO 'My Measurement', ON
:PLUG:CCAP:ACQ:CAPO? 'My Measurement'
1
:PLUGin:CCAPture:ACQuisition:PROEditable[?]

Syntax
:PLUGin:CCAPtue:ACQuisition:PROEditable 'identifier', <ON|OFF|1|0>
:PLUGin:CCAPture:ACQuisition:PROEditable? 'identifier'

Input
'identifier': Specify the measurement name.

Parameters
ON | OFF | 1 | 0

Return Range
0 or 1

Description
This command enables/disables the acquisition parameters of the compare and capture measurement.

This query returns “0” when the acquisition parameters are in disabled state or the measurement is in running state. This query returns “1” when the acquisition parameters are in enabled state.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

Example
:PLUG:CCAP:ACQ:PROE 'MyMeasurement', 1
:PLUG:CCAP:ACQ:PROE? 'MyMeasurement', 1
This subnode has the following SCPI structure:

```
:RUN
  - :MESSage?
  - :PROGress?
  - :STATus?
  - :LOG?
```

This subnode has the following commands:

**Table 95**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
</table>
:PLUGin:CCAPture:RUN:MESSage?


Input Parameters: 'identifier': Specify the measurement name.

Description: This command returns a string describing the state of the compare & capture measurement. Possible states include NotStarted, Running, Stopped and Triggered. This SCPI is applicable for M8041A and M8051A.

Example: The following example returns the state of a compare and capture measurement with the measurement name identifier called 'MyMeasurement':

:PLUG:CCAP:RUN:MESS? 'MyMeasurement'

Running

:PLUGin:CCAPture:RUN:PROGress?


Input Parameters: 'identifier': Specify the measurement name.

Return Range: 0.0 to 1.0

Description: This command returns a number in the range of 0.0 to 1.0 to indicate the progress of a compare and capture measurement addressed by the measurement name identifier. A 0.0 indicates that the measurement has not started and 1.0 indicates the measurement is finished. This SCPI is applicable for M8041A and M8051A.

Example: The following example returns the progress of a compare and capture measurement with the measurement name identifier called 'MyMeasurement':

:PLUG:CCAP:RUN:PROG? 'MyMeasurement'

0.51
**:PLUGin:CCAPture:RUN:STATus?**

**Syntax**

`:PLUGin:CCAPture:RUN:STATus? 'identifier'`

**Input Parameters**

- `'identifier'`: Specify the measurement name.

**Return Range** 0|1

**Description**

This command returns the running status of the compare & capture measurement. A 0 indicates the measurement is not running and a 1 indicates the measurement is running.

This SCPI is applicable for M8041A and M8051A.

**Example**

The following example returns the running status of a compare & capture measurement with the measurement name identifier called 'MyMeasurement':

`:PLUG:CCAP:RUN? 'MyMeasurement'

---

**:PLUGin:CCAPture:RUN:LOG?**

**Syntax**

`:PLUGin:CCAPture:RUN:LOG? 'identifier'`

**Input Parameters**

- `'identifier'`: Specify the measurement name.

**Description**

This command returns the logs for the addressed measurement.

**Example**

The following example returns the logs for the measurement name identifier called 'MyMeasurement':


The format of log returned is in the following way:

```
#XY Message Log
```

where X denotes the length of digits for Y, Y denotes the number of characters in the log then followed by the log message.

For example:

- #10 means, there is 1 digit after 1 and there is 0 character in the log.
- #2492 means, there are 3 digits after 2 and there are 492 characters in the log message.
SYSTem Subsystem

The SYSTem subsystem is used for general system functions.

This subsystem has the following SCPI structure:

```
:SYSTem
  :INSTRument
  :RESet
  :GROup
  -DEFine
  -DELeete
  -MEMBer?
  -[LIST]?  
  -LTraining
  -[LOG]?
  -[LOG]
  -CLEar
  -BLOCk
  -EQUalization
  -[STATUs]?
  :INFormation
  -[INSTRument]
  -IDENTifier?
  -LOCations?
  -GROUPs
  -USER?
  -DEtails
  -JSON?
  -VALUE?
  -XML?
  -LICenses
  -HOST:[LIST]?
  -MODules:[LIST]?
  :COMMunicate
  -TCPip
  -CONTrol?
  :DCINterface
  -[PROGRAM]
  -LOAD
  -UNLoad
  -SELected?
  -EXECute[:INIT]
  -DEVice
  -CONNect[?]?
  -MODE[?]?
  -LANE
  -MODE[?]?
  :RESet
  -[ALL]
  -INSTRument
```
This subsystem has the following commands:

Table 96

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:INSTrument:RESet</td>
<td>:SYstem:INSTrument:RESet on page 496</td>
</tr>
<tr>
<td>:INSTrument:GROup:DEFine</td>
<td>:SYstem:INSTrument:GROup:DEFine on page 496</td>
</tr>
<tr>
<td>:INSTrument:GROup:DELete</td>
<td>:SYstem:INSTrument:GROup:DELete on page 496</td>
</tr>
<tr>
<td>:INSTrument:GROup[:LIST]?</td>
<td>:SYstem:INSTrument:GROup[:LIST]? on page 497</td>
</tr>
<tr>
<td>:INSTrument:LTRaining[:LOG]?</td>
<td>:SYstem:INSTrument:LTRaining[:LOG]? on page 498</td>
</tr>
<tr>
<td>:INSTrument:LTRaining[:LOG]:CLEar</td>
<td>:SYstem:INSTrument:LTRaining[:LOG]:CLEar on page 499</td>
</tr>
<tr>
<td>:INSTrument:LTRaining[:LOG]:EQUalization[:STATus]?</td>
<td>:SYstem:INSTrument:LTRaining[:LOG]:EQUalization[:STATus]? on page 500</td>
</tr>
<tr>
<td>:INFormation[:INSTrument]:IDENtifier?</td>
<td>:SYstem:INFormation[:INSTrument]:IDENti fier? on page 501</td>
</tr>
<tr>
<td>:INFormation[:INSTrument]:DETail:VAue?</td>
<td>:SYstem:INFormation[:INSTrument]:DETail:VAue? on page 502</td>
</tr>
<tr>
<td>Name</td>
<td>Description under</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>:DCINterface[:PROgram]:LOAD</td>
<td>:SYSTem:DCINterface[:PROgram]:LOAD on page 505</td>
</tr>
<tr>
<td>:DCINterface[:PROgram]:UNLoad</td>
<td>:SYSTem:DCINterface[:PROgram]:UNLoad on page 506</td>
</tr>
<tr>
<td>:RESet:INSTrument</td>
<td>:SYSTem:RESet:INSTrument on page 510</td>
</tr>
</tbody>
</table>
:SYSTem:INStrument:RESet
Syntax   :SYSTem:INStrument:RESet
Description This command resets the instrument state to its default values. This command does not influence the SCPI status system as *RST does.
Example   :SYST:INST:RES

:SYSTem:INStrument:GROup:DEFine
Syntax   :SYSTem:INStrument:GROup:DEFine
   'GroupName','identifier','identifier','identifier','...
Input Parameters
   'GroupName': Assign a name to this group of location identifiers.
   'identifier': Define the group of location identifiers.
Description This command defines a group of location identifiers under one group name. All properties belonging to these location identifiers can be addressed simultaneously using the group name.
          VOLT:AMPL 'Outputs',0.05

NOTE If you define a group name that already exists, the existing name with its group of defined location identifiers will be redefined without notification.

Example   :SYST:INST:GRO:DEL 'Outputs'

:SYSTem:INStrument:GROup:DELeTe
Syntax   :SYSTem:INStrument:GROup:DELeTe 'GroupName'
Input Parameters
   'GroupName': Specify group name to delete.
Description This command deletes the specified group.
Example   :SYST:INST:GRO:DEL 'Outputs'
**:SYSTem:INSTRument:GROup:MEMBer?**

**Syntax**
`:SYSTem:INSTRument:GROup:MEMBer? 'GroupName'

**Input**
'GroupName': Specify group name.

**Parameters**

**Description**
This command returns the comma separated list of member names contained within the specified group.

**Example**
`:SYST:INST:GRO:MEMB? 'Outputs'
"M1.DataOut1", "M1.DataOut2"

**:SYSTem:INSTRument:GROup[:LIST]?**

**Syntax**
`:SYSTem:INSTRument:GROup[:LIST]?

**Description**
This command returns a comma separated list of all group names.

**Example**
`:SYST:INST:GRO?
"Outputs"
:SYSTem:INSTrument:LTRaining[:LOG]?

Syntax :SYSTem:INSTrument:LTRaining[:LOG]?

Description This command returns accumulated Link Training logs for the selected DataOut locations in text format.

Example :SYST:INST:LTR:LOG?


<table>
<thead>
<tr>
<th>State</th>
<th>Execution Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detect.Active</td>
<td>1.04864 ms</td>
</tr>
<tr>
<td>Polling.Active</td>
<td>2.064992 ms</td>
</tr>
<tr>
<td>Polling.Configuration</td>
<td>152.112 us</td>
</tr>
<tr>
<td>Configuration.Linewidth.Start</td>
<td>2.896 us</td>
</tr>
<tr>
<td>Configuration.Linewidth.Accept</td>
<td>304 ns</td>
</tr>
<tr>
<td>Configuration.Lanenum.Wait</td>
<td>2.8 us</td>
</tr>
<tr>
<td>Configuration.Lanenum.Accept</td>
<td>544 ns</td>
</tr>
<tr>
<td>Configuration.Complete</td>
<td>3.696 us</td>
</tr>
<tr>
<td>Configuration.Idle</td>
<td>256 ns</td>
</tr>
<tr>
<td>L0</td>
<td>224 ns</td>
</tr>
<tr>
<td>Recovery.RcvrLock</td>
<td>3.52 us</td>
</tr>
<tr>
<td>Recovery.RcvrCfg</td>
<td>2.464 us</td>
</tr>
<tr>
<td>Recovery.Speed</td>
<td>8.464 us</td>
</tr>
<tr>
<td>Recovery.RcvrLock</td>
<td>448 ns</td>
</tr>
<tr>
<td>Recovery.Equalization.Phase1</td>
<td>2.20608 ms</td>
</tr>
<tr>
<td>Recovery.Equalization.Phase2</td>
<td>21.033456 ms</td>
</tr>
<tr>
<td>Recovery.Equalization.Phase3</td>
<td>1.904 us</td>
</tr>
<tr>
<td>Recovery.RcvrLock</td>
<td>2.016 us</td>
</tr>
<tr>
<td>Recovery.RcvrCfg</td>
<td>656 ns</td>
</tr>
<tr>
<td>Recovery.Idle</td>
<td>112 ns</td>
</tr>
<tr>
<td>Loopback.Entry</td>
<td>2.816 us</td>
</tr>
<tr>
<td>Loopback.Active</td>
<td>-</td>
</tr>
</tbody>
</table>

BERT Tx Equalization

<table>
<thead>
<tr>
<th>Accept</th>
<th>PresetNumber</th>
<th>PreCursor</th>
<th>MainCursor</th>
<th>PostCursor</th>
<th>FullSwing</th>
<th>LowFrequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>P7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>24</td>
<td>8</td>
</tr>
<tr>
<td>True</td>
<td>P4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>24</td>
<td>8</td>
</tr>
</tbody>
</table>

DUT Tx Equalization

<table>
<thead>
<tr>
<th>Event</th>
<th>Accept</th>
<th>PresetNumber</th>
<th>PreCursor</th>
<th>MainCursor</th>
<th>PostCursor</th>
<th>FullSwing</th>
<th>LowFrequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>-</td>
<td>P7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>40</td>
<td>12</td>
</tr>
<tr>
<td>Target</td>
<td>-</td>
<td>P4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>40</td>
<td>12</td>
</tr>
<tr>
<td>Reported</td>
<td>True</td>
<td>P4</td>
<td>0</td>
<td>40</td>
<td>0</td>
<td>40</td>
<td>12</td>
</tr>
</tbody>
</table>

Keysight M8000 Series of BER Test Solutions Programming Guide
:SYSTem:INSTRument:LTRaining[:LOG]:CLEar

Syntax    :SYSTem:INSTRument:LTRaining[:LOG]:CLEar
Description This command clears the Link Training logs.
Example    :SYST:INST:LTR:LOG:CLE

:SYSTem:INSTRument:LTRaining[:LOG]:BLOCK?

Syntax    :SYSTem:INSTRument:LTRaining[:LOG]:BLOCK?
Description The content of the log file is returned as definite block. (see :
SYSTem:INSTRument:LTRaining[:LOG]?) Control character (e.g. Line Feed) are
contained in the log file. This can interrupt the transfer to the host computer. A
definite block is used to transfer data streams that can contain special characters. see
SCPI language definition or IEEE 488.2 -BLOCK PROGRAM DATA.
Example    :SYSTem:INSTRument:LTRaining:LOG:BLOCk?
**:SYSTem:INStrument:LTRaining:EQUalization[:STATus]?**

**Syntax**: 
**:SYSTem:INStrument:LTRaining:EQUalization[:STATus]? 'identifier'**

**Input Parameters**

- **'identifier'**: 'M*.DataOut1' or 'M*.DataOut2'

**Description**

This query returns the Dut transmitter equalization information from the link up block. The results are combined to an expression. An expression is comparable to a struct in a programming language. It's a set of parameter of different types separated by a comma and enclosed by parenthesis. The returned expression contains the location and a set of values.

(*Location", Accepted, PreCursor, MainCursor, PostCursor, FullSwing,
LowFrequency)

- **Location**: It's a string referred to the place the Dut tx equalization information is located. In this case it's either M*.DataOut1 or M*.DataOut2. This parameter is important if a group name is used as 'identifier'.
- **Accepted**: Requested coefficients are accepted or rejected, either 0 or 1.
- **PreCursor**: Received pre-cursor coefficient value during Phase 2 or 3 of the equalization. If no data is present, 9.91 E 37 is returned.
- **MainCursor**: Received data-cursor coefficient value during Phase 2 or 3 of the equalization. If no data is present, 9.91 E 37 is returned.
- **PostCursor**: Received post-cursor coefficient value during Phase 2 or 3 of the equalization. If no data is present, 9.91 E 37 is returned.
- **FullSwing**: Full swing value received during Phase 1 of the equalization. If no data is present, 9.91 E 37 is returned.
- **LowFrequency**: Low Frequency value received during Phase 1 of the equalization. If no data is present, 9.91 E 37 is returned.

If this query is sent to a ‘group’ of inputs the response will be expanded by additional sets of counted values. The expression contains for every input belonging to a group a separate set of values starting with the location.

**Example**: 
**:SYST:INST:LTR:EQU:STAT? 'M1.DataOut1'**
:SYSTem:INFormation[:INStrument]:IDENtifier?

Syntax    :SYSTem:INFormation[:INStrument]:IDENtifier?
Description This command returns a list of all identifiers defined in the system.
Example    :SYST:INF:IDEN?

:SYSTem:INFormation[:INStrument]:LOCations?

Syntax    :SYSTem:INFormation[:INStrument]:LOCations?
Description This command returns a list of all locations defined in the system.
Example    :SYST:INF:LOC?
(ClkGen,System,RefClkOut,TrigOut,SysOutA,SysOutB,CtrlOutA,SysInA,SysInB,CtrlInA,CtrlInB,DataIn1,DataIn2,DataOut1,DataOut2,Simulation1,Simulation2,MuxMode,ElectIdleIn)

:SYSTem:INFormation[:INStrument]:GROups:USER?

Syntax    :SYSTem:INFormation[:INStrument]:GROups:USER?
Description This command returns a list of all user defined parameter group names currently in use by the system.
Example    :SYST:INF:GRO:USER?
(Outputs,Inputs)

:SYSTem:INFormation[:INStrument]:DETails subnode

The following commands provide information about a specific part of the instrument.

The returned information contains
- Identifier
- Product number
- Connection status (Online/Offline)
List of all options
User friendly name of the instrument
Serial number
VISA resource string
Software Version
All the above information for sub-components like remote heads, or external clock recovery modules

`:SYSTem:INFormation[:INStrument]:DETails:JSON?

Syntax :SYSTem:INFormation[:INStrument]:DETails:JSON? 'Identifier'

Description This command returns the instrument details in JSON (JavaScript Object Notation) data structure.

Input Parameters
Identifier - Any module name. For example; 'M1', 'M2'

Return Value Returns the sequence of following values:
Identifier, ProductNumber, Connection, Options, FriendlyName of module, SerialNumber, VisaResource, SoftwareVersion, Subcomponents

Example :SYST:INF:INST:DET:JSON? 'M2'
{"Identifier":"M2","ProductNumber":"M8046A","Connection":"Offline","Options":["A32","A64","0P3","0P6","0A9","U64","UP3","UP6","UA9"],"FriendlyName":"M8046A 64GBd Error Analyzer","SerialNumber":"DE5250000003","VisaResource":"OFFLINE::211::INSTR","SoftwareVersion":"5.0.125.1","Subcomponents":[]}

`:SYSTem:INFormation[:INStrument]:DETails[:VALue]?

Syntax :SYSTem:INFormation[:INStrument]:DETails[:VALue]? 'Identifier'

Description This command returns the information as SCPI expression. Individual fields are separated by comma, and sub-structures are enclosed in round brackets.

Input Parameters
Identifier - Any module name. For example; 'M1', 'M2'
Return Value
Returns the instrument details value in the following sequence:
(identifier, product-number, connection-status, (installed-options),
friendly-name, serial-number, VISA-resource, software-version,
((sub-component-info-1), (sub-component-info-2), ...)
(sub-component-info-n))

Example
:SYST:INF:INST:DET:VAL? 'M2'
('M2', 'M8046A', 'Offline', (A32, A64, 0P3, 0P6, 0A9, U64, UP6, U64), 'M8046
A 64GBd Error Analyzer', 'DE5250000003', 'OFFLINE::211::INSTR',
'5.0.125.1', ()

:SYSTem:INFormation[:INSTrument]:DETails:XML?

Syntax
:SYSTem:INFormation[:INSTrument]:DETails:XML? 'Identifier'

Description
This command returns the instrument details in XML format.

Input
Parameters
Identifier - Any module name. For example; 'M1', 'M2'

Return Value
Returns the following instrument details in XML format:
Identifier, Product Number, Connection type, License options, friendly
name of module, Serial Number, Visa resource, Software version,
Subcomponents

Example
<?xml version="1.0" encoding="utf-16"?>
<DetailedComponentInfo
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><Identifier
M2</Identifier><ProductNumber>M8046A</ProductNumber><Connectio
n>Offline</Connection><Options>A32, A64, 0P3, 0P6, 0A9, U64, UP6, U64
<Options><FriendlyName>M8046A 64GBd Error
Analyzer</FriendlyName><SerialNumber>DE5250000003</SerialNumber>
<VisaResource>OFFLINE::211::INSTR</VisaResource><SoftwareVersion
>5.0.125.1</SoftwareVersion></DetailedComponentInfo>
**SYSTem:INFormation:LICenses:HOST[:LIST]?**

**Syntax**

```
SYSTem:INFormation:LICenses:HOST[:LIST]?
```

**Description**

This command lists the description of all the host licenses supported by the device.

**Example**

```
SYSTem:INFormation:LICenses:HOST[:LIST]?

(PSERNO, "AB99889386", "M8070A-0TP", "0TP", "1.0", "System Software for M8000 Series of BER Test Solutions, transportable, perpetual license", "Not installed", "Perpetual", "12/31/9999 11:59:59 PM", "clockgroup-wide")
(PSERNO, "AB99889386", "M8085A-CT1", "CT1", "1.0", "MIPI C-PHY Editor for M819xA AWG, Transportable, Perpetual License", "Not installed", "Perpetual", "12/31/9999 11:59:59 PM", "clockgroup-wide")
(PSERNO, "AB99889386", "M8085A-CN1", "CN1", "1.0", "MIPI C-PHY Editor for M819xA AWG, Network/Floating, Perpetual License", "Not installed", "Perpetual", "12/31/9999 11:59:59 PM", "clockgroup-wide")
(PSERNO, "AB99889386", "M8085A-CD1", "CD1", "1.0", "MIPI C-PHY Editor for M819xA AWG, Temporary License", "Not installed", "Perpetual", "12/31/9999 11:59:59 PM", "clockgroup-wide")
(PSERNO, "AB99889386", "M8085A-DT1", "DT1", "1.0", "MIPI D-PHY Editor for M819xA AWG, Transportable, Perpetual License", "Not installed", "Perpetual", "12/31/9999 11:59:59 PM", "clockgroup-wide")
``
::SYSTem::INFormation::LICenses::MODules[:LIST]?

Syntax ::SYSTem::INFormation::LICenses::MODules[:LIST]?

Description This command lists the description of all the module licenses supported by the device.

Example ::SYSTem::INFormation::LICenses::MODules::LIST?

::SYSTem::COMMunicate::TCPip::CONTrol?

Syntax ::SYSTem::COMMunicate::TCPip::CONTrol?

Description This command implements SCPI over TCP with Control Connection.

The Keysight Automation Advantage (AAA) standard requires that instruments that support SCPI also support SCPI traffic over TCP with a control connection. This control channel is set up by sending SYSTem:COMMunicate:TCPip:CONTrol? to the instrument. To use the control port:

- Open a socket connection to port 5025, the standard SCPI-over-sockets port.
- Over that socket connection, issue the SYST:COMM:TCP:CONT? query to retrieve the control port.
- Open a second socket connection to the control port.

Now you can send control commands: for example, send "DCL\n" on the control port to issue a device clear.

Example ::SYST:COMM:TCP:CONT?

::SYSTem::DCINterface[:PROGram]:LOAD

Syntax ::SYSTem::DCINterface[:PROGram]:LOAD "PathToPythonProgramFile"

Description This commands loads the given Python script into the DUT control interface. This DUT control interface program can be used to read the bit error rate of an externally connected DUT and provide these to the M8070A software as if the counters are part of the instrument. These values can then be used by the built in Bit Error Ratio measurement or Jitter Tolerance measurement.

In addition the script can implement functions that extend the measurements in order to adapt the measurement to the specific test setup and device under test.
Python scripts can be part of settings and shared areas. The addressing scheme looks like:

"Factory/HookExample.py" // Read only factory shared area
"Shared/HookExample.py" // User shared area
"Current/HookExample.py" // Current setting

Example

:SYSTem:DCInterface:PROGram:LOAD
"factory/DutCounterIntegration.py"

:SYSTem:DCInterface[:PROGram]:UNLoad

Syntax :SYSTem:DCInterface[:PROGram]:UNLoad

Description This command unloads the currently used DUT Control Interface script and all the resources that the script integrated into the system.

Example :SYSTem:DCInterface:PROGram:UNLoad

:SYSTem:DCInterface[:PROGram]:SELected?

Syntax :SYSTem:DCInterface[:PROGram]:SELected?

Description This query returns the DUT Control Interface script that is currently in use. If the DUT Control Interface is not using a script, an empty string is returned.

Example :SYSTem:DCInterface:PROGram:SELected?

:SYSTem:DCInterface:EXEcute[:INIT]

Syntax :SYSTem:DCInterface:EXEcute[:INIT] 'Location', ['ArgumentsForInit']

Input 'Location': "*.Lane"

Parameters

Description This command matches the DUT Control Interface hook function DUT_Init(location, argument).

Sending this command results in executing the function DUT_Init with the given parameters if it is implemented by the script that is currently in use (see syst:dcin:load).

The Location argument shall be one of the locations defined by the scripts DUT_getLocations, but as full system location identifier.
For example, if DUT_getLocations defines the locations "Lane1" and "Lane2", then valid Location arguments are "DCI.Lane1" and "DCI.Lane2"
The argument string is not checked anyway, but simply passed to the script implementation of DUT_Init.

Example

:SYSTEM:DCInterface:execute:init "DCI.Lane1"

:SYSTEM:DCInterface:DEVICE:CONNect[?]

Syntax

:SYSTEM:DCInterface:DEVICE:CONNect 'identifier', <ON|OFF|1|0>

:SYSTEM:DCInterface:DEVICE:CONNeck 'identifier'?

Input Parameters

'identifier': "DCI.Control"

Description

This command relates to the DUT Control Interface hook functions DUT_connect and DUT_disconnect.

When these functions are implemented by the currently used script (see syst:dcin:load), then this command is controlling the Connection parameter under the function Device for the Control location of the device under test.

The location argument must be the Control location as full system location identifier as displayed in the parameter editor (e.g. "DCI.Control").

This query returns the current setting.

Example

:SYSTEM:DCInterface:DEVICE:CONNect 'DCI.Control', 1
:SYSTem:DCINterface:DEVice:MODE[?]

Syntax

:SYSTem:DCINterface:DEVice:MODE 'identifier', 'mode'
:SYSTem:DCINterface:DEVice:MODE? 'identifier'

Input Parameters

'identifier': 'DCI.Control'
'mode': Individual modes or configurations.

Description

This command relates to the DUT Control Interface hook functions DUT_getDeviceModes and DUT_setDeviceMode.

When these functions are implemented by the currently used script (see syst:dcin:load), then this command is controlling the Mode parameter under the function Device for the Control location of the device under test.

The location argument must be the Control location as full system location identifier as displayed in the parameter editor (e.g. "DCI.Control").

The mode argument must be one of the strings returned by DUT_getDeviceModes.

This can be used to control configurations that are valid for the whole device under test.

When sending this command in one SCPI transaction with :syst:dcin:lane:mode, then the device mode hook (DUT_setDeviceMode) will be executed before the lane specific mode (DUT_setLaneMode).

This query returns the current setting.

Example

:SYSTem:DCINterface:DEVice:MODE 'DCI.Control','Configuration1'
Syntax

`:SYSTem:DCINterface:LANE:MODE[?]`

- **Input Parameters**
  - `'Location'`: "*.Lane"
  - `'mode'`: Individual modes or configurations.

**Description**

This command relates to the DUT Control Interface hook functions `DUT_getLaneModes` and `DUT_setLaneMode`.

When these functions are implemented by the currently used script (see `syst:dcin:load`), then this command is controlling the Mode parameter under the function Lane for the given lane of the device under test.

The location argument must be the Control location as full system location identifier as displayed in the parameter editor (e.g. "DCI.Lane1").

The mode argument must be one of the strings returned by `DUT_getLaneModes`.

This can be used to control configurations that are valid for an individual lane or channel of the device under test.

When sending this command in one SCPI transaction with `:syst:dcin:dev:mode`, then the device mode hook (`DUT_setDeviceMode`) will be executed before the lane specific mode (`DUT_setLaneMode`).

This query returns the current setting.

**Example**

`:SYSTem:DCINterface:LANE:MODE 'DCI.Lane1', 'MODE1'`
:SYSTem:RESet[:ALL]

Syntax :SYSTem:RESet[:ALL] [HARD | SOFT]

Input [HARD | SOFT]

Parameters

Description This command resets the instrument.

If SOFT is passed the instrument will load its default values and defined 'Groups' in the GUI will be untouched.

If HARD is passed the instrument will load the factory setting and all 'Group' definitions will be cleared.

Independent of passing HARD or SOFT all plug-ins (measurements) are closed.

Example :SYSTem:RESet:ALL SOFT

:SYSTem:RESet:INStrument

Syntax :SYSTem:RESet:INStrument [HARD | SOFT]

Input [HARD | SOFT]

Parameters

Description This command resets the instrument.

If SOFT is passed the instrument will load its default values and defined 'Groups' in the GUI will be untouched.

If HARD is passed the instrument will load the factory setting and all 'Group' definitions will be cleared.

Example :SYSTem:RESet:INStrument SOFT
DATA Subsystem

The DATA subsystem is used to select patterns, define symbol parameters, define sequence blocks and loops and synchronize the pattern.

This subsystem has the following SCPI structure:

```
:DATA
  :PATTern
  :SEQUence
  :SYNC
  :LINecoding
  :FEC
```

The subsystem has the following subnodes:

Table 97

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:PATTern</td>
<td>:DATA:PATTern Subnode on page 512</td>
</tr>
<tr>
<td>:SEQUence</td>
<td>:DATA:SEQUence Subnode on page 520</td>
</tr>
<tr>
<td>:SYNC</td>
<td>:DATA:SYNC Subnode on page 529</td>
</tr>
<tr>
<td>:LINecoding</td>
<td>:DATA:LINecoding Subnode on page 531</td>
</tr>
<tr>
<td>:FEC</td>
<td>:DATA:FEC Subnode on page 536</td>
</tr>
</tbody>
</table>
:DATA:PATTer

For additional information about patterns, refer to Working with Patterns on page 38.

This subnode has the following SCPI structure:

```
:DATA
  :PATTern
    -- :USE[?]
    -- :IDATa[?]
    -- :BLENgh?
    -- :CATalog?
    -- :CINFormation?
    -- :DESCription[?]
    -- :DELeTe
    -- :PACKed[?]
    -- :IMPorT
    -- :EXPorT
```

This subnode has the following commands:

Table 98

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:USE[?</td>
<td>:DATA:PATTer:USE[?] on page 513</td>
</tr>
<tr>
<td>:IDATa[?</td>
<td>:DATA:PATTer:IDATa[?] on page 514</td>
</tr>
<tr>
<td>:DESCription[?]</td>
<td>:DATA:PATTer:DESCription?[?] on page 516</td>
</tr>
<tr>
<td>:DELeTe</td>
<td>:DATA:PATTer:DELeTe on page 516</td>
</tr>
<tr>
<td>:PACKed[?]</td>
<td>:DATA:PATTer:PACKed[?] on page 517</td>
</tr>
<tr>
<td>:IMPorT</td>
<td>:DATA:PATTer:IMPorT on page 517</td>
</tr>
<tr>
<td>:EXPorT</td>
<td>:DATA:PATTer:EXPorT on page 519</td>
</tr>
</tbody>
</table>
:DATA:PATTern:USE[?]

Syntax

:DATA:PATTern:USE <pattern-name>,<num-symbols>[,<symbol-coding>
[,<mask-used>[,<squelch-used>,<error-used>]]]

:DATA:PATTern:USE? <pattern-name>

Input

<pattern-name>: Name of the pattern. Name consists of root node
("factory:“, "shared:“, or "current:“) plus folder names separated by ‘/’ plus
pattern name.

<num-symbols>: Length of the pattern in number of symbols.

<symbol-coding>: BIT|B8B10|B128B130|B128B132 for bit coding, 8b10b
coding, 128/130 coding or 128/132 coding. Default is BIT.

<mask-used>: 0|1|OFF|ON to specify if mask attribute is used. Default is
OFF. Not using mask is mostly useful for bit coding to save space.

<squelch-used>: 0|1|OFF|ON to specify if squelch attribute is used. Default
is OFF. Not using squelch is mostly useful for bit coding to save space.

<error-used>: 0|1|OFF|ON to specify if error attribute is used. Default is
OFF. Error Attribute is used for captured patterns. When <error-used> is
ON, <mask-used> and <squelch-used> must be OFF.

Description

This command creates a new pattern. If a pattern exists with the same
name, it is overwritten.

Example

:DATA:PATT:USE *current:demo",20
Example

:DATA:PATT:USE *current:demo8b10b",5,b8b10
Example

:DATA:PATT:USE *current:demo128b130b",5,b128b130
Example

:DATA:PATT:USE *current:demo128b132b",5,b128b132
Example


5,BIT,0,0,0
Example

:DATA:PATT:USE? *factory:SATA/LTDP_short_b8b10*

512,B8B10,1,1,0
:DATA:PATT:IDATa

Syntax
:DATA:PATT:IDATa
<pattern-name>,<bit-offset>,<num-bits>,<data-block>

:DATA:PATT:IDATa? <pattern-name>,<bit-offset>,<num-bits>

Input

<pattern-name>: Name of the pattern. Name consists of root node
("shared:" or "current:" plus folder names separated by "/") plus pattern
name. The "factory:" root node is not allowed here, as these patterns are
read only and cannot be modified.

<bit-offset>: Offset into the pattern bit sequence. Bit offset is used, to
allow modifications within symbols, for example just modifying a single
attribute bit.

<num-bits>: Number of bits to modify. Specified in bits to allow
modifications within symbols, e.g. just modifying a single attribute bit.

<data-block>: Bit sequence as definite length block. Number of bits
encoded in one byte is specified by the :DATA:PATT:PACK command.

Return Range
<pattern-name>, <bit-offset>, <num-bits>

Description
This command is used to modify a pattern.

Example
The following example enables the scrambler in the first symbol of an
8b10b coded pattern.

:DATA:PATT:IDAT 'current:demo8b10b',12,1,#111

:DATA:PATT:BLENgth?

Syntax
:DATA:PATT:BLENgth? <pattern-name>

Input

<pattern-name>: Name of the pattern. Name consists of root node
("factory:" or "shared:" or "current:" plus folder names separated by "/") plus pattern
name.

Return Range
<num-bits>

Description
This command returns the pattern length in bits of a specified pattern.

Example
:DATA:PATT:BLEN? "factory:SATA/LTDP_short_b8b10"
8192
:DATA:PRGM:CATalog?
Syntax :DATA:PRGM:CATalog? <pattern-path>
Input <pattern-path>: Location of patterns.
Parameters
Description This command returns a list of patterns found at the specified path.
Example :DATA:PRGM:CAT? "factory:XAUI"
"CJPAT_bit,CRPAT_b8b10,CRPAT_bit"

:DATA:PATTern:CINFormation?
Syntax :DATA:PATTern:CINFormation?
<symbol-coding>,<mask-used>,<squelch-used>
Input <symbol-coding>: BIT|B8B10|B128B130|B128B132 for bit coding, 8b10b coding.
Parameters <mask-used>: 0|1|OFF|ON to specify if mask attribute is used. Default is OFF. Not using mask is mostly useful for bit coding to save space.
<squelch-used>: 0|1|OFF|ON to specify if squelch attribute is used. Default is OFF. Not using squelch is mostly useful for bit coding to save space.
Return Range <symbol-coding>, <mask-used>, <squelch-used>
Description This command returns coding information.
Example :DATA:PRGM:CINF? b8b10,on,on
:DATA:PATTern:DESCription[?]

Syntax: :DATA:PATTern:DESCription <pattern-name>,<description>
:DATA:PATTern:DESCription? <pattern-name>

Input:

- <pattern-name>: Name of the pattern. Name consists of root node (*factory:*, *shared:*, or *current:*), plus folder names separated by ‘/’ plus pattern name; like *shared:SATA/LTDP_short_b8b10*.
- <description>: Enter a pattern description.

Return Range: <description>

Description: This command is used to add a description to the specified pattern. This SCPI doesn't allow adding description to factory patterns.

Example: Create a new pattern by the name "bit_pattern" and save it to "shared" memory location. Now, use the following example to set "sample bit pattern" as the description.

```
:DATA:PATT:DESC "shared:bit_pattern","sample bit pattern"
```

:DATA:PATTern:DELeete

Syntax: :DATA:PATTern:DELeete <pattern-name>

Input:

- <pattern-name>: Name of the pattern. Name consists of root node (*factory:*, *shared:*, or *current:*), plus folder names separated by ‘/’ plus pattern name.

Description: This command deletes the specified pattern. A factory pattern can’t be deleted.

Example: :DATA:PATT:DEL "shared:SATA/LTDP_short_b8b10"
:DATA:PATTern:PACKed[?]  
Syntax :DATA:PATTern:PACKed <packing>  
 :DATA:PATTern:PACKed?

Input Parameters  
<packing>: Specify number of bits (1|4|8) that are encoded in a byte when using the :DATA:PATT:IDAT command or query.

Return Range 1|4|8  
Description This command specifies how many bits are encoded in a byte when using the :DATA:PATT:IDAT command or query. The default is "8" for most efficient coding.

Example :DATA:PATTern:PACKed 1

:DATA:PATTern:IMPort  
Syntax :DATA:PATTern:IMPort <pattern-name>, <file-name>  

Input Parameters  
<pattern-name>: Specify the pattern name.
<file-name>: Specify the location of the pattern file to be imported.

Description This command imports an N4903B J-BERT pattern or M8020A/M8030A pattern file.

M8020A/M8030A File Format:
File content consists of a short header and the actual symbol sequence.
The file content looks like the following:
Version=<version>
Use=<num-symbols>[,[<symbol-coding>[,[<mask-used>[,[<squelch-used>[,[<error-used>]]]]]]]]
[<Description=<description>]
[Pack=<packing>]
Data=<data>

Optional lines or fields are enclosed in square brackets. Format and notation is intended to follow the concepts of the other :DATA:PATTern SCPI commands. The first line must include a line with the file format version. Accepted value for <version> is M8000 1.0.0 or M8000 1.0.1.
<error-used> field requires version 1.0.1.

Note that if <error-used> is 1, <mask-used> and <squelch-used> must both be 0. Also the coding must be BIT or B8B10. B128B130 and B128B132 are not supported.

The next line describes the number of symbols, symbol coding, mask usage and squelch usage as described in :DATA:PATTern:USE command.

The next line allows specifying a description.

The next line allows changing the packing of the actual <data> part similar to the :DATA:PATTern:PACK command. If not specified, 8 is assumed. For details on packing and its parameters, refer to :DATA:PATTern:PACKed[*] on page 517.

The <data> in a separate line contains the sequence of bytes as specified by the packing. For more details on how to edit data, refer to the chapter “Setting up Patterns” of the M8000 Series User Guide.

Example File:

Version=M8000 1.0.0
Use=4
Pack=1
N4903B J-BERT Patterns

N4903B J-BERT patterns are always imported as bit coded patterns with no mask and no squelch. The B part of an alternate pattern will be imported as <pattern-name>B.

The following is an example of an N4903B J-BERT pattern import:

Example: DATA:PATT:IMP "current:x", "C:\N4903B\Pattern\xxx.ptrn"

:DATA:PATTern:EXPort

Syntax: :DATA:PATT:EXPort <pattern-name>, <file-name>

Input Parameters:
- <pattern-name>: Specify the pattern name.
- <file-name>: Specify the location of the pattern file to be exported.

Description: This command exports a pattern to a file in the M8020A/M8030A import file format.

:DATA:SEQuence Subnode

With this subsystem sequencing can be changed programmatically. For additional information, refer to Creating Pattern Sequences on page 45.

This subnode has the following SCPI structure:

```
:DATA
  :SEQuence
    [:SET]?  
    [:SET]:VALue[?]
    :NEW
    :NAMes?
    :BIND[?]
    :DELete
    :DALL
    :BREak
    :RESTart
    ::IDENTifier
    ::IDENTifier?
    ::GRANularity?
    ::IDENTifier?
```

This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:SET]?</td>
<td>:DATA:SEQuence[:SET]? on page 521</td>
</tr>
<tr>
<td>[:SET]:VALue[?]</td>
<td>:DATA:SEQuence[:SET]:VALue[?] on page 523</td>
</tr>
<tr>
<td>:NEW</td>
<td>:DATA:SEQuence:NEW on page 524</td>
</tr>
<tr>
<td>:BIND[?]</td>
<td>:DATA:SEQuence:BIND[?] on page 525</td>
</tr>
<tr>
<td>:DELete</td>
<td>:DATA:SEQuence:DELete on page 525</td>
</tr>
<tr>
<td>:DALL</td>
<td>:DATA:SEQuence:DALL on page 525</td>
</tr>
<tr>
<td>BREak</td>
<td>:DATA:SEQuence:BREak on page 526</td>
</tr>
</tbody>
</table>
### :DATA:SEQUence[:SET][?] Syntax

:DATA:SEQUence[:SET] <sequence-name | location | group-name>, <pattern type>, <pattern-value>

:DATA:SEQUence[:SET]? <sequence-name | location>

:DATA:SEQUence[:SET] is a command to set a sequence of patterns. It can also be used to query the current settings.

**Input Parameters**

- **<sequence-name | location | group-name>** is a string name for which possible values are:
  - <sequence-name> Name of the existing sequence, either default sequence name or user created sequence name.
  - <location> Individual Analyzer or Generator location e.g. "M1.DataIn1"
  - <group-name> A user created group name containing Analyzer(s) or Generator(s) locations. In this case the group name must only contain numbers and letters.

- **<pattern type>**
  - PRBS: Sets a PRBS of specified length.
  - PULSE: Sets a pulse of specified width.
  - CLOCK: Sets a clock of specified divider.
  - STATIC: Sets a static signal of specified value.
  - MEMORY: Sets a memory pattern saved in the setting.
<pattern-value>: String value parameter according to the pattern type.

- For PRBS possible values are: \texttt{"2^7-1"}, \texttt{"2^9-1"}, \texttt{"2^10-1"}, \texttt{"2^11-1"}, \texttt{"2^13-1"}, \texttt{"2^15-1"}, \texttt{"2^23-1"}, \texttt{"2^23-1p"} or \texttt{"2^23-1 PCle"}, \texttt{"2^31-1"}, \texttt{"2^33-1"}, \texttt{"2^35-1"}, \texttt{"2^39-1"}, \texttt{"2^41-1"}, \texttt{"2^45-1"}, \texttt{"2^47-1"}, \texttt{"2^49-1"}, \texttt{"2^51-1"}
- For PULSE it's a specified width value e.g. \texttt{"32"}
- For CLOCK it's a specified divider value e.g. \texttt{"2"}
- For STATIC it's a signal value e.g. \texttt{"0"}, \texttt{"1"}, \texttt{"low"}, \texttt{"high"}
- For MEMORY it's a name of the pattern. Name consists of root node ("factory:", "shared:" or "current:"), plus folder names separated by "/" plus pattern name. e.g. \texttt{"factory:CEI/CEIstress_bit"}

Description
This command allows you to:
- set sequence for a specified location with the specified pattern.
- set or change a pattern for single location, sequence or multiple locations (executing user group).
- create/modify a sequence for the given location with a single block pattern in an infinite loop.

For more verbose setting use, :DATA:SEQUence[:SET]:VALue command providing XML value.

This query returns the following:
- current pattern set for the given existing sequence or Analyzer/Generator location.
- pattern and value if the current sequence for the given location contains single data block pattern.

If there are more than one block or block targets are used in the sequence then use, :Data:SEQ:SET:VALue? query to retrieve xml sequence setting value.

Example
The following are few command examples:


The following are few query examples:

:DATA:SEQ:SET? "Generator"
  > "Generator",PRBS,"2^9-1"
:DATA:SEQ:SET? "M1.DataIn1"
  > "M1.DataIn1",PULSE,"32"
  > "M1.DataOut1",CLOCK,"2^9-1"
:DATA:SEQ:SET? "Analyzer"
  > "Analyzer",STATIC,"1"
:DATA:SEQ:SET? "M1.DataIn1"
  > "M1.DataIn1",MEMORY,"factory:CEI/CEIstress_bit"

**:DATA:SEQuence[:SET]:VALue[?]**

**Syntax**: :DATA:SEQuence[:SET]:VALue <sequence-name>,<sequence-string>

:DATA:SEQuence[:SET]:VALue? <sequence-name>

**Input Parameters**

- <sequence-name>: Specify sequence name.
- <sequence-string>: Enter definite length block and pattern sequence XML string.

**Return Range**

<sequence-string>

**Description**

This command is used to enter a sequence string consisting of the pattern sequence parameters for the specified sequence name and download it to the hardware. A definite length block must be entered indicating the length of the string to download.

**NOTE**

It is recommended to edit the sequence string in the Sequence Editor in either "UI" or "<Xml>" mode and then copy/paste the xml sequence string.
In the following example, #3340 defines the length of the string:

#3 indicates that a definite length arbitrary data block follows and its length is specified in 3 digits.

340 indicates that 340 bytes follow. In this example, the bytes are all readable ascii characters.

Example:
```
:DATA:SEQUence:VALue 'Generator',#3342
```

Example:
```
:DATA:SEQ:NEW 'MySequence'
```

Example:
```
:DATA:SEQ:NAM?
"Generator","Analyzer","MySequence"
```
:DATA:SEQuence:BIND

Syntax: :DATA:SEQuence:BIND <sequence-name>,<identifier-list>

- :DATA:SEQuence:BIND? <sequence-name>

Input Parameters:
- <sequence-name>: Specify sequence name.
- <identifier-list>: Specify location or group name identifier.

Return Range: <identifier-list>

Description: This command binds the identifiers to a specified sequence. The identifier is either a location or a group name identifier. If locations are already used in another sequence, they get re-assigned to this sequence.

This SCPI is applicable for M8041A, M8051A, M8195A, M8196A and M8046A. However, for M8195A and M8196A, it is only valid for Data Out locations.

Example: :DATA:SEQ:BIND 'MySequence','M1.DataIn1'

:DATA:SEQuence:DELete

Syntax: :DATA:SEQuence:DELete <sequence-name>

Input Parameters:
- <sequence-name>: Specify sequence name.

Description: This command deletes the specified sequence.

Example: :DATA:SEQ:DEL 'MySequence'

:DATA:SEQuence:DALL

Syntax: :DATA:SEQuence:DALL

Description: This command deletes all sequences. This is a useful function if you do not want to use the reset sequence ("Generator", "Analyzer").

Example: :DATA:SEQ:DALL
:DATA:SEQuence:BREak

Syntax :DATA:SEQuence:BREak 'identifier'

Input <identifier>: 'M*.System'

Parameters

Description This command sends a break signal addressed by the identifier. This will leave a loop and execute the next block or execute an arbitrary jump to another block.

Example :DATA:SEQ:BRE 'M1.System'

:DATA:SEQuence:RESTart

Syntax :DATA:SEQuence:RESTart [<sequence-name>,<sequence-block-id>]

Input <sequence-name>: Specify sequence name.
<sequence-block-id>: Specify sequence block ID.

Parameters

Description This command restarts all sequences if no argument is specified. This command restarts a specified sequence if <sequence-name> is specified. This command jumps to a specific sequence block if <sequence-name> and <sequence-block-id> is specified.

• Limitations on sequence block ID selection:
  In case the sequencer block ID selected is contained inside an outer loop and it is not the starting segment the outer loop will not be executed.

   NOTE Analyzer(s) are not able to do a break. So using this command, you can branch to any other block, also on analyzers.

Example :DATA:SEQ:REST

   :DATA:SEQ:REST 'Analyzer'
   :DATA:SEQ:REST 'Generator', 3
:DATA:SEQUence:RESTart:IDENTifier
Syntax :DATA:SEQUence:RESTart:IDENTifier <identifier>
Input <identifier>: 'M*.DataIn1', 'M*.DataIn2', 'M*.DataIn', 'M*.DataOut',
Parameters 'M*.DataOut1' and 'M*.DataOut2'
Description This command restarts the specified sequence (jumps to the first block).
For example, this command is useful for ED sequences. Usually there is a
synch block first and then a loop to compare the synchronized pattern.
Performing a restart there causes a synchronization of the sequence.
Example :DATA:SEQ:REST:iden 'M1.DataIn1'

:DATA:SEQUence:BLOCk?
Syntax :DATA:SEQUence:BLOCk? <sequence-name>
Input <sequence-name>: Specify sequence name.
Parameters
Description This command returns the block of sequences currently executed.
Example :DATA:SEQ:BLOCk? 'Generator'

:DATA:SEQUence:BLOCk:IDENTifier?
Syntax :DATA:SEQUence:BLOCk:IDENTifier? <identifier>
Input <identifier>: 'M*.DataIn1', 'M*.DataIn2', 'M*.DataIn', 'M*.DataOut',
Parameters 'M*.DataOut1' and 'M*.DataOut2'
Description This command returns the block of sequences currently executed for a
specific location.
Example :DATA:SEQ:BLOC:iden 'M1.DataIn1'
:DATA:SEQuence:GRANularity?

Syntax  
:DATA:SEQuence:GRANularity? <sequence-name>

Input  
<sequence-name>: Specify sequence name, i.e., 'Sequence1'

Parameters  

Description  
This query returns the granularity in bits of sequence. It changes with the symbol width. The following table shows the granularity for various symbol widths for different modules:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>64</td>
<td>128</td>
<td>512</td>
</tr>
<tr>
<td>10</td>
<td>80</td>
<td>160</td>
<td>Not supported</td>
</tr>
<tr>
<td>130</td>
<td>130</td>
<td>260</td>
<td>Not supported</td>
</tr>
<tr>
<td>132</td>
<td>132</td>
<td>264</td>
<td>Not supported</td>
</tr>
</tbody>
</table>

Example  
:DATA:SEQ:GRAN? 'Sequence1'

:DATA:SEQuence:GRANularity:IDENtifier?

Syntax  
:DATA:SEQuence:GRANularity:IDENtifier? <identifier>

Input  
<identifier>: 'M*.DataIn1', 'M*.DataIn2', 'M*.DataIn', 'M*.DataOut', 'M*.DataOut1' and 'M*.DataOut2'

Parameters  

Description  
This query returns the current granularity in bits of location or group. It changes with the symbol width. Refer to the table shown in the command :DATA:SEQuence:GRANularity:IDENtifier? on page 528.

Example  
:DATA:SEQ:GRAN:IDEN? 'M1.DataIn1'
:DATA:SYNC Subnode

This subnode has the following SCPI structure:

```
:DATA
  :SYNC
    [:ONCE]
    [:TYPE?]?
    [:THReshold?]?
```

This subnode has the following commands:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>[:ONCE]</td>
<td>:DATA:SYNC[:ONCE] on page 529</td>
</tr>
<tr>
<td>[:TYPE?]</td>
<td>:DATA:SYNC:TYPE[?] on page 530</td>
</tr>
<tr>
<td>[:THReshold?]</td>
<td>:DATA:SYNC:THReshold[?] on page 530</td>
</tr>
</tbody>
</table>

**:DATA:SYNC[:ONCE]**

**Syntax**: :DATA:SYNC[:ONCE] <identifier>

**Input Parameters**

<identifier>: 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'

**Description**: This command initiates a pattern resynchronization.

This SCPI is applicable for M8041A, M8051A, M8062A and M8046A.

**Example**: :DATA:SYNC 'M1.DataIn1'
:DATA:SYNC:TYPE [?]

Syntax
:DATA:SYNC:TYPE 'identifier', AUTO|MANual
:DATA:SYNC:TYPE? 'identifier'

Input
<identifier>: 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn'

Parameters
AUTO|MANual: Enable/disable automatic resynchronization.

Description
This command enables/disables automatic resynchronization.

Example
:DATA:SYNC:TYPE 'M1.DataIn1', AUTO

:DATA:SYNC:THReshold [?]

Syntax
:DATA:SYNC:THReshold 'identifier', <NRf>
:DATA:SYNC:THReshold? 'identifier'

Input
<identifier>: 'M*.DataIn1', 'M*.DataIn2' or 'M*DataIn', 'INF*.DataIn1'

Parameters
<NRF>: Set the threshold level.

Return Range
1E-8 to 1E-1

Description
This command sets the threshold level of error ratio at which synchronization is successful. Exceeding this BER threshold triggers a pattern synchronization when in automatic sync mode.

Example
:DATA:SYNC:THR 'M1.DataIn1', 1E-5
:DATA:LINecoding Subnode

The :DATA:LINecoding subnode controls which line coding is being used when sending or receiving the digital data.

This subnode has the following SCPI structure:

```
:DATA
  :LINecoding
    [:VALue][?]  
    :PAM4
      :MAPPING[?]
      :MAPPING:CUSTom[?]
      [:SYMBOL]:LEVEL(1|2)[?]
      :PRECoding[?]
```

This subnode has the following commands:

Table 101

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
<tbody>
<tr>
<td>:DATA:LINecoding[:VALue]?</td>
<td>:DATA:LINecoding[:VALue]? on page 532</td>
</tr>
<tr>
<td>:DATA:LINecoding:PAM4[:SYMBOL]:LEVEL(1</td>
<td>2)[?]</td>
</tr>
</tbody>
</table>
:DATA:LINecoding[:VALue]?

**Syntax**
:DATA:LINecoding[:VALue] 'identifier', NRZ | PAM4
:DATA:LINecoding[:VALue]? 'identifier'

**Input Parameters**
- `<identifier>`: Any valid data output or data input identifier. M*.DataOut1, M*.DataOut2, M*.DataOut3, M*.DataOut4, M*DataIn and INF*.DataIn1

**Description**
The command selects the line coding to be used. The availability of individual selections depends on the hardware capabilities.

This query returns the present state.

This SCPI is applicable for M8195A, M8196A, M8045A, M8046A, DSO634A, DSA634A, DSOX96204A and DSAX96204A.

**Example**
:DATA:LIN 'M1.DataOut1', PAM4
:DATA:LIN 'M2.DataIn1', PAM4

:DATA:LINecoding:PAM4:MAPPing?

**Syntax**
:DATA:LINecoding:PAM4:MAPPing 'identifier', NONE | GRAY | CUSTOM
:DATA:LINecoding:PAM4:MAPPing? 'identifier'

**Input Parameters**
- `<identifier>`: Any valid data output or data input identifier. M*.DataOut1, M*.DataOut2, M*.DataOut3, M*.DataOut4, M*DataIn and INF*.DataIn1

**Description**
The command selects how the pattern bits are mapped to symbol numbers.

- Select NONE if the mapping shall be 'transparent'. This is useful if the pattern does already contain the required coding scheme. This option is available for M8195A, M8045A and M8046A.
- Select GRAY if consecutive pairs of bits shall be Gray coded in order to map to the PAM4 symbols. This option is available for M8195A M8045A and M8046A.
- Select CUSTOM to use a user defined mapping definition. This option is only available for M8045A and M8046A.

This query returns the present state.

This SCPI is applicable for M8195A, M8196A, M8045A, M8046A, DSO634A, DSA634A, DSOX96204A and DSAX96204A.

**Example**
:DATA:LIN:PAM4:MAPP 'M1.DataOut1', NONE
:DATA:LIN:PAM4:MAPP 'M2.DataIn1', CUSTOM
:DATA:LINEcoding:PAM4:MAPPing:CUSTom[?]

Syntax: :DATA:LINEcoding:PAM4:MAPPing:CUSTom 'identifier', <"mapping definition">

:DATA:LINEcoding:PAM4:MAPPing:CUSTom? 'identifier'

Input Parameters:
- <identifier>: Any valid data output or data input identifier.

Description:
The command defines a custom bit sequence to symbol number mapping. The mapping defined with this command will be used when CUSTom is selected as the active mapping.

The mapping is defined as a string that contains a comma separated list of bit pairs. The position in the list corresponds to the symbol number.

"ww,xx,yy,zz" is mapping to Symbol 0 to Symbol 3 according to the following table.

<table>
<thead>
<tr>
<th>Symbol Number</th>
<th>Bit Pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbol 0</td>
<td>ww</td>
</tr>
<tr>
<td>Symbol 1</td>
<td>xx</td>
</tr>
<tr>
<td>Symbol 2</td>
<td>yy</td>
</tr>
<tr>
<td>Symbol 3</td>
<td>zz</td>
</tr>
</tbody>
</table>

This query returns the present state.

This SCPI is applicable for M8045A, M8046A, DSO634A, DSA634A, DSOX96204A and DSAX96204A.

Example: :DATA:LIN:PAM4:MAPP 'M1.DataIn1', '11,00,01,10'

This command defines that Symbol 0 is being used for '11', Symbol 1 is being used for '00', Symbol 2 is being used for '01' and Symbol 3 is being used for '10'.

Example: :DATA:LIN:PAM4:MAPP 'M1.DataIn1', '11,00,01,10'
:DATA:LINecoding:PAM4[:SYMBol]:LEVel(1|2) [?]

Syntax
:DATA:LINecoding:PAM4[:SYMBol]:LEVel(1|2) 'identifier', <NRf>
:DATA:LINecoding:PAM4:MAPPing? 'identifier'

Input
<identifier>: Any valid data output identifier.

Parameters
M*.DataOut1, M*.DataOut2, M*.DataOut3 and M*.DataOut4

Range
Symbol 1 Level: 1% to 65%
Symbol 2 Level: 34% to 99%

Description
The command controls the PAM4 level mappings. It can be used to adjust the actual output level that is being generated by a data output for a specific PAM4 symbol.

The levels of PAM4 symbol 0 and 3 cannot be adjusted and are always kept at the value of 0% and 100% of the actual output voltage swing.

It is best practice to configure symbol level 1 and 2 in a combined remote command transaction (see example below) to minimize hardware reconfiguration duration. This is especially important when the location is addressing a data output that is located on an arbitrary waveform generator module (e.g. M8195A).

This query returns the present state.

This SCPI is applicable for M8195A, M8196A and M8045A.

Example
:DATA:LIN:PAM4:LEV1 'M1.DataOut1', 35pct;LEVel2 'M1.DataOut1', 70pct

:DATA:LINecoding:PAM4:PRECoding[?]

Syntax
:DATA:LINecoding:PAM4:PRECoding 'identifier', <ON|OFF|1|0>
:DATA:LINecoding:PAM4:PRECoding? 'identifier'

Input
<identifier>: Any valid data output or data input identifier.

Parameters
M*.DataOut1, M*.DataOut2

Description
The command enables/disables the pre-coding of the PAM4 symbols.

Pre-coding is an optional extra step between the symbol mapping and PAM4 level mapping. It is normally used together with a Gray code for the symbol mapping.
The pre-coder does not enforce using a Gray code for symbol mapping. This allows device specific custom symbol mapping (in case of LSB first devices), or using patterns that already contain the required symbol mapping (e.g. already gray encoded data).

This query returns the present state.

This SCPI is applicable for M8045A.

Example: 
`.DATA:LIN:PAM4:PREC 'M1.DataOut1', ON`
The DATA:FEC subnode describes programming specification for the FEC error insertion feature.

FEC support in M8070A is enabled by installing a module specific FEC license. FEC licenses are available for M8045A.

The following FEC types are available
- For IEEE 802.3cd 50GBASE-R
  - 1 lane at 26.5125GBd using PAM4
  - FEC RS(544,514)

FEC is enabled by selecting one of the FEC factory patterns.
- IEEE_802_3cd_RS_544_514_Scrambled_Idle
- IEEE_802_3cd_RS_544_514_Remote_Fault

Due to the differences between FEC frame length and sequencer word width, there will be phases of invalid FEC data when transitioning from a FEC encoded sequence lock into a non-FEC sequence block and vice versa.

The DATA:FEC:EINSertion Subsystem

When a FEC pattern is used in the sequence, then the pattern generator offers a FEC specific error insertion.

The FEC error insertion is independent from the error insertion controlled with the :OUTPut:EINSertion subnode (see :OUTPut:EINSertion Subnode on page 228). Both error insertions can be used at the same time.

This subnode has the following SCPI structure:

```
:DATA
  :FEC
    :EINSertion
      :STATE[?]
      :ERRors[?]
      :BIP:ONCE[?]
```
This subnode has the following commands:

Table 102

<table>
<thead>
<tr>
<th>Name</th>
<th>Description under</th>
</tr>
</thead>
</table>

**:DATA:FEC:EINSertion:STATe[?]**

**Syntax**

:DATA:FEC:EINSertion::STATE 'identifier', <ON|OFF|1|0>

:DATA:FEC:EINSertion::STATE[?] 'identifier'

**Input Parameters**

<identifier>: Any valid data output identifier. 'M*.DataOut1','M*.DataOut2'

ON | OFF | 1 | 0

**Return Range**

1 | 0

**Description**

This command enables or disables the continuous FEC error insertion. The inserted errors are after the FEC encoding. This query returns the present state. This SCPI is applicable for M8045A.

**Example**

:DATA:FEC:EINS:STAT 'M1.DataOut1',ON


1

**:DATA:FEC:EINSertion:ERRors[?]**

**Syntax**

:DATA:FEC:EINSertion:ERRors 'identifier', <NRf>

:DATA:FEC:EINSertion:ERRors ? 'identifier'

**Input Parameters**

<identifier>: Any valid data output identifier. 'M*.DataOut1','M*.DataOut2'

ON | OFF | 1 | 0

**Description**

This command enables or disables the continuous FEC error insertion. The inserted errors are after the FEC encoding. This query returns the present state. This SCPI is applicable for M8045A.
Return Range: 0 to 16

Description: This command specifies the number of symbols to be corrupted within a single FEC frame. The inserted errors are at random symbol positions. Each corrupted symbol will contain exactly one bit error at a random bit position.

This query returns the present state.

This SCPI is applicable for M8045A.

Example:
:DATA:FEC:EINS:ERR 'M1.DataOut1',8
8

:DATA:FEC:EINSertion:BIP:ONCE[?]?


Input Parameters:
<identifier>: Any valid data output identifier. 'M*.DataOut1', 'M*.DataOut2'

Return Range: 1 | 0

Description: This command inserts a single Bit Interleaved Parity (BIP) error. This is a pre-FEC error and can be inserted at any time, even if the FEC error insertion is disabled.

This query returns the present state.

This SCPI is applicable for M8045A.

Example:
Index

Symbols

*CLS, 103
*ESE[?], 103
*ESR?, 103
*IDN?, 104
*OPC, 89, 104
*OPC?, 90, 104
*OPT?, 105
*RCL, 105
*RST, 105
*SAV, 105
*SRE[?], 106
*STB?, 106
*TST?, 106
*WAI, 89, 107

A
 analyzer sequence, 46

B
 block controls, global, 54
 block length, 50
 blocks, 49
 branch, conditional, 52

C
 Clock Commands
 :CLOCK:FREQuency[?], 279, 280, 284, 285, 286, 287, 288, 289
 :CLOCK:SOURce[?], 276, 278, 283
 coding configuration
 MIPI Phy, 57
 PCIe 1/2, 56
 SAS, 57
 SATA, 56

USB 3.0, 55
 condition register, 31
 Configuration
 32G BERT, 18, 19
 Base, 17, 21, 22
 M8020A 32 Gb/s BERT, 18, 19
 M8030A multi-channel BERT, 19, 21
 M8041A with M8061A, 18

D
 Data Commands
 :DATA:PATTern:BLENgth?, 514
 :DATA:PATTern:CATalog?, 515
 :DATA:PATTern:CIFormation?, 515
 :DATA:PATTern:DELete, 516
 :DATA:PATTern:DESCription[?], 516
 :DATA:PATTern:EXPort, 519
 :DATA:PATTern:IData[?], 514
 :DATA:PATTern:IMPort, 517
 :DATA:PATTern:PACKed[?], 517
 :DATA:PATTern:USE[?], 513
 :DATA:SEQUence:BIND[?], 525
 :DATA:SEQUence:BLOCK:IDENTifier?, 527, 528
 :DATA:SEQUence:BLOCK?, 527
 :DATA:SEQUence:BREak, 526
 :DATA:SEQUence:DALL, 525
 :DATA:SEQUence:DELeTe, 525
 :DATA:SEQUence:NAMES?, 524
 :DATA:SEQUence:NEW, 522, 524
 :DATA:SEQUence:RESTail, 526
 :DATA:SEQUence:RESTart:IDENTifier, 527
 :DATA:SEQUence:VALUE[?], 523
 :DATA:SYNC:THReshold[?], 521, 530, 532, 534, 537, 538
 :DATA:SYNC:TYPE[?], 530

E
 event enable register, 32
 event register, 32

F
 Fetch Commands
 :FETCH:BCounter?, 290
 :FETCH:IBERate?, 293
 FlexDCA, 281
 FlexDCA N1000-Series System Software, 281

G
 generator sequence, 46
 granularity, pattern generation, 48
 group name identifier, 24

I
 Identifiers
 Group Name, 23
 Location, 23
 Measurement Name, 23
 IEEE, 88
 IEEE Mandatory Commands
 *CLS, 103
 *ESE[?], 103
 *ESR?, 103
 *IDN?, 104
Index

**OPC**, 104  
**OPC?**, 104  
**OPT?**, 105  
**RCL**, 105  
**RST**, 105  
**SAV?**, 105  
**SRE?**, 106  
**STB?**, 106  
**TST?**, 106  
**WAI**, 107

IEEE mandatory commands, 102, 109, 110

Input Commands:  
:INPut:ALIGnment:EYE:ABORt, 2 64  
:INPut:ALIGnment:EYE:RESult:THReshold?, 266  
:INPut:ALIGnment:EYE:RESult:WIDDTh?, 265  
:INPut:ALIGnment:EYE:TCEN, 264  
:INPut:ALIGnment:EYE:THReshold?, 265  
:INPut:ALIGnment:EYE:[AUTO], 2 60, 261, 263, 273, 274  
:INPut:CDR:AUTO(?), 255  
:INPut:CDR:CTRL(?), 254  
:INPut:CDR:FIrst:LBANdwidth(?), 257  
:INPut:CDR:FIrst:TDENsity(?), 257  
:INPut:CDR:HIGHTD(?), 256  
:INPut:CDR:LORDer(?), 254  
:INPut:CDR:OPTimize, 256  
:INPut:CDR:RELOck, 256  
:INPut:CDR:SECond:LBANdwidth[?], 258  
:INPut:CDR:SECond:PEAKing[?], 259

:INPut:CDR:SECond:TDENsity(?), 258  
:INPut:CDR:[STATE]?, 255  
:INPut:CMODe[?], 248  
:INPut:CMVoltage:DETect, 246  
:INPut:CMVoltage[?], 246  
:INPut:COUPling[?], 249  
:INPut:DELay[?], 245  
:INPut:DRAta?, 244  
:INPut:EQUalization:[PREset]?, 271, 272  
:INPut:POLarity[?], 245  
:INPut:SENSitivity[?], 249, 250  
:INPut:TCONfig[?], 247  
:INPut:THReshold[?], 247  
:INPut:TVOLtage[?], 247

Link Training  
Configuration, 58  
Detect State, 53  
Error, 53  
Target State, 53  
location identifier, 23  
loop nesting, 49  
loops, 49

M  
M8020A  
Four channel configuration, 17  
Modules, 21  
M8030A, 19  
10-channel system configuration, 19, 21  
M8030A-BU1, 19  
M8030A-BU2, 19  
M8040A, 21  
M8041A module, 21  
M8045A Pattern Generator Module, 21  
M8045A, M8046A and M8057A configuration, 22  
M8046A Analyzer Module, 22  
M8057A Remote Head, 22  
M8061A 32 Gb/s multiplexer with de-emphasis, 18  
M8192A multi-channel synchronization module, 20  
M9505A AXle chassis, 21  
M9514A, 20  
M9514A AXle 14-slot chassis, 19  
M9537A AXle Embedded Controller, 21  
mask, 38  
measurement name identifier, 24

Memory Commands:  
:MEMory:WORKspace:IMPORT[:USER], 299  
:MEMory:WORKspace:IMPORT[:USER]:CATalog?, 300  
:MEMory:WORKspace:IMPORT[:USER]:DELete, 300  
:MEMory:WORKspace:SETTing:FACTory:CATalog?, 298  
:MEMory:WORKspace:SETTing:FACTory:RECall, 298  
:MEMory:WORKspace:SETTing[:USER]:DELete, 298  
:MEMory:WORKspace:SETTing[:USER]:RECall, 297  
:MEMory:WORKspace:SETTing[:USER]:STORe, 297

N  
N71000 Subsystem, 281  
nesting, loop, 49

O  
operation status register, 36  
Output Commands:  
:OUTPut:COUPling[?], 216  
:OUTPut:DATA:FI2JJitter[?], 218, 219  
:OUTPut:DATA:XEVer[?], 218  
:OUTPut:DEEmphasis:CONFigure:SIGN(*)[?], 225, 226, 227
Index

:PLUGinERATio:RUN:MESSaget?, 326
:PLUGinERATio:RUN:PROGrEss?, 326
:PLUGinERATio:RUN:[STATUs]?, 324, 327, 345, 364, 410, 412
:PLUGinERATio:STARt, 322
:PLUGinERATio:STOP, 322

Plugin Eye Diagram Commands
:PLUGin:EDIagram:ACQuisition:ACLocation[?], 430, 458
:PLUGin:EDIagram:ACQuisition:PNERSistence[?], 431, 460
:PLUGin:EDIagram:ACQuisition:TIME[?], 431
:PLUGin:EDIagram:BREak, 425
:PLUGin:EDIagram:CATalog?, 42
:PLUGin:EDIagram:CONTinue, 42
:PLUGin:EDIagram:DELete, 426, 451
:PLUGin:EDIagram:DELete, 426, 451
:PLUGin:EDIagram:EVALuation:BERThresh:THReshold[?], 434
:PLUGin:EDIagram:EVALuation:TIME[?], 434
:PLUGin:EDIagram:FETCh[:RESULT]:AMPLitude?, 436
:PLUGin:EDIagram:FETCh[:RESULT]:CVOLtage?, 437
:PLUGin:EDIagram:FETCh[:RESULT]:DCDistortion?, 437
:PLUGin:EDIagram:FETCh[:RESULT]:FALLtime?, 437
:PLUGin:EDIagram:FETCh[:RESULT]:HEIGHT?, 438
:PLUGin:EDIagram:FETCh[:RESULT]:HILevel?, 438
:PLUGin:EDIagram:FETCh[:RESULT]:JPPeak?, 438
:PLUGin:EDIagram:FETCh[:RESULT]:JRMSwere?, 439
:PLUGin:EDIagram:FETCh[:RESULT]:LOLevel?, 439
:PLUGin:EDIagram:GRAPh:CLIMits[?], 412
:PLUGin:EDIagram:GRAPh:TLIMits[?], 411
:PLUGin:EDIagram:GRAPh:TPO ints[?], 411
:PLUGin:EDIagram:INSTrument:ANALyzer[?], 396
:PLUGin:EDIagram:INSTrument:GENerator[?], 396
:PLUGin:JTOLerance:BSETup:ARTime[?], 398
:PLUGin:JTOLerance:BSETup:CLEver[?], 397
:PLUGin:JTOLerance:BSETup:FRTime[?], 398
:PLUGin:JTOLerance:CATalog?, 4
:PLUGin:JTOLerance:CONTinue, 415
:PLUGin:JTOLerance:DELete, 416
:PLUGin:JTOLerance:FETCh:DAT A?, 412, 413, 414
:PLUGin:JTOLerance:GRAPh:CLIMits[?], 412
:PLUGin:JTOLerance:GRAPh:TLIMits[?], 411
:PLUGin:JTOLerance:GRAPh:TPO ints[?], 411
:PLUGin:JTOLerance:INSTrument:ANALyzer[?], 396
:PLUGin:JTOLerance:INSTrument:GENerator[?], 396
:PLUGin:JTOLerance:MSETup:ALGOrithm[?], 405
:PLUGin:JTOLerance:MSETup:Bl Nary:SSIZE[?], 408
:PLUGin:JTOLerance:MSETup:CLAuto[?], 410
:PLUGin:JTOLerance:MSETup:CMAargin[?], 404
:PLUGin:JTOLerance:MSETup:FR EQuency:STARt[?], 402
:PLUGin:JTOLerance:MSETup:LINear:SSIZE[?], 408
:PLUGin:JTOLerance:MSETup:LOG:SSIZE[?], 409

542 Keysight M8000 Series of BER Test Solutions Programming Guide
Index

:PLUGin:JTOLerance:MSETup:MODEnt(?) , 404
:PLUGin:JTOLerance:MSETup:MT Points(?) , 403
:PLUGin:JTOLerance:MSETup:NPoints(?) , 403
:PLUGin:JTOLerance:MSETup:TIMEPlate:FILE(?) , 399, 400
:PLUGin:JTOLerance:NEW , 416
:PLUGin:JTOLerance:RESet , 417
:PLUGin:JTOLerance:RUN:HISTORY:CLEar , 419
:PLUGin:JTOLerance:RUN:HISTORY[:STATE](?) , 418
:PLUGin:JTOLerance:RUN:LOG?, 419
:PLUGin:JTOLerance:RUN:MESSAGE?, 420
:PLUGin:JTOLerance:RUN:PROGRESS?, 420
:PLUGin:JTOLerance:RUN:MESSEnge?, 420
:PLUGin:JTOLerance:RUN:PROGRESS, 420
:PLUGin:JTOLerance:RUN:[STATE]s(?) , 421, 422
:PLUGin:JTOLerance:START , 417
:PLUGin:JTOLerance:STEP , 418
:PLUGin:JTOLerance:STOP , 417

Plugin Output Level Commands

:PLUGin:OLEVel:ACQuisition:ACQuisition:ALOCation(?) , 386
:PLUGin:OLEVel:ACQuisition:COMPared(?) , 386
:PLUGin:OLEVel:ACQuisition:EBENabled(?) , 367
:PLUGin:OLEVel:ACQuisition:ERRORed(?) , 367
:PLUGin:OLEVel:ACQuisition:HILevel(?) , 388
:PLUGin:OLEVel:ACQuisition:LOLevel(?) , 368
:PLUGin:OLEVel:ACQuisition:TRESolution(?) , 369
:PLUGin:OLEVel:BREAK(?) , 361
:PLUGin:OLEVel:CATalog?, 361
:PLUGin:OLEVel:CONTINUE , 361
:PLUGin:OLEVel:DELETE , 362

Plugin Output Timing Commands

:PLUGin:OTIMing:ACQuisition:ALOCation(?) , 332
:PLUGin:OTIMing:ACQuisition:COMPared(?) , 334, 336, 432, 489
:PLUGin:OTIMing:ACQuisition:TEMPorative(?) , 335
:PLUGin:OTIMing:ACQuisition:ERROred(?) , 335
:PLUGin:OTIMing:ACQuisition:FJBER(?) , 333
:PLUGin:OTIMing:ACQuisition:OPTimization(?) , 334

:PLUGin:OLEVel:EVALuation:BERThresh(?) , 370
:PLUGin:OLEVel:EVALuation:MBQFactor(?) , 371
:PLUGin:OLEVel:FETCH:DATA?, 373
:PLUGin:OLEVel:FETCH:FREQUENCY?, 373
:PLUGin:OLEVel:FETCH:LEVEL:AMplitude?, 375
:PLUGin:OLEVel:FETCH:LEVEL:HIgh?, 375
:PLUGin:OLEVel:FETCH:LEVEL:MEAN?, 376
:PLUGin:OLEVel:FETCH:LEVEL:THMargin?, 376
:PLUGin:OLEVel:FETCH:LEVEL:HIGH?, 382
:PLUGin:OLEVel:FETCH:LEVEL:QFACtor?, 381
:PLUGin:OLEVel:FETCH:LEVEL:QOThreshold?, 381
:PLUGin:OLEVel:FETCH:LEVEL:QRBER?, 382
:PLUGin:OLEVel:NEW , 362
:PLUGin:OLEVel:RESet , 363
:PLUGin:OLEVel:RUN:HISTORY:COMPLETE , 390
:PLUGin:OLEVel:RUN:HISTORY[:STATE] , 390
:PLUGin:OLEVel:RUN:LOG?, 391
:PLUGin:OLEVel:RUN:MESSEnge?, 389
:PLUGin:OLEVel:RUN:PROGRESS?, 389
:PLUGin:OLEVel:RUN:STATUS?, 389
:PLUGin:OLEVel:SHOW:GRAPHICs?, 386
:PLUGin:OLEVel:SHOW:HISTORY[:STATE] , 386
:PLUGin:OLEVel:SHOW:SLLEGENDs?, 387, 447
:PLUGin:OLEVel:SHOW:SMPOINts ?, 386
:PLUGin:OLEVel:START , 363
:PLUGin:OLEVel:STEP , 363
:PLUGin:OLEVel:STOP , 364

Keysight M8000 Series of BER Test Solutions Programming Guide

543
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>:PLUGin:OTIMing:BREak</td>
<td>343</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:CONTinue</td>
<td>344</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:DELeate</td>
<td>342</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:EVAluation:BE RThresh[?],</td>
<td>336</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:EVAluation:ER Type[?],</td>
<td>338</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:EVAluation:ETJ Ber[?],</td>
<td>337</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:EVAluation:MB JPartition[?],</td>
<td>337</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:EVAluation:SM Points[?],</td>
<td>338</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:EVAluation:UN IT[?],</td>
<td>339</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:EVAluation:VS Cal[?],</td>
<td>339</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:FETCh:DATA?,</td>
<td>356</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:FETCh:DELay?,</td>
<td>357</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:FETCh:EDGE:RI GHt:RSQuare?,</td>
<td>351</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:FETCh:EDGE:RI GHt[POINTS]?</td>
<td>351</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:FETCh:EDGE[:LEFT]:POINTS?</td>
<td>350</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:FETCh:EDGE:FREQuency?,</td>
<td>358</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:FUNCTION:MINTeqration[?],</td>
<td>207, 208, 210</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:FREQuency[?],</td>
<td>142</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:INTerference:CMODE: AMPlitude[?],</td>
<td>196</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:INTerference:CMODE: GAIN[?],</td>
<td>197</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:INTerference:CMODE: SOURc[?],</td>
<td>196</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:INTerference:CMODE[ :STAte][?],</td>
<td>196</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:INTerference:DMODE: AMPlitude[?],</td>
<td>198</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:INTerference:DMODE: GAIN[?],</td>
<td>198, 199, 200</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:INTerference:DMODE: SOURc[?],</td>
<td>198</td>
</tr>
<tr>
<td>:PLUGin:OTIMing:INTerference:DMODE[ :STAte][?],</td>
<td>197</td>
</tr>
</tbody>
</table>

**Q**
- questionable status register, 36

**R**
- registers, 30

**S**
- safety summary, 3
- SCPI command keywords, 92
- command separators, 93
- command structure, 94
- command syntax, 92
Index

[:SOURce]:INTerference:HFRequency[:SOURce][?], 195, 204, 205
[:SOURce]:INTerference:LFRequency[:SOURce][?], 195
[:SOURce]:JITTer:CONFIGure[:DElay][?], 155, 156
[:SOURce]:JITTer:HFRequency:B UNCorrelate:AMPLitude[?], 164
[:SOURce]:JITTer:HFRequency:B UNCorrelate:DRATe[?], 185
[:SOURce]:JITTer:HFRequency:B UNCorrelate:FILTER[:SE lect][?], 165, 166
[:SOURce]:JITTer:HFRequency:E XTernal[:STATe][?], 158
[:SOURce]:JITTer:HFRequency:P ERiodic(*):AMPLitude[?], 161
[:SOURce]:JITTer:HFRequency:P ERiodic(*):FREQuency[?], 162
[:SOURce]:JITTer:HFRequency:P ERiodic(*)[:STATe][?], 160
[:SOURce]:JITTer:HFRequency:P MMode[?], 159
[:SOURce]:JITTer:HFRequency:R ANDom:AMPLitude[?], 188
[:SOURce]:JITTer:HFRequency:R ANDom:HFRequency[:STATe][?], 170
[:SOURce]:JITTer:HFRequency:R ANDom:FILTER[:LPASs][?], 169
[:SOURce]:JITTer:HFRequency:R ANDom[:STATe][?], 168
[:SOURce]:JITTer:HFRequency:S PECtrally:AMPLitude(*):[?], 172
[:SOURce]:JITTer:HFRequency:S PECtrally::FILTER[:LPASs][:STAT e][?], 172
[:SOURce]:JITTer:HFRequency:S PECtrally:PJTer[:LPASs][:STAT e][?], 173
[:SOURce]:JITTer:HFRequency:U NIt[?], 158
[:SOURce]:JITTer:LFRequency:PE Riodic:AMPLitude[?], 177
[:SOURce]:JITTer:LFRequency:PE Riodic:FREQuency[?], 177
[:SOURce]:JITTer:LFRequency:PE Riodic[:STATe][?], 177
[:SOURce]:JITTer:LFRequency:R SSClocking:AMPLitude[?], 179
[:SOURce]:JITTer:LFRequency:R SSClocking:FREQuency[?], 179
[:SOURce]:JITTer:LFRequency:R SSClocking:AMPLitude[?], 179
[:SOURce]:JITTer:LFRequency:R SSClocking:FREQuency[?], 179
[:SOURce]:JITTer:LFRequency:R SSClocking:AMPLitude[?], 179
[:SOURce]:JITTer:LFRequency:R SSClocking:FREQuency[?], 179
[:SOURce]:JITTer:LFRequency:S PECtrally:AMPLitude(*)[?], 184
[:SOURce]:JITTer:LFRequency:S PECtrally:AMPLitude[?], 184
[:SOURce]:JITTer:LFRequency:S PECtrally:AMPLitude[?], 184
[:SOURce]:JITTer:LFRequency:S PECtrally:AMPLitude[?], 184
[:SOURce]:JITTer:LFRequency:U NIt[?], 158
[:SOURce]:JITTer:SWEep:AMPLitude:MODE[?], 185
[:SOURce]:JITTer:SWEep:AMPLitude[?], 184
[:SOURce]:JITTer:SWEep:DATA:FI LE[?], 186
[:SOURce]:JITTer:SWEep:FREQuency:STARt[?], 182
[:SOURce]:JITTer:SWEep:FREQuency:STOP[?], 183
[:SOURce]:JITTer:SWEep:PLOT:INPut?, 185
[:SOURce]:JITTer:SWEep:PLOT?, 185
[:SOURce]:JITTer:SWEep:STEP:D istance[?], 183
[:SOURce]:JITTer:SWEep:STEP:TIME[?], 183
[:SOURce]:JITTer:SWEep:S WP? , 185
[:SOURce]:JITTer:SWEep:S WP?, 185
[:SOURce]:JITTer:SWEep:S WP?, 185
[:SOURce]:JITTer:SWEep:S WP?, 185
[:SOURce]:SSCLocking:FREQuency[?], 192
[:SOURce]:SSCLocking:PROFile[?], 191
[:SOURce]:SSCLocking:SHAPE[?], 192
[:SOURce]:SSCLocking:TYPE[?], 191
[:SOURce]:SSCLocking:STATe[?], 188
[:SOURce]:VOLTage:HIGH[?], 151
[:SOURce]:VOLTage:LOW[?], 152
[:SOURce]:VOLTage:OFFSet[?], 1
51
[:SOURce]:VOLTage:RANGE:AUT O[?], 153
[:SOURce]:VOLTage:RANGE:SELECT[?], 152
[:SOURce]:VOLTage:AMPLitude[?], 150
squelch, 38
standard event status register, 35
status byte, 34
Status Commands
:STATus:INSTRument:CDR:ULOCked?, 130
:STATus:INSTRument:CLOSes?, 127
:STATus:INSTRument:DLOSes?, 127
:STATus:INSTRument:ERROR?, 128
:STATus:INSTRument:OVERload?, 129
:STATus:INSTRument:RCInput?, 1
29
:STATus:INSTRument:RUN?, 129
:STATus:INSTRument:SALoss?, 128
:STATus:INSTRument:SLOSe?, 128
:STATus:OPERation:CONDition?, 1
123
:STATus:OPERation:ENABled[?], 1
23

545 Keysight M8000 Series of BER Test Solutions Programming Guide
### Status Register Structure

status register structure, 31
status reporting, 33
symbol coding, 128/130, 128/132, 42
8b10b, 41
bit, 39
sequences, 44
symbol width, 48
symbols, 38
sync and loop, 50

### System Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>:SYSTem:DCINterface[:PROGram]:UNLoad</td>
<td>506</td>
</tr>
<tr>
<td>:SYSTem:INFormation:LICenses:HOST[:LIST]?</td>
<td>504</td>
</tr>
<tr>
<td>:SYSTem:INFormation[:INSTrument]:LOCations?</td>
<td>501</td>
</tr>
<tr>
<td>:SYSTem:INInstrument:GROup:DEVice:DFine</td>
<td>496</td>
</tr>
<tr>
<td>:SYSTem:INInstrument:GROup:DEVice:Lete</td>
<td>496</td>
</tr>
<tr>
<td>:SYSTem:INInstrument:GROup:MEr</td>
<td>497</td>
</tr>
<tr>
<td>:SYSTem:INInstrument:GROup[:LI ST]</td>
<td>497</td>
</tr>
<tr>
<td>:SYSTem:INInstrument:GROup[:LITP]</td>
<td>500</td>
</tr>
<tr>
<td>:SYSTem:INInstrument:LRaning[:EQUalization][:STATus]</td>
<td>500</td>
</tr>
<tr>
<td>:SYSTem:INInstrument:LRaning[:LOG]:BLOCK?</td>
<td>499</td>
</tr>
<tr>
<td>:SYSTem:INInstrument:LRaning[:LOG]:CLEar</td>
<td>499</td>
</tr>
<tr>
<td>:SYSTem:INInstrument:RESET</td>
<td>496</td>
</tr>
<tr>
<td>:SYSTem:RESet:INInstrument</td>
<td>510</td>
</tr>
<tr>
<td>:SYSTem:RESet[:ALL]</td>
<td>510</td>
</tr>
</tbody>
</table>

### Trigger Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>:TRIGger:CMULtiplier:FREQuency:DETect</td>
<td>139</td>
</tr>
<tr>
<td>:TRIGger:CMULtiplier:FREQuency:DIVider?</td>
<td>139</td>
</tr>
</tbody>
</table>

This, 297

### Transition Filters

transition filters, 32
Index

:TRIGger:CMULtiplier:FREQuency ?, 138
:TRIGger:CMULtiplier:LBWidth?, 137
:TRIGger:DIRect[:FREQuency]:DE
tect, 137
:TRIGger:DIRect[:FREQuency][?], 
:TRIGger:INTernal[:SOURce][?], 135
:TRIGger:REference[:FREQuency ][?], 136
:TRIGger[:SOURce]:FREQuency?, 
:TRIGger[:SOURce][?], 134
trigger, external device, 53
trigger, sequencer, 51

V
voltages, changing, 29

X
XML, 46