



DEWETRON

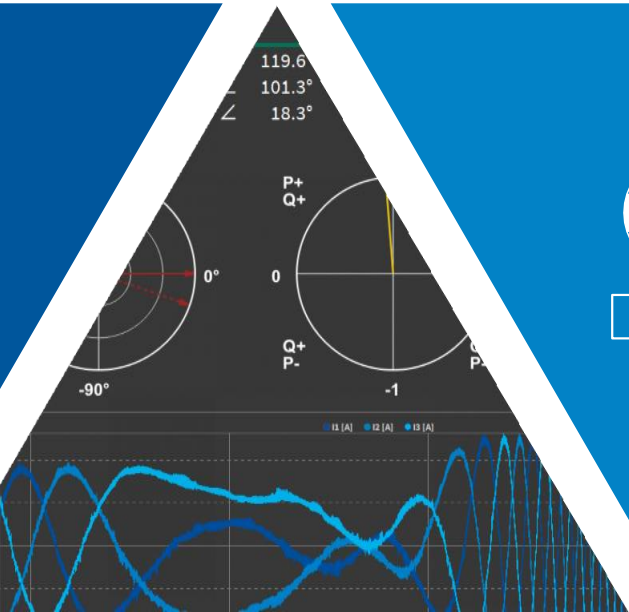


# OXYGEN R7.5

## SOFTWARE MANUAL



ISO 9001



# CONTENTS

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Scope . . . . .	3
1.2	Conventions and Structure . . . . .	3
1.3	Related Documents . . . . .	4
<b>2</b>	<b>Document History</b>	<b>5</b>
<b>3</b>	<b>Syntax Conventions</b>	<b>9</b>
3.1	SCPI Message Structure . . . . .	10
3.2	Parameter And Response Types . . . . .	11
<b>4</b>	<b>SCPI with OXYGEN</b>	<b>13</b>
4.1	Setup OXYGEN Measurement Software for SCPI use . . . . .	14
<b>5</b>	<b>Common Commands</b>	<b>15</b>
5.1	*IDN? . . . . .	15
5.2	*VER? . . . . .	15
5.3	*CLS . . . . .	16
5.4	*ESE . . . . .	16
5.5	*ESE? . . . . .	17
5.6	*ESR? . . . . .	17
5.7	*OPC . . . . .	18
5.8	*OPC? . . . . .	18
5.9	*RST . . . . .	18
5.10	*SRE . . . . .	19
5.11	*SRE? . . . . .	19
5.12	*STB? . . . . .	20
5.13	*TST? . . . . .	20
5.14	*WAI . . . . .	21
<b>6</b>	<b>Application Control</b>	<b>23</b>
6.1	Setup . . . . .	23
6.1.1	:SETup:LOAD . . . . .	23
6.1.2	:SETup:APPLY "XML-String" . . . . .	24
6.1.3	:SETup:SAVE "Path" . . . . .	25
6.1.4	:SETup:READ? . . . . .	26
6.1.5	:SETup:NAME? . . . . .	26
6.1.6	:SETup:ASync:LOAD "Path" . . . . .	27
6.1.7	:SETup:ASync:STATE? . . . . .	28

6.2	UI Control . . . . .	29
6.2.1	:SYSTem:DATE? . . . . .	29
6.2.2	:SYSTem:KLOCK {ON OFF} . . . . .	29
6.2.3	:SYSTem:KLOCK? . . . . .	30
6.2.4	:SYSTem:TIME? . . . . .	30
6.2.5	:SYSTem:TZONE? . . . . .	31
6.2.6	:COMMunicate:HEADer {ON OFF} . . . . .	31
6.2.7	:COMMunicate:HEADer? . . . . .	32
6.2.8	:COMMunicate:VERBose {ON OFF} . . . . .	32
6.2.9	:COMMunicate:VERBose? . . . . .	33
<b>7</b>	<b>Acquisition Control</b>	<b>35</b>
7.1	:ACQUisition:START . . . . .	35
7.2	:ACQUisition:STOP . . . . .	35
7.3	:ACQUisition:RESTART . . . . .	35
7.4	:ACQUisition:STATe? . . . . .	36
<b>8</b>	<b>Recording Control</b>	<b>37</b>
8.1	:STORe:FILE:NAME "PATH" . . . . .	37
8.2	:STORe:FILE:NAME? . . . . .	38
8.3	:STORe:START . . . . .	38
8.4	:STORe:PAUSE . . . . .	38
8.5	:STORe:STOP . . . . .	39
8.6	:STORe:STATe? . . . . .	39
8.7	WAVEform access . . . . .	40
8.8	:STORe:WAVEform:MODE? . . . . .	40
8.9	:STORe:WAVEform:MODE {Continuous   Eventbased   Disabled} . . . . .	40
8.10	:STORe:WAVEform:CONTInuous . . . . .	41
8.11	:STORe:WAVEform:EVENTbased . . . . .	41
8.12	STORe:WAVEform:DISabled . . . . .	41
8.13	:STORe:WAVEform:PREtime? . . . . .	42
8.14	:STORe:WAVEform:PREtime {ON OFF <Nrf>}   {{ON OFF},<Nrf>} . . . . .	42
8.15	:STORe:WAVEform:PAFTer? . . . . .	43
8.16	:STORe:WAVE:PAFTer {ON OFF <Nrf>}   {{ON OFF},<Nrf>} . . . . .	43
8.17	:STORe:WAVEform:POSTtime? . . . . .	44
8.18	:STORe:WAVEform:POSTtime {ON OFF <Nrf>}   {{ON OFF},<Nrf>} . . . . .	44
8.19	:STORe:STATistics? . . . . .	45
8.20	:STORe:STATistics {ON OFF <Nrf>}   {{ON OFF},<Nrf>} . . . . .	45
8.21	:STORe:AUTOStart? . . . . .	46
8.22	:STORe:AUTOStart <Boolean> . . . . .	46
8.23	:STORe:RACQuisition? . . . . .	46
8.24	:STORe:RACQuisition <Boolean> . . . . .	47
8.25	:STORe:SAFTer? . . . . .	47
8.26	:STORe:SAFTer {ON OFF <Nrf>}   {{ON OFF},<Nrf>} . . . . .	48
8.27	:STORe:ADVanced? . . . . .	48
8.28	:STORe:ADVanced <Boolean> . . . . .	49
<b>9</b>	<b>Channel List Access</b>	<b>51</b>
9.1	:CHANNEllist:NAMes? . . . . .	51
9.2	:CHANNEllist:IDs? . . . . .	52
9.3	:CHANNEllist:ITEM<ChannelID>:ATTR . . . . .	52
9.4	:CHANNEllist:ITEM<ChannelID>:ATTR:NAMes? . . . . .	55

9.5	:CHANNELlist:ITEM<ChannelID>:ATTR:VAL?	55
9.6	:CHANNELlist:ITEM<ChannelID>:REMove	56
9.7	:CHANNELlist:ADDchannel?	57
9.8	:CHANNELlist:PROPerTy?	58
9.9	:CHANNELlist:PROPerTy	59
9.10	:CHANNELlist:CONSTRaint?	60
9.11	:CHANNELlist:TIMing:HIGHest?	60
9.12	:CHANNELlist:TIMing:LOWest?	61
9.13	:CHANNELlist:SATuration:VALue?	61
9.14	:CHANNELlist:SATuration:RESet	62
9.15	:CHANNELlist:FIResponse	63
9.16	<b>Channel Actions</b>	64
9.17	:CHANNELlist:ITEM<ChannelID>:ACTion:<action>:AVAIl?	65
9.18	:CHANNELlist:ITEM<ChannelID>:ACTion:<action>:EXEc	65
9.19	CHANNELlist:ACTion:<action>:AVAIl? <channelname_or_id>[,<channelname_or_id>[,..]]	66
9.20	CHANNELlist:ACTion:<action>:EXEc <channelname_or_id>[,<channelname_or_id>[,..]]	66
<b>10</b>	<b>Measurement Values</b>	<b>67</b>
10.1	:RATE {<num>[<unit>] NONE}	67
10.2	:RATE?	67
10.3	:NUMeric:NORMal:ITEMS <channel>[,<channel>[,...]]	68
10.4	:NUMeric:NORMal:ITEMS?	68
10.5	:NUMeric:NORMal:ITEM<x> <channel>	69
10.6	:NUMeric:NORMal:ITEM<x>?	69
10.7	:NUMeric:NORMal:CLEar {ALL <NUM>[,<NUM>]}	70
10.8	:NUMeric:NORMal:DELeTe <NUM>[,<NUM>]	70
10.9	:NUMeric:NORMal:NUMBER {<num> ALL}	71
10.10	:NUMeric:NORMal:NUMBER?	71
10.11	:NUMeric:NORMal:DIM<x> { <i_max>   <i_list>   MAX }	72
10.12	:NUMeric:NORMal:DIM<x>?	73
10.13	:NUMeric:NORMal:DIMS?	73
10.14	:NUMeric:NORMal:FORMat {ASCII BIN_INTEL BIN_MOTOROLA}	74
10.15	:NUMeric:NORMal:FORMat?	74
10.16	:NUMeric:NORMal:VALue? [<NUM>]	75
<b>11</b>	<b>External Data Logging (ELOG)</b>	<b>77</b>
11.1	:ELOG:ITEMs <channel>[,<channel>[,...]]	77
11.2	:ELOG:ITEMs?	78
11.3	:ELOG:PERiod <Duration>	78
11.4	:ELOG:PERiod?	79
11.5	:ELOG:CALCulations {AVG MIN MAX RMS}	79
11.6	:ELOG:CALCulations?	80
11.7	:ELOG:PERiod?	80
11.8	:ELOG:FORMat {ASCII BIN_INTEL BIN_MOTOROLA}	80
11.9	:ELOG:FORMat?	81
11.10	:ELOG:TIMestamp {OFF REL ABS ELOG}	81
11.11	:ELOG:TIMestamp?	82
11.12	:ELOG:START	82
11.13	:ELOG:FETCh? [<NUM>]	84
11.14	:ELOG:STOP	85
11.15	:ELOG:RESet	85
11.16	:ELOG:STATe?	86

<b>12 Data Streaming (DSTREAM)</b>	<b>87</b>
12.1 :DSTream:ITEMs[<GRP>] <channel>[,<channel>[,...]]	88
12.2 :DSTream:ITEMs[<GRP>]?	88
12.3 :DSTream:PORT[<GRP>] <PORT>	89
12.4 :DSTream:PORT[<GRP>]?	89
12.5 :DSTream:INIT [<GRP>   ALL]	90
12.6 :DSTream:STARt [<GRP>   ALL]	90
12.7 :DSTream:STOP [<GRP>   ALL]	91
12.8 :DSTream:DELeTe [<GRP>   ALL]	91
12.9 : DSTream:RESet	92
12.10 : DSTream:STATe[<GRP>]?	92
12.11 : DSTream:TRIG[<GRP>] {ON   OFF}	93
12.12 : DSTream:TRIG[<GRP>]?	93
12.13 :DSTream:REPLAY[<GRP>] {LIVE   BULK}	94
12.14 : DSTream:REPLAY[<GRP>]?	94
12.15 :DSTream:INTERVAL[<GRP>] <INTV>	95
12.16 : DSTream:INTERVAL[<GRP>]?	95
<b>13 EXPORT Commands</b>	<b>97</b>
13.1 :EXPort:DIRectory	97
13.2 :EXPort:DIRectory "path"	97
13.3 :EXPort:AUTO?	98
13.4 :EXPort:AUTO {ON   OFF}	98
13.5 <b>:EXPort:ITEMS Commands and Queries</b>	99
13.6 :EXPort:ITEMS[:LIST]?	99
13.7 :EXPort:ITEMS[:LIST] <channel>[,<channel>[,...]]	100
13.8 :EXPort:ITEMS:AVAIL?	100
13.9 :EXPort:ITEMS:CLear	101
13.10 :EXPort:ITEMS:ADD <channel>[,<channel>[,...]]	101
13.11 :EXPort:ITEMS:DELeTe <channel>[,<channel>[,...]]	102
<b>14 Marker Commands</b>	<b>103</b>
14.1 :MARKer:ADD <label>[,<description>   <time>   ,<description>,<time>]]	103
<b>15 Synchronisation Queries</b>	<b>105</b>
15.1 :SYNC:STATe?	105
15.2 :SYNC:ENCLOSURES[:LIST]?	105
15.3 :SYNC:ENClosure[<ENR>]:NAME?	106
15.4 :SYNC:ENClosure[<ENR>]:SERIAL?	106
15.5 :SYNC:ENClosure[<ENR>]:NODEName?	107
15.6 :SYNC:ENClosure[<GRP>]:IN[<INR>]:MODE?	107
15.7 :SYNC:ENClosure[<GRP>]:OUTPUTS[:LIST]?	108
15.8 :SYNC:ENClosure[<GRP>]:OUT[<ONR>]:NAME?	108
15.9 :SYNC:ENClosure[<GRP>]:OUT[<ONR>]:CONNECTor?	109
15.10 :SYNC:ENClosure[<GRP>]:OUT[<ONR>]:MODE?	109
<b>16 Measurement Screen Commands</b>	<b>111</b>
16.1 :SCReen:INSTRUMENTs:OUTputchannel:STARt	111
16.2 :SCReen:INSTRUMENTs:OUTputchannel:PAUSE	111
16.3 :SCReen:INSTRUMENTs:OUTputchannel:STOP	112
16.4 :SCReen:INSTRUMENTs:OUTputchannel:STATe?	112
16.5 :SCReen:SAVE "PATH"	113

16.6	:SCReen:ITEM<ScreenNumber>:SAVE "PATH"	114
<b>17</b>	<b>Measurement Report Commands</b>	<b>115</b>
17.1	:REPort:SAVE[:ALL] "PATH"	115
17.2	:REPort:ITEM<PageNumber>:SAVE "PATH"	116
<b>18</b>	<b>Measurement Header Data</b>	<b>117</b>
18.1	:HEADer:ADD <key>,<description>	117
18.2	:HEADer:GET? <key>	118
18.3	:HEADer:KEYs?	118
18.4	:HEADer:SET <key>,<description>	119
18.5	:HEADer:DELeTe <key>[,<key>[,...]]	120
18.6	:HEADer:VALues?	120
<b>19</b>	<b>Utility Commands</b>	<b>121</b>
19.1	:SYSTem:VERsion?	121
19.2	:SYSTem:HELP:HEADers?	121
<b>20</b>	<b>Trigger Events</b>	<b>123</b>
20.1	:TRIGger[:GET]?	123
20.2	:TRIGger:RESet	124
20.3	:TRIGger:ADDevent	124
20.4	<b>:TRIGger:EvEnt&lt;context&gt; Commands and Queries</b>	124
20.5	:TRIGger:EvEnt<event-number>[:SETup]?	125
20.6	:TRIGger:EvEnt<event-number>[:SETup] {<String>}   {{ON OFF},<String>}	126
20.7	:TRIGger:EvEnt<event-number>:VALId?	126
20.8	:TRIGger:EvEnt<event-number>:DELeTe	127
20.9	:TRIGger:EvEnt<event-number>:ADDCondition	127
20.10	:TRIGger:EvEnt<event-number>:ADDAction	128
20.11	:TRIGger:EvEnt<event-number>:CONDition<condition-number>:GET?	129
20.12	:TRIGger:EvEnt<event-number>:CONDition<condition-number>:VALId?	130
20.13	:TRIGger:EvEnt<event-number>:CONDition<condition-number>:DELeTe	130
20.14	:TRIGger:EvEnt<event-number>:CONDition<condition-number>:HIGHlevel:SETup <nrf>,<literal> <nrf>,<String>[,<String>[,...]]	131
20.15	:TRIGger:EvEnt<event-number>:CONDition<condition-number>:LOWlevel:SETup <nrf>,<literal> <nrf>,<String>[,<String>[,...]]	132
20.16	:TRIGger:EvEnt<event-number>:CONDition<condition-number>:INwindow:SETup <nrf>,<nrf>,<literal> <nrf>,<literal> <nrf>,<String>[,<String>[,...]]	133
20.17	:TRIGger:EvEnt<event-number>:CONDition<condition-number>:OUTwindow:SETup <nrf>,<nrf>,<literal> <nrf>,<literal> <nrf>,<String>[,<String>[,...]]	134
20.18	:TRIGger:EvEnt<event-number>:CONDition<condition-number>:KEYBoard:SETup	135
20.19	:TRIGger:EvEnt<event-number>:CONDition<condition-number>:TIME:SETup	136
20.20	:TRIGger:EvEnt<event-number>:ACTion<action-number>:GET?	137
20.21	:TRIGger:EvEnt<event-number>:ACTion<action-number>:VALId?	137
20.22	:TRIGger:EvEnt<event-number>:ACTion<action-number>:DELeTe	138
20.23	:TRIGger:EvEnt<event-number>:ACTion<action-number>:RECORDing:SETup <literal>	138
20.24	:TRIGger:EvEnt<event-number>:ACTion<action-number>:DIGOut:SETup <lit- eral> <nrf>,<literal> <nrf>,<literal>,<String>[,<String>[,...]]	139
20.25	:TRIGger:EvEnt<event-number>:ACTion<action-number>:ALARm:SETup <literal>,<lit- eral> <nrf>,<literal> <nrf>,<literal>,<String>[,<String>[,...]]	140
20.26	:TRIGger:EvEnt<event-number>:ACTion<action-number>:MARKer:SETup <String>,<lit- eral>	141

20.27	:TRIGger:EVent<event-number>:ACTIon<action-number>:SNAPshot:SETup <literal>,<nrf>,<String>[,<String>[,...]]	142
20.28	:TRIGger:EVent<event-number>:ACTIon<action-number>:ARM:SETup <literal>	143
<b>21</b>	<b>Analysis Control</b>	<b>145</b>
21.1	:ANALysis:ACTIve?	145
21.2	:ANALysis:OPEN <file_name> <path>[,<file_name> <path>[,...]]	146
21.3	:ANALysis:CLOSe	147
21.4	:ANALysis:FILEs?	147
<b>22</b>	<b>Error Handling</b>	<b>149</b>
22.1	:SYSTem:ERRor[:NEXT]?	149
22.2	:SYSTem:ERRor:ALL?	149
22.3	:SYSTem:ERRor:CODE[:NEXT]?	150
22.4	:SYSTem:ERRor:CODE:ALL?	150
22.5	:SYSTem:ERRor:COUNt?	150
22.6	:SYSTem:ERRor:ENABle:ADD (<num>:<num>)	151
22.7	:SYSTem:ERRor:ENABle:DELeTe (<num>:<num>)	151
22.8	:SYSTem:ERRor:ENABle[:LIST]?	152
<b>23</b>	<b>Error Codes</b>	<b>153</b>
<b>24</b>	<b>EXAMPLES</b>	<b>155</b>
24.1	Fetch Online Measurement Data	155
24.2	Store Measurement Data on Device	155
24.3	Set Channel Properties	156
24.3.1	Set a bool Item example	156
24.3.2	Set a string Item example	156
24.3.3	Set a floating point Item example	157
24.3.4	Set an enum item example	157
24.3.5	Set scalar item example	157
24.3.6	Set range item example	158
24.3.7	Change sample rate with the sample rate divider example	158

**Technical Reference Manual**





## INTRODUCTION

This Technical Reference Document describes the Standard Commands for Programmable Instruments (SCPI) remote control interface to communicate with the DEWETRON Oxygen Software (OXYGEN). The intended audience of this document are instrument programmers who are responsible for writing SCPI-based programs to control the OXYGEN software product.

### 1.1 Scope

This document describes a set of SCPI commands and queries usable to programmers of SCPI-based devices and controllers via Ethernet based TCP networking, and interfacing to the OXYGEN software. This document also defines the basic TCP operation parameters necessary for a successful connection attempt to the OXYGEN software.

### 1.2 Conventions and Structure

Notes provide useful information about the context and awareness of special emphasis. Examples provide overview of real world data transmission.

The organization of this document is as follows to provide you a programmer-friendly guide for communicating with the OXYGEN software:

- Chapter “Syntax Conventions” describes the syntax conventions used.
- Chapter “SCPI with OXYGEN” provides information about the TCP networking and an overview of the mapping from the OXYGEN software model to corresponding SCPI systems.
- Chapter “Common Commands” describes the common SCPI commands and queries that are available for the OXYGEN software.
- Chapter “Application Control” describes the SCPI commands and queries that configure basic operation of the OXYGEN software.
- Chapter “Acquisition Control” describes the SCPI commands and queries that act on the acquisition module of the OXYGEN software.
- Chapter “Recording Control” describes the SCPI commands and queries that act on the recording module of the OXYGEN software.
- Chapter “Channel List Access” describes the access to channels and their properties of the OXYGEN softwareChapter “Measurement Values” describes the SCPI commands and queries that act on the retrieval of measurement values of the OXYGEN software.

## 1.3 Related Documents

Refer to the following documents for more information:

- *OXYGEN Feature Manual*. This document describes the operation of OXYGEN software and its software and related hardware components.
- *OXYGEN Power Technical Reference Manual*. This document describes the operation of OXYGEN software as a highly configurable and most accurate Power Analyzer.
- *Standards Commands for Programmable Instruments (SCPI)*, Volume 1-4, Version 1999.0 May 1999, SCPI Consortium.
- *Standard digital interface for programmable instrumentation – Part 2: Codes, formats, protocols, and common commands*, IEC 60488-2 First Edition 2004-5, IEEE.

**DOCUMENT HISTORY**

<b>Date</b>	<b>Changes</b>	<b>Revision</b>
08/30/2017	First release version of the document.	1.5
10/02/2017	<ul style="list-style-type: none"> <li>• Added section on document information and history.</li> <li>• Added :NUM:DIM command and query.</li> <li>• Added :NUM:DIMS query.</li> <li>• Updated description of :NUM:VAL query for array channels.</li> <li>• Updated parameter description of :NUM:NORM:ITEM command and query.</li> <li>• Updated explanation of :RATE command for array channels.</li> </ul>	1.6
09/29/2018	<ul style="list-style-type: none"> <li>• Removed loose Reference</li> </ul>	1.6.1
10/09/2018	<ul style="list-style-type: none"> <li>• Added :ELOG commands and queries</li> <li>• Added:DStream command and queries</li> <li>• Updated ELOG:ITEMs &amp; DSTREAM:ITEMs entries with error handling</li> <li>• Added ELOG limitation info</li> <li>• Added ELOG:PERiod limitation info</li> <li>• Update ELOG supported channels</li> </ul>	1.7
02/11/2019	<ul style="list-style-type: none"> <li>• Added: ELOG new timestamp format</li> </ul>	1.8
03/18/2019	<ul style="list-style-type: none"> <li>• Trigger support for STORE subsystem</li> </ul>	1.9
04/28/2020	<ul style="list-style-type: none"> <li>• Updated documentation</li> </ul>	1.11
06/09/2020	<ul style="list-style-type: none"> <li>• Corrected quotation mark character for simple copy&amp;paste</li> </ul>	1.12

Date	Changes	Revision
09/01/2020	<ul style="list-style-type: none"> <li>• Added :SETUP:ASYNC:LOAD command for async load configuration from file</li> <li>• Added :SETUP:ASYNC:STATE? query for the async loading state</li> <li>• Added :SETUP:NAME? query for the current setup name</li> </ul>	1.13
10/01/2020	<ul style="list-style-type: none"> <li>• Fixed Typo</li> </ul>	1.14
12/22/2020	<ul style="list-style-type: none"> <li>• Added SYNC:STATE? query</li> </ul>	1.15
02/23/2021	<ul style="list-style-type: none"> <li>• Added HEAD commands and queries</li> </ul>	1.16
05/10/2021	<ul style="list-style-type: none"> <li>• Extended HEAD commands for numeric constants</li> </ul>	1.17
08/12/2021	<ul style="list-style-type: none"> <li>• Add channel list zero action for scaling offset reset</li> </ul>	1.18
08/24/2021	<ul style="list-style-type: none"> <li>• Add export commands and queries</li> <li>• Update wrong sync state documentation</li> </ul>	1.19
01/13/2022	<ul style="list-style-type: none"> <li>• Support for binary :NUM:VAL? formats</li> </ul>	1.20
08/08/2022	<ul style="list-style-type: none"> <li>• Added :CHANNELlist:PROPerTy command for manipulate channel attributes</li> <li>• Added :CHANNELlist:CONSTRaint query</li> <li>• Added :CHANNELlist:TIMing? query</li> <li>• Added Channel:ID? with Parameter description</li> <li>• Added CHANNELlist:SATuration? query</li> <li>• Added CHANNELlist:SATuration? query</li> <li>• Documentation update for restructured text</li> </ul>	1.21
12/15/2022	<ul style="list-style-type: none"> <li>• Added :CHANNELlist:SATuration ALL</li> <li>• Added :CHANNELlist:SATuration:RESet</li> <li>• Documentation update</li> </ul>	1.22
06/15/2023	<ul style="list-style-type: none"> <li>• Added :CHANNELlist:FIResponse? to query a channel filter response</li> </ul>	1.23

Date	Changes	Revision
09/15/2023	<ul style="list-style-type: none"> <li>• Added :SCReen:SAVE for saving one or more OXYGEN screens</li> <li>• Added :SCReen:ITEM&lt;ScreenNumber&gt;:SAVE for saving one OXYGEN screen</li> <li>• Added :REPort:SAVE for saving one or more OXYGEN reports</li> <li>• Added :REPort:ITEM&lt;ReportNumber&gt;:SAVE for saving one OXYGEN report</li> </ul>	1.24
12/15/2023	<ul style="list-style-type: none"> <li>• Added :TRIGger event subsystem to setup OXYGEN triggers</li> <li>• Added :STORE:WAVEform subsystem to setup OXYGEN trigger waveform modes</li> <li>• Added :STORE:STATistics command</li> <li>• Added :STORE:AUTOStart? query</li> <li>• Added :STORE:AUTOStart command</li> <li>• Added :STORE:RACQuisition? query</li> <li>• Added :STORE:RACQuisition command</li> <li>• Added :STORE:SAFTer? query</li> <li>• Added :STORE:SAFTer command</li> <li>• Added :STORE:ADVanced? query</li> <li>• Added :STORE:ADVanced command</li> </ul>	1.25
04/15/2024	<ul style="list-style-type: none"> <li>• Added :ANALysis subsystem</li> </ul>	1.26
06/15/2024	<ul style="list-style-type: none"> <li>• Added :CHANNELlist actions bridge balance, shunt on and shunt off</li> <li>• Added lower and upper rearm level to in/out window trigger condition. Attention: parameter set changed from previous version</li> <li>• Added DStream:INTERVAL? query</li> <li>• Update :CHANNELlist:CONSTRAINT? return string for ranges</li> <li>• Update calculation for :CHANNELlist:TIMing:HIGHeSt? and :CHANNELlist:TIMing:LOWeSt?</li> </ul>	1.27
10/04/2024	<ul style="list-style-type: none"> <li>• Added :EXPort:ITEMS command and queries</li> <li>• Remove window parameter for ACTUAL mode from trigger event action snapshot command</li> </ul>	1.28
12/18/2024	<ul style="list-style-type: none"> <li>• Added :SYNC:ENCLOSURE and :SYNC:ENCLOSURES groups and several queries</li> <li>• Added CHANNEL_ID and CHANNEL_ID_VECTOR config item type descriptions</li> </ul>	1.29

<b>Date</b>	<b>Changes</b>	<b>Revision</b>
02/05/2025	<ul style="list-style-type: none"><li>• Added :CHANNEL:ADD? and :CHANNEL:ADD to query software channel and add software channel</li></ul>	1.30

## SYNTAX CONVENTIONS

The SCPI (Standard Commands for Programmable Instruments) is a universal ASCII-based textual remote programming language for electronic test and measurement (T&M) instruments. Based on the IEC 60488-2 specification, the remote transportation interface to the OXYGEN software is Ethernet based, as defined by the IEEE 802.3 working group, and the networking protocol is TCP.

Note: Check the reference manual of your DEWETRON measurement device for the availability and configuration options of the Ethernet interface.

The SCPI defines messages in the form of commands and queries to control the operation and functions for T&M instruments. The related topics below describe the syntax of these commands and queries, and the conventions that the OXYGEN software uses to process them. Commands modify settings and parameters of the OXYGEN software. Further, commands tell the OXYGEN software to perform a specific action. Queries cause the OXYGEN software to return data and status information.

Refer to the following table for the symbols used to describe the syntax of commands and queries.

Symbol	Meaning
< >	A defined element
::=	Is defined as
	Exclusive OR
{ }	Option group; one element is required
[ ]	Optional; can be omitted
...	Previous elements can be repeated



### 3.1 SCPI Message Structure

The SCPI messages may consist of five element types, defined in the following table.

Element	Meaning
<Header>	This element represents the basic command name. In case if the header ends with a question mark, the command is a query. The header may begin with a colon or asterisk character. A header consists of one or more <Mnemonic> elements, representing the system or subsystem of the command or query category. The only exception
<Mnemonic>	A part of the <Header>. Some commands have only one <Mnemonic> defined by their <Header>. Mnemonics can have a short and a long form. The syntax of the mnemonics in this document is to capitalize the short form while the long form finalizes the <Mnemonic> in lower case characters.
<Parameter>	This is a parameter used as input to the addressed command or query identified by the <Header>. Some commands have no parameters while others have multiple parameters. A <Space> separates parameters from the <Header>, while a <Comma> separates parameters from each other.
<Comma>	A single comma separates parameters of multiple-parameter commands.
<Space>	A white space character separates the <Header> from its <Parameter> list.

Commands have the structure:

```
<Header> [<Space><Parameter> [<Comma><Parameter>] . . . ]
```

Queries have the structure:

```
<Header>? [<Space><Parameter> [<Comma><Parameter>] . . . ]
```

SCPI defines systems and subsystems as a grouping for mnemonics. For example, the mnemonics operating on the acquisition control are summarized within a group :ACQUISITION, containing the mnemonics START, STOP, PAUSE and STATE. Subsystems are additional groupings within a system or a subsystem. Some SCPI systems have no subsystem while others have many subsystems. Not all functions are grouped in systems or subsystems. Therefore, this document uses a logical grouping of function blocks.

A SCPI message can be composed of multiple commands and queries by separating them with a semicolon. The following rules apply when concatenating commands and queries:

1. Separate completely different headers by a semicolon and by the beginning colon on all headers. For example, the command :ACQUISITION:START and the query :STORE:STATE?, can be concatenated into the following single message:  
 :ACQUISITION:START; :STORE:STATE?
2. If concatenated headers differ by only the last mnemonics, you can abbreviate the subsequent headers and eliminate the beginning colon. For example, to concatenate the command :ACQUISITION:START and the query :ACQUISITION:STATE?, the following single message can be formed:  
 :ACQUISITION:START; STATE?

3. When concatenating multiple queries, a single response message is been generated by concatenating the responses to all queries.
4. The processing order of concatenated commands and queries is the order received.

An <EOM> message terminator must terminate each SCPI message. The OXYGEN software allows the usage of LF and CR LF characters as a valid message termination. The OXYGEN software evaluates only terminated messages.

### 3.2 Parameter And Response Types

Angle brackets, such as <num>, indicate parameters of commands. Angle brackets containing the parameter type indicate a response type of queries, such as <String>. There are several different types of parameters and response types defined by the SCPI and used by the RC\_SCPI of the OXYGEN software, as listed in the following table:

Parameter Type	Description	Example
Arbitrary Block	A block of ASCII raw data described by a simple header of the format: #[1-9][0-9]* The hashtag indicates the beginning of an arbitrary block header. The following digit indicates the number of subsequent digits specifying the length of the data in bytes.	#14AHOI
Arbitrary ASCII String	An unquoted block of ASCII character terminated by a <CR>	Lorem Ipsum
String	Quoted alphanumeric characters, by either a single or a double quote.	"17E24 * 37E87 = BIG NUMBER" 'Lorem Ipsum'
Character	Unquoted alphanumeric characters in the format [a-zA-Z][a-zA-Z0-9 _]* The first character must not be a digit.	Test123
Boolean	Boolean numbers or values. The response type returns only boolean numbers 0 and 1.	ON or ≠ 0 OFF or 0
Hexadecimal	Hexadecimal coded integers of the form #Hxxxx	#HFF
Octal	Octal coded integers of the form #Qxxxx	#Q77
Binary	Binary coded integers of the form #Bxxxx	#B1101
NR1	Integers	0, 1, 42, -37
NR2	Decimal numbers 3.432,	-74.43
NR3	Decimal numbers in scientific notation	1.55433E+6
NRf	Flexible decimal numbers that may be of type NR1, NR2 or NR3	See NR1, NR2 and NR3 examples

SCPI defines special numeric values as possible parameters for commands and queries. If mentioned in the specific command or query documentation, you can use the following special character values



instead of the numeric values:

<b>Special Character Value</b>	<b>Description</b>	<b>Numeric Value</b>
NAN	Not a Number, as defined in IEEE 754 for 32-bit floating point numbers	9.91 E 37
INFINITY NINFINITY	Infinity and negative infinity, as defined in IEEE 754 for 32-bit floating point numbers	9.9 E 37 -9.9 E 37
MINIMUM MAXIMUM	Denotes the minimum and maximum value of a range of numeric values	For the range 20..100 MIN = 20 MAX = 100
DEFAULT	The preset value which is set by the *RST command	See NR1, NR2 and NR3 examples

## SCPI WITH OXYGEN

The OXYGEN SCPI remote control (RC\_SCPI) interface is available via TCP networking using the TCP endpoint port number 10001. You can easily use HyperTerminal or PuTTY in raw TCP configuration and ASCII encoding to test the connection and issue SCPI commands and queries.

Note: The RC\_SCPI interface permits only one active client connection to transmit TCP data. Multiple connections are not possible.

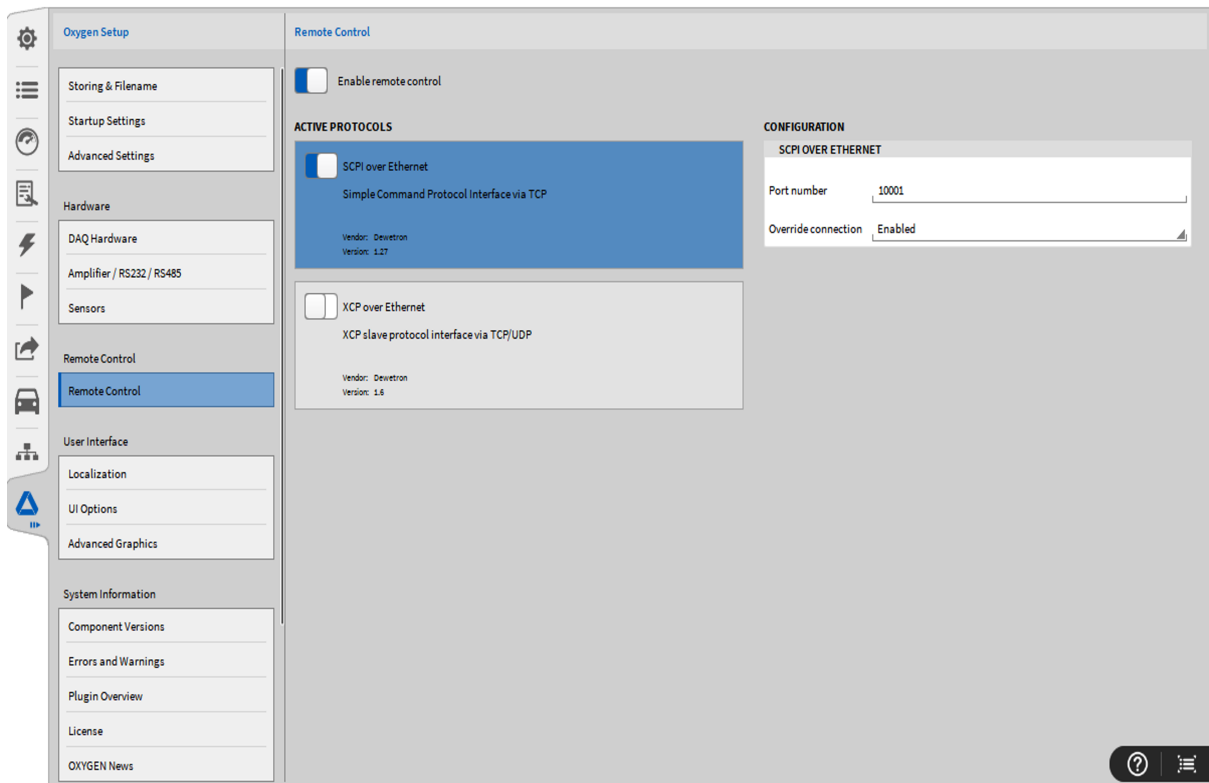
The OXYGEN software executes the received SCPI messages in sequential order. Each SCPI command and query is blocking and does not allow any further commands or queries executed until it is finished.

The internal application model of the OXYGEN software consists of major functions, potentially mapped to SCPI systems. The following table gives an overview to those mappings:

<b>Part of OXYGEN</b>	<b>Description</b>	<b>SCPI Headers and Systems</b>
Application Control	Control the application state and configuration via setups	:SETup:* :SYSTEM:* :COMMUNICATE:*
Acquisition Control	Control the acquisition state of the application.	:ACQUISITION:*
Recording Control	Control the recording state of the application	:STORE:*
Measurement Values	Read out actual measurement values.	:RATE:* :NUMERIC:*
Channel Access	Get channel information and get/set properties.	:CHANNELlist:*

## 4.1 Setup OXYGEN Measurement Software for SCPI use

1. Navigate to System Settings on the Dewetron Measurement device with Oxygen Software
2. Go to “Remote Control” Tab on the left
3. Enable the Remote Control feature and select the protocol type
4. Change the TCP/IP Port number if needed



## COMMON COMMANDS

### 5.1 \*IDN?

<b>Syntax</b>	<b>*IDN?</b>
<b>Description</b>	Query the ID string of the instrument
<b>Parameter</b>	None
<b>Explanation</b>	The query returns a colon-separated four-field ASCII string. The first field contains the manufacturer name, the second field is the product name, the third field is the device serial number, and the fourth field is the product revision number
<b>Return Format</b>	<Arbitrary ASCII String>
<b>Example</b>	<pre>-&gt; *IDN? &lt;- DEWETRON, OXYGEN, 0, 2.5.0</pre>

### 5.2 \*VER?

<b>Syntax</b>	<b>*VER?</b>
<b>Description</b>	Query the software and SCPI interface version string
<b>Parameter</b>	None
<b>Explanation</b>	The query returns the version information for the relevant parts. The SCPI version, the RC_SCPI plugin version and the OXYGEN version is mandatory and always reported
<b>Return Format</b>	<Character>,<String>[,<Character>,<String>]...
<b>Example</b>	<pre>-&gt; *VER? &lt;- SCPI, "1999.0", RC_SCPI, "1.10", OXYGEN, "5.1.1"</pre>

### 5.3 \*CLS

<b>Syntax</b>	<b>*CLS</b>
<b>Description</b>	Clears the standard event register, extended event register, and error queue
<b>Parameter</b>	None
<b>Explanation</b>	If the *CLS command is located immediately after the program message terminator, the output queue is also cleared
<b>Example</b>	-> *CLS

### 5.4 \*ESE

<b>Syntax</b>	<b>*ESE &lt;num&gt;</b>								
<b>Description</b>	Sets the standard event status enable register								
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;num&gt;</td> <td>Integer</td> <td>0 - 255</td> <td>0</td> </tr> </tbody> </table>	Name	Type	Range	Default	<num>	Integer	0 - 255	0
Name	Type	Range	Default						
<num>	Integer	0 - 255	0						
<b>Explanation</b>	<ul style="list-style-type: none"> <li>Specify the value as a sum of decimal values of each bit</li> <li>For example, specifying *ESE 251 will cause the standard enable register to be set to "11111011". In this case, bit 2 of the standard event register is disabled which means that bit 5 (ESB) of the status byte register is not set to 1, even if a "query error" occurs</li> <li>A query using ESE? will not clear the contents of the standard event enable register</li> </ul>								
<b>Example</b>	<pre>-&gt; *ESE 251 -&gt; *ESE? &lt;- 251</pre>								

## 5.5 \*ESE?

<b>Syntax</b>	<b>*ESE?</b>
<b>Description</b>	Queries the current setting of the standard event status enable register
<b>Parameter</b>	None
<b>Explanation</b>	The query returns the content of the standard event status enable register
<b>Return Format</b>	<NR1>
<b>Example</b>	<pre>-&gt; *ESE 251 -&gt; *ESE? &lt;- 251</pre>

## 5.6 \*ESR?

<b>Syntax</b>	<b>*ESR?</b>
<b>Description</b>	Queries the standard event status register and clears the register
<b>Parameter</b>	None
<b>Explanation</b>	<ul style="list-style-type: none"> <li>• A sum of decimal values of each bit is returned</li> <li>• You can check what type of events occurred when an SRQ is generated</li> <li>• For example, if a value of “32” is returned, this indicates that the standard event register is set to “00100000.” In this case, you can see that the SRQ occurred due to a “command syntax error.”</li> <li>• A query using <code>ESR?</code> will clear the contents of the standard event register</li> </ul>
<b>Return Format</b>	<NR1>
<b>Example</b>	<pre>-&gt; *ESR? &lt;- 32</pre>



## 5.7 \*OPC

<b>Syntax</b>	<b>*OPC</b>
<b>Description</b>	Sets bit 0 (OPC bit) of the standard event register to 1 upon the completion of the specified overlap command
<b>Parameter</b>	None
<b>Explanation</b>	Currently all operations are non-overlapped. *OPC sets the OPC bit and *OPC? returns the state of the OPC bit
<b>Example</b>	<pre>-&gt; *OPC -&gt; *OPC? &lt;- 1</pre>

## 5.8 \*OPC?

<b>Syntax</b>	<b>*OPC?</b>
<b>Description</b>	Queries the state of specified overlapped command
<b>Parameter</b>	None
<b>Explanation</b>	Currently all operations are non-overlapped. *OPC sets the OPC bit
<b>Return Format</b>	<Boolean>
<b>Example</b>	<pre>-&gt; *OPC -&gt; *OPC? &lt;- 1</pre>

## 5.9 \*RST

<b>Syntax</b>	<b>*RST</b>
<b>Description</b>	Initializes the device to default settings
<b>Parameter</b>	None
<b>Explanation</b>	<ul style="list-style-type: none"> <li>• Ejects all open data files and switches to live mode</li> <li>• Stops the recording</li> <li>• Restarts the acquisition</li> <li>• Clears the event queue</li> <li>• Resets all settings except communication settings to factory default values</li> </ul>
<b>Example</b>	<pre>-&gt; *RST</pre>

## 5.10 \*SRE

<b>Syntax</b>	<b>*SRE &lt;num&gt;</b>			
<b>Description</b>	Sets the service request enable register			
<b>Parameter</b>				
	<b>Name</b>	<b>Type</b>	<b>Range</b>	<b>Default</b>
	<num>	Integer	0 - 255	0
<b>Explanation</b>	<ul style="list-style-type: none"> <li>Specify the value as a sum of decimal values of each bit</li> <li>For example, specifying *SRE 239 will cause the <i>service request enable</i> register to be set to "11101111." In this case, bit 4 of the <i>service request enable</i> register is disabled which means that bit 4 (MAV) of the status byte register is not set to 1, even if "the output queue is not empty".</li> <li>Bit 6 (MSS) of the status byte register is the MSS bit itself, and therefore, is ignored</li> <li>A query using *SRE? will not clear the contents of the service request enable register</li> </ul>			
<b>Example</b>	<pre>-&gt; *SRE 239 -&gt; *SRE? &lt;- 175</pre>			

## 5.11 \*SRE?

<b>Syntax</b>	<b>*SRE?</b>
<b>Description</b>	Queries the current setting of the service request enable register
<b>Parameter</b>	None
<b>Explanation</b>	The query returns the content of the service request enable register
<b>Return Format</b>	<NR1>
<b>Example</b>	<pre>-&gt; *SRE 239 -&gt; *SRE? &lt;- 175</pre>

## 5.12 \*STB?

<b>Syntax</b>	<b>*STB?</b>
<b>Description</b>	Queries the status byte register
<b>Parameter</b>	None
<b>Explanation</b>	<p><b>The query returns the content of the status byte register.</b></p> <ul style="list-style-type: none"> <li>• Bits 0, 1, 3, 4 and 7 Not used (always 0)</li> <li>• Bit 2 EAV (Error Available) Set to 1 when the error queue is not empty. In other words, this bit is set to 1 when an error occurs. See the page 7-9.</li> <li>• Bit 5 ESB (Event Summary Bit) Set to 0 when the logical product of the standard event register and the corresponding enable register is 1. In other words, this bit is set to 1, when an event takes place inside the instrument.</li> <li>• Bit 6 RQS (Request Service)/MSS (Master Status Summary) Set to 1 when the logical AND of the status byte excluding Bit 6 and the service request enable register is not 0. In other words, this bit is set to 1 when the instrument is requesting service from the controller. RQS is set to 1 when the MSS bit changes from 0 to 1, and cleared when serial polling is carried out or when the MSS bit changes to 0.</li> </ul>
<b>Return Format</b>	<NR1>
<b>Example</b>	<pre>-&gt; *STB? &lt;- 4</pre>

## 5.13 \*TST?

<b>Syntax</b>	<b>*TST?</b>
<b>Description</b>	Performs a self-test and queries the result
<b>Parameter</b>	None
<b>Explanation</b>	Currently a self-test is not defined and this query is no-op.
<b>Return Format</b>	<NR1>
<b>Example</b>	<pre>-&gt; *TST? &lt;- 0</pre>

## 5.14 \*WAI

<b>Syntax</b>	<b>*WAI</b>
<b>Description</b>	Holds the subsequent command until the completion of the asynchronous operation
<b>Parameter</b>	None
<b>Explanation</b>	Currently all operations are synchronous operations
<b>Example</b>	-> *WAI



## APPLICATION CONTROL

### 6.1 Setup

#### 6.1.1 :SETup:LOAD

<b>Syntax</b>	<b>:SETup:LOAD &lt;file_name&gt; &lt;path&gt;</b>															
<b>Description</b>	Load specified setup file of the current configuration directory or an absolute path															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;file_name&gt;</td> <td>String</td> <td>The filename of the setup file in the default directory (data folder)</td> <td>None</td> </tr> <tr> <td>&lt;path&gt;</td> <td>String</td> <td>The absolute path of the setup file to load</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<file_name>	String	The filename of the setup file in the default directory (data folder)	None	<path>	String	The absolute path of the setup file to load	None
Name	Type	Range	Default													
<file_name>	String	The filename of the setup file in the default directory (data folder)	None													
<path>	String	The absolute path of the setup file to load	None													
<b>Explanation</b>	This command loads a setup (measurement configuration) file from the current configuration location or an absolute path and applies it directly. The <file_name> does not need to include the file extension ".dms".															
<b>Example</b>	<pre>-&gt; :SETup:LOAD "setup1.dms" &lt;- :SETup:LOAD "C:/DATA/setup1.dms"</pre>															

### 6.1.2 :SETup:APPLY “XML-String”

<b>Syntax</b>	:SETup:APPLY #<num_int><num_char><xml_setup>			
<b>Description</b>	Upload a measurement setup to device and apply it			
<b>Parameter</b>	<b>Name</b>	<b>Type</b>	<b>Range</b>	<b>Default</b>
	<num_int>	NR1	Number of Integers following after this Integer 1 to 9	None
	<num_char>	NR1	Number of Characters to be transferred 1 to 999999999	None
	<xml_setup>	Arbitrary Data	Setup data (XML format)	None
<b>Explanation</b>	This command uploads a measurement setup to the device and applies it after transfer complete. The measurement setup is a XML-String which can be downloaded via :SETUP:READ?.			
<b>Example</b>	-> :SETup:APPLY #41234...			
<b>Related Commands</b>	:SETup:READ?			

### 6.1.3 :SETup:SAVE "Path"

<b>Syntax</b>	<b>:SETup:SAVE &lt;file_name&gt; &lt;path&gt;</b>															
<b>Description</b>	Saves to current setup to the specified file in the current configuration directory or to an absolute path															
<b>Parameter</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Name</th> <th style="width: 25%;">Type</th> <th style="width: 30%;">Range</th> <th style="width: 20%;">Default</th> </tr> </thead> <tbody> <tr> <td>&lt;file_name&gt;</td> <td>String</td> <td>The filename of the setup file in the default directory (data folder)</td> <td>None</td> </tr> <tr> <td>&lt;path&gt;</td> <td>String</td> <td>The absolute path of the setup file</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<file_name>	String	The filename of the setup file in the default directory (data folder)	None	<path>	String	The absolute path of the setup file	None
Name	Type	Range	Default													
<file_name>	String	The filename of the setup file in the default directory (data folder)	None													
<path>	String	The absolute path of the setup file	None													
<b>Explanation</b>	This command loads a setup (measurement configuration) file from the current configuration location or an absolute path and applies it directly. The <file_name> does not need to include the file extension ".dms".															
<b>Example</b>	<pre>-&gt; :SETup:SAVE "setup1.dms" &lt;- :SETup:SAVE "C:/DATA/setup1.dms"</pre>															



### 6.1.4 :SETup:READ?

<b>Syntax</b>	<b>:SETup:READ? [&lt;path&gt;]</b>											
<b>Description</b>	Download the current measurement setup or, if a path is specified, the measurement setup referenced to path from the device											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;path&gt;</td> <td>String</td> <td>The absolute path of the setup file to read (XML)</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<path>	String	The absolute path of the setup file to read (XML)	None
	Name	Type	Range	Default								
	<path>	String	The absolute path of the setup file to read (XML)	None								
<b>Explanation</b>	Download the current measurement setup or, if a path is specified, the measurement setup referenced to path from the device.											
<b>Return Format</b>	<Arbitrary Data>											
<b>Example</b>	<pre>-&gt; :SETup:READ? &lt;- #43432..... ... -&gt; :SETup:READ? "C:/Data/test.dms" &lt;- #41242.....</pre>											
<b>Related Commands</b>	<pre>:SETup:LOAD :SETup:APPLY</pre>											

### 6.1.5 :SETup:NAME?

<b>Syntax</b>	<b>:SETup:NAME?</b>
<b>Description</b>	Get name of the current loaded OXYGEN setup
<b>Parameter</b>	None
<b>Explanation</b>	This query returns the filename of the current loaded setup file
<b>Return Format</b>	<String>   NONE
<b>Example</b>	<pre>-&gt; :SETup:NAME? &lt;- :SET:NAM "D:/DATA/last.dms"</pre>

6.1.6 :SETup:ASync:LOAD "Path"

<b>Syntax</b>	:SETup:ASync:LOAD <file_name> <path>															
<b>Description</b>	Starts loading a specified setup file of the current configuration directory or an absolute path															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;file_name&gt;</td> <td>String</td> <td>The filename of the setup file in the default directory (data folder)</td> <td>None</td> </tr> <tr> <td>&lt;path&gt;</td> <td>String</td> <td>The absolute path of the setup file to load</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<file_name>	String	The filename of the setup file in the default directory (data folder)	None	<path>	String	The absolute path of the setup file to load	None
Name	Type	Range	Default													
<file_name>	String	The filename of the setup file in the default directory (data folder)	None													
<path>	String	The absolute path of the setup file to load	None													
<b>Explanation</b>	<p>This command asynchronously loads a setup (measurement configuration) file from the current configuration location or an absolute path and applies it. The &lt;file_name&gt; does not need to include the file extension ".dms".</p> <p><b>Since this command only starts loading, other device commands or queries could be invalid or causes problems, when the loading is not finished yet.</b></p> <p>Use the query :SETup:ASync:STATE? (see next command) to acquire the state of the loading operation.</p>															
<b>Example</b>	<pre>-&gt; :SETup:ASync:LOAD "setup1.dms" &lt;- :SETup:ASync:LOAD "C:DATAsetup1.dms"</pre>															

6.1.7 :SETup:ASync:STATe?

<b>Syntax</b>	<b>:SETup:ASync:STATe?</b>
<b>Description</b>	Queries the state of the last async load configuration operation
<b>Parameter</b>	None
<b>Explanation</b>	This command queries the state of the last async config load operation. The state could be <ul style="list-style-type: none"> <li>• IDLE</li> <li>• LOADing</li> </ul> Some execution errors are added, if the load operation is not finished correctly.
<b>Return Format</b>	<String>
<b>Example</b>	<pre> -&gt; :SETup:ASync:STATe? &lt;- IDLE -&gt; :SETup:ASync:LOAD "C:DATAsetup1.dms" -&gt; :SETup:ASync:STATe? &lt;- LOAD -&gt; :SETup:ASync:STATe? &lt;- LOAD ... -&gt; :SETup:ASync:STATe? &lt;- IDLE ... -&gt; :SETup:ASync:LOAD "C:DATAUnknown.dms" -&gt; :SETup:ASync:STATe? &lt;- IDLE -&gt; :SYST:ERR:ALL? &lt;- -256, "File name not found" </pre>

## 6.2 UI Control

### 6.2.1 :SYSTEM:DATE?

<b>Syntax</b>	<b>:SYSTEM:DATE?</b>
<b>Description</b>	Queries the system date
<b>Parameter</b>	None
<b>Explanation</b>	Returns the system date in the form <year>,<month>,<day> <ul style="list-style-type: none"> <li>• &lt;year&gt;: INTEGER, four digit number</li> <li>• &lt;month&gt;: INTEGER with the range 1 to 12</li> <li>• &lt;day&gt;: INTEGER with the range 1 to 31</li> </ul>
<b>Return Format</b>	<NR1>,<NR1>,<NR1>
<b>Example</b>	<pre>-&gt; :SYSTEM:DATE? &lt;- 2017,07,25</pre>
<b>Related Commands</b>	<pre>:SYSTEM:TIME? :SYSTEM:TZONE?</pre>

### 6.2.2 :SYSTEM:KLOCK {ON|OFF}

<b>Syntax</b>	<b>:SYSTEM:KLOCK {ON 1 OFF 0}</b>								
<b>Description</b>	Locks/Unlocks the Screen								
<b>Parameter</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Name</th> <th style="width: 25%;">Type</th> <th style="width: 25%;">Range</th> <th style="width: 25%;">Default</th> </tr> </thead> <tbody> <tr> <td>{ON 1 OFF 0}</td> <td>Boolean</td> <td>ON 1 OFF 0</td> <td>None</td> </tr> </tbody> </table>	Name	Type	Range	Default	{ON 1 OFF 0}	Boolean	ON 1 OFF 0	None
Name	Type	Range	Default						
{ON 1 OFF 0}	Boolean	ON 1 OFF 0	None						
<b>Explanation</b>	Locks/Unlocks the Screen of Oxygen Software								
<b>Example</b>	<pre>-&gt; :SYSTEM:KLOCK ON -&gt; :SYSTEM:KLOCK OFF</pre>								
<b>Related Commands</b>	<pre>-&gt; :SYSTEM:KLOCK?</pre>								

### 6.2.3 :SYSTem:KLOCK?

<b>Syntax</b>	<b>:SYSTem:KLOCK?</b>
<b>Description</b>	Queries the Screen Lock status
<b>Parameter</b>	None
<b>Explanation</b>	Queries the Screen Lock status
<b>Return Format</b>	<Boolean>
<b>Example</b>	<pre>-&gt; :SYSTem:KLOCK ON -&gt; :SYSTem:KLOCK? &lt;- 1</pre>
<b>Related Commands</b>	-> :SYSTem:KLOCK

### 6.2.4 :SYSTem:TIME?

<b>Syntax</b>	<b>:SYSTem:TIME?</b>
<b>Description</b>	Queries the system time
<b>Parameter</b>	None
<b>Explanation</b>	<p>Returns the local system time in the form &lt;hour&gt;,&lt;minute&gt;,&lt;second&gt;</p> <ul style="list-style-type: none"> <li>• &lt;hour&gt;: INTEGER with the range 0 to +23</li> <li>• &lt;minute&gt;: INTEGER with the range 0 to 59</li> <li>• &lt;second&gt;: INTEGER with the range 0 to 59.99999...</li> </ul> <p>If Acquisition is not running, :SYSTEM:TIME returns the last known Time</p>
<b>Return Format</b>	<Boolean>
<b>Example</b>	<pre>-&gt; :SYSTem:TIME? &lt;- 17,31,1.0237</pre>
<b>Related Commands</b>	<pre>:SYSTem:DATE? :SYSTem:TZONE?</pre>

### 6.2.5 :SYSTem:TZONE?

<b>Syntax</b>	<b>:SYSTem:TZONE?</b>
<b>Description</b>	Queries the system time zone
<b>Parameter</b>	None
<b>Explanation</b>	<p>Returns the system time zone offset in the form &lt;hour&gt;,&lt;minute&gt;</p> <ul style="list-style-type: none"> <li>• &lt;hour&gt;: INTEGER with the range 0 to +23</li> <li>• &lt;minute&gt;: INTEGER with the range 0 to 59</li> </ul> <p>When each field is subtracted from the value of the TIME command, the result is the correct universal coordinated time (also known as UCT, Zulu, Greenwich Mean Time).  </p>
<b>Return Format</b>	<NR1>,<NR1>
<b>Example</b>	<pre>-&gt; :SYSTem:TZONE? &lt;- 2,0</pre>
<b>Related Commands</b>	<pre>:SYSTem:DATE? :SYSTem:TIME?</pre>

### 6.2.6 :COMMunicate:HEADer {ON|OFF}

<b>Syntax</b>	<b>:COMMunicate:HEADer {ON 1 OFF 0}</b>								
<b>Description</b>	Sets whether query responses should generate an output for the header.								
<b>Parameter</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Name</th> <th style="width: 25%;">Type</th> <th style="width: 25%;">Range</th> <th style="width: 25%;">Default</th> </tr> </thead> <tbody> <tr> <td>{ON 1 OFF 0}</td> <td>Boolean</td> <td>ON 1 OFF 0</td> <td>None</td> </tr> </tbody> </table>	Name	Type	Range	Default	{ON 1 OFF 0}	Boolean	ON 1 OFF 0	None
Name	Type	Range	Default						
{ON 1 OFF 0}	Boolean	ON 1 OFF 0	None						
<b>Explanation</b>	Sets whether query responses include the header.								
<b>Example</b>	<pre>-&gt; :COMMunicate:HEADer ON -&gt; :SYSTem:TIME? &lt;- :SYSTEM:TIME 15,14,15.7644</pre>								
<b>Related Commands</b>	<pre>:COMMunicate:HEADer? :COMMUNICATE:VERBOSE</pre>								

### 6.2.7 :COMMunicate:HEADer?

<b>Syntax</b>	<b>:COMMunicate:HEADer?</b>
<b>Description</b>	Queries the communication response header setting
<b>Parameter</b>	None
<b>Explanation</b>	Queries the setting whether the response includes the header or not
<b>Return Format</b>	<Boolean>
<b>Example</b>	<pre>-&gt; :COMMunicate:HEADer? &lt;- 1</pre>
<b>Related Com- mands</b>	:COMMunicate:HEADer

### 6.2.8 :COMMunicate:VERBose {ON|OFF}

<b>Syntax</b>	<b>:COMMunicate:VERBose {ON 1 OFF 0}</b>											
<b>Description</b>	Sets whether query responses should generate the long or short form of the header											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>{ON 1 OFF 0}</td> <td>Boolean</td> <td>ON 1 OFF 0</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	{ON 1 OFF 0}	Boolean	ON 1 OFF 0	None
Name	Type	Range	Default									
{ON 1 OFF 0}	Boolean	ON 1 OFF 0	None									
<b>Explanation</b>	Sets whether query responses should generate the long or short form of the header											
<b>Example</b>	<pre>-&gt; :COMMunicate:HEADer ON -&gt; :COMMunicate:VERBose OFF -&gt; :SYSTem:TIME? &lt;- :SYST:TIME 15,14,15.7644 -&gt; :COMMunicate:VERBose ON -&gt; :SYSTem:TIME? &lt;- :SYSTEM:TIME 15,14,32.7265</pre>											
<b>Related Com- mands</b>	:COMMunicate:HEADer											

## 6.2.9 :COMMunicate:VERBose?

<b>Syntax</b>	<b>:COMMunicate:VERBose?</b>
<b>Description</b>	Queries the setting, whether the response header is short or long form
<b>Parameter</b>	None
<b>Explanation</b>	Queries the setting, whether the response header is short or long form
<b>Return Format</b>	<Boolean>
<b>Example</b>	-> :COMMunicate:VERBose? <- 1
<b>Related Com- mands</b>	:COMMunicate:VERBose





## ACQUISITION CONTROL

### 7.1 :ACQUisition:START

<b>Syntax</b>	<b>:ACQUisition:START</b>
<b>Description</b>	Starts the OXYGEN acquisition
<b>Parameter</b>	None
<b>Explanation</b>	Start the acquisition if not already running
<b>Example</b>	-> :ACQUisition:START

### 7.2 :ACQUisition:STOP

<b>Syntax</b>	<b>:ACQUisition:STOP</b>
<b>Description</b>	Stops the acquisition
<b>Parameter</b>	None
<b>Explanation</b>	Stops the acquisition if not already stopped
<b>Example</b>	-> :ACQUisition:STOP

### 7.3 :ACQUisition:RESTART

<b>Syntax</b>	<b>:ACQUisition:RESTART</b>
<b>Description</b>	Restarts the acquisition
<b>Parameter</b>	None
<b>Explanation</b>	Stops, then starts the acquisition. If the acquisition was stopped or paused before, it will only be started.
<b>Example</b>	-> :ACQUisition:RESTART

## 7.4 :ACQUisition:STATe?

<b>Syntax</b>	<b>:ACQUisition:STATe?</b>
<b>Description</b>	Queries the current acquisition state
<b>Parameter</b>	None
<b>Explanation</b>	<b>Returns the acquisition state as string:</b> <ul style="list-style-type: none"><li>• Stopped</li><li>• Waiting_for_sync</li><li>• Started</li></ul>
<b>Return format</b>	Literal
<b>Example</b>	<pre>-&gt; :ACQUisition:START -&gt; ACQUisition:STATe? &lt;- Started</pre>

## RECORDING CONTROL

### 8.1 :STORe:FILE:NAME "PATH"

<b>Syntax</b>	(1) :STORe:FILE:NAME <filename> (2) :STORe:FILE:NAME <path>															
<b>Description</b>	Set file name or absolute path of the recording file															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;file_name&gt;</td> <td>String</td> <td>The filename of the setup file in the default directory (data folder)</td> <td>None</td> </tr> <tr> <td>&lt;path&gt;</td> <td>String</td> <td>The absolute path of the setup file</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<file_name>	String	The filename of the setup file in the default directory (data folder)	None	<path>	String	The absolute path of the setup file	None
Name	Type	Range	Default													
<file_name>	String	The filename of the setup file in the default directory (data folder)	None													
<path>	String	The absolute path of the setup file	None													
<b>Explanation</b>	Set File Name or Path of the Recording File. The File will be overwritten on storing start, if the file already exists. If the file extension is not provided (.dmd), it will be appended automatically.															
<b>Example</b>	<pre>-&gt; :STORe:FILE:NAME "Testrun_1" -&gt; :STORe:FILE:NAME "D:/DATA/Testrun_1"</pre>															

## 8.2 :STORe:FILE:NAME?

<b>Syntax</b>	<b>:STORe:FILE:NAME?</b>
<b>Description</b>	Get File Name and Path of the Recording File during recording
<b>Parameter</b>	None
<b>Explanation</b>	Get the File Name and Path of the Recording File while recording is started. If recording is currently stopped, NONE is returned.
<b>Return Format</b>	<String> NONE
<b>Example</b>	<pre>-&gt; :STORe:FILE:NAME? &lt;- :STOR:FILE "D:/DATA/Testrun_1.dmd" -&gt; :STORe:FILE:NAME? &lt;- :STOR:FILE NONE</pre>

## 8.3 :STORe:START

<b>Syntax</b>	<b>:STORe:START</b>
<b>Description</b>	Start storing operation
<b>Parameter</b>	None
<b>Explanation</b>	<p>Starts the storing operation or arms configured triggers. If the triggers are already armed forces storing start as if a store trigger were encountered. The file will be overwritten if it exists.</p> <p>If the storing operation was paused, the storing operation will be resumed in the same file.</p>
<b>Example</b>	

## 8.4 :STORe:PAUSE

<b>Syntax</b>	<b>:STORe:PAUSE</b>
<b>Description</b>	Pause storing operation
<b>Parameter</b>	None
<b>Explanation</b>	Pauses the storing operation. Pausing in triggered recording is not supported and the PAUSE command will return an error if triggered recording is configured.
<b>Example</b>	

## 8.5 :STORe:STOP

<b>Syntax</b>	<b>:STORe:STOP</b>
<b>Description</b>	Stop storing operation
<b>Parameter</b>	None
<b>Explanation</b>	Stops the storing operation or disarms the triggers.
<b>Example</b>	

## 8.6 :STORe:STATe?

<b>Syntax</b>	<b>:STORe:STATe?</b>
<b>Description</b>	Queries the state of the recording operation {Started Stopped Stopping Paused Armed}
<b>Parameter</b>	None
<b>Explanation</b>	Returns the recording state: <ul style="list-style-type: none"> <li>• Stopped (Storing can be started now)</li> <li>• Stopping (Waiting for a previous store to stop)</li> <li>• Paused (A current storing operation is pause)</li> <li>• Started (Storing is started)</li> <li>• Armed (Triggers for storing are armed)  </li> </ul>
<b>Return Format</b>	Literal
<b>Example</b>	<pre>-&gt; :STORe:START -&gt; :STORe:STATe? &lt;- Started</pre>

## 8.7 WAVEform access

### 8.8 :STORe:WAVEform:MODE?

<b>Syntax</b>	<b>:STORe:WAVEform:MODE?</b>
<b>Description</b>	Queries the current recording waveform mode {Continuous Eventbased Disabled}
<b>Parameter</b>	None
<b>Explanation</b>	Returns the recording waveform mode: <ul style="list-style-type: none"> <li>• Continuous (Continuous recording can be started now)</li> <li>• EventBased (Eventbased triggering system is active)</li> <li>• Disabled (System is disabled)</li> </ul> This is the default query in the waveform group
<b>Return Format</b>	Literal
<b>Example</b>	<pre>-&gt; :STORe:WAVEform:MODE? &lt;- Eventbased ... -&gt; :STORe:WAVEform?           //default &lt;- Eventbased</pre>

### 8.9 :STORe:WAVEform:MODE {Continuous | Eventbased | Disabled}

<b>Syntax</b>	<b>STORe:WAVEform:MODE {Continuous Eventbased Disabled}</b>								
<b>Description</b>	Set the current waveform mode								
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;Waveform Mode&gt;</td> <td>Literal</td> <td>{Continuous Event-based Disabled}</td> <td>None</td> </tr> </tbody> </table>	Name	Type	Range	Default	<Waveform Mode>	Literal	{Continuous Event-based Disabled}	None
Name	Type	Range	Default						
<Waveform Mode>	Literal	{Continuous Event-based Disabled}	None						
<b>Explanation</b>	Sets the current waveform mode. This is the default command in the waveform group								
<b>Example</b>	<pre>-&gt; :STORe:WAVEform:MODE Eventbased ... -&gt; :STORe:WAVEform Eventbased           //default</pre>								

## 8.10 :STORe:WAVEform:CONTInuous

<b>Syntax</b>	<b>:STORe:WAVEform:CONTInuous</b>
<b>Description</b>	Sets the waveform mode to continuous
<b>Parameter</b>	None
<b>Explanation</b>	
<b>Example</b>	-> :STORe:WAVEform:CONTInuous

## 8.11 :STORe:WAVEform:EVENTbased

<b>Syntax</b>	<b>:STORe:WAVEform:EVENTbased</b>
<b>Description</b>	Sets the waveform mode to eventbased
<b>Parameter</b>	None
<b>Explanation</b>	Enables the oxy trigger event system
<b>Example</b>	-> :STORe:WAVEform:EVENTbased

## 8.12 STORe:WAVEform:DISabled

<b>Syntax</b>	<b>:STORe:WAVEform:DISabled</b>
<b>Description</b>	Sets the waveform mode to disabled
<b>Parameter</b>	None
<b>Explanation</b>	Disables the oxy trigger event system
<b>Example</b>	-> :STORe:WAVEform:DISabled



### 8.13 :STORe:WAVEform:PREtime?

<b>Syntax</b>	<b>:STORe:WAVEform:PREtime?</b>
<b>Description</b>	Queries the current waveform pretime values
<b>Parameter</b>	None
<b>Explanation</b>	Shows the current state (ON OFF) and the current value of the waveform pretime
<b>Return Format</b>	<Boolean>,<Nrf>
<b>Example</b>	<pre>-&gt; :STORe:WAVEform:PREtime? &lt;-  OFF,1.0</pre>

### 8.14 :STORe:WAVEform:PREtime {ON|OFF|<Nrf>} | {{ON|OFF},<Nrf>}

<b>Syntax</b>	<b>:STORe:WAVEform:PREtime {ON OFF &lt;Nrf&gt;}   {{ON OFF},&lt;Nrf&gt;}</b>															
<b>Description</b>	Sets the waveform pretime															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>Pretime state or value</td> <td>Literal   Nrf</td> <td>ON   OFF   1–max seconds</td> <td>None</td> </tr> <tr> <td>Pretime value</td> <td>Nrf   None</td> <td>1–max seconds</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	Pretime state or value	Literal   Nrf	ON   OFF   1–max seconds	None	Pretime value	Nrf   None	1–max seconds	None
Name	Type	Range	Default													
Pretime state or value	Literal   Nrf	ON   OFF   1–max seconds	None													
Pretime value	Nrf   None	1–max seconds	None													
<b>Explanation</b>	Sets the current state and/or value of the waveform pretime. The state should be ON or OFF. The value is numeric in seconds. This command accepts one or two parameters (see examples)															
<b>Example</b>	<pre>-&gt; :STORe:WAVEform:PREtime OFF -&gt; :STORe:WAVEform:PREtime? &lt;-  OFF,0.0 -&gt; :STORe:WAVEform:PREtime 1.0           //sets the -&gt; pretime enabled and to 1sec -&gt; :STORe:WAVEform:PREtime? &lt;-  ON,1.0 -&gt; :STORe:WAVEform:PREtime OFF,0.5 -&gt; :STORe:WAVEform:PREtime? &lt;-  OFF,5.0E-1</pre>															

### 8.15 :STORe:WAVEform:PAFTer?

<b>Syntax</b>	<b>:STORe:WAVEform:PAFTer?</b>
<b>Description</b>	Queries the current pause after values
<b>Parameter</b>	None
<b>Explanation</b>	Shows the current state (ON OFF) and the current value of the waveform pause after
<b>Return Format</b>	<Boolean>,<Nrf>
<b>Example</b>	<pre>-&gt; :STORe:WAVEform:PAFTer? &lt;-  OFF,1.0</pre>

### 8.16 :STORe:WAVE:PAFTer {ON|OFF|<Nrf>} | {{ON|OFF},<Nrf>}

<b>Syntax</b>	<b>:STORe:WAVEform:PAFTer {ON OFF &lt;Nrf&gt;}   {{ON OFF},&lt;Nrf&gt;}</b>															
<b>Description</b>	Sets the waveform pause after															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>Pause after state or value</td> <td>Literal   Nrf</td> <td>ON   OFF   1–max seconds</td> <td>None</td> </tr> <tr> <td>Pause after value</td> <td>Nrf   None</td> <td>1–max seconds</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	Pause after state or value	Literal   Nrf	ON   OFF   1–max seconds	None	Pause after value	Nrf   None	1–max seconds	None
Name	Type	Range	Default													
Pause after state or value	Literal   Nrf	ON   OFF   1–max seconds	None													
Pause after value	Nrf   None	1–max seconds	None													
<b>Explanation</b>	Sets the current state and/or value of the waveform pause after. The state should be ON or OFF. The value is numeric in seconds. This command accepts one or two parameters (see examples)															
<b>Example</b>	<pre>-&gt; :STORe:WAVEform:PAFTer OFF -&gt; :STORe:WAVEform:PAFTer? &lt;-  OFF,0.0 -&gt; :STORe:WAVEform:PAFTer 1.0           //sets the -&gt; :STORe:WAVEform:PAFTer? &lt;-  ON,1.0 -&gt; :STORe:WAVEform:PAFTer OFF,0.5 -&gt; :STORe:WAVEform:PAFTer? &lt;-  OFF,5.0E-1</pre>															

### 8.17 :STORe:WAVEform:POSTtime?

<b>Syntax</b>	<b>:STORe:WAVEform:POSTtime?</b>
<b>Description</b>	Queries the current waveform posttime values
<b>Parameter</b>	None
<b>Explanation</b>	Shows the current state (ON OFF) and the current value of the waveform posttime
<b>Return Format</b>	<Boolean>,<Nrf>
<b>Example</b>	<pre>-&gt; :STORe:WAVEform:POSTtime? &lt;-  OFF,1.0</pre>

### 8.18 :STORe:WAVEform:POSTtime {ON|OFF|<Nrf>} | {{ON|OFF},<Nrf>}

<b>Syntax</b>	<b>:STORe:WAVEform:POSTtime {ON OFF &lt;Nrf&gt;}   {{ON OFF},&lt;Nrf&gt;}</b>															
<b>Description</b>	Sets the waveform posttime															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>Posttime state or value</td> <td>Literal   Nrf</td> <td>ON   OFF   1–max seconds</td> <td>None</td> </tr> <tr> <td>Posttime value</td> <td>Nrf   None</td> <td>1–max seconds</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	Posttime state or value	Literal   Nrf	ON   OFF   1–max seconds	None	Posttime value	Nrf   None	1–max seconds	None
Name	Type	Range	Default													
Posttime state or value	Literal   Nrf	ON   OFF   1–max seconds	None													
Posttime value	Nrf   None	1–max seconds	None													
<b>Explanation</b>	Sets the current state and/or value of the waveform posttime. The state should be ON or OFF. The value is numeric in seconds. This command accepts one or two parameters (see examples)															
<b>Example</b>	<pre>-&gt; :STORe:WAVEform:POSTtime OFF -&gt; :STORe:WAVEform:POSTtime? &lt;-  OFF,0.0 -&gt; :STORe:WAVEform:POSTtime 1.0           //sets the -&gt;  pretime enabled and to 1sec -&gt; :STORe:WAVEform:POSTtime? &lt;-  ON,1.0 -&gt; :STORe:WAVEform:POSTtime OFF,0.6 -&gt; :STORe:WAVEform:POSTtime? &lt;-  OFF,6.0E-1</pre>															

## 8.19 :STORe:STATIstics?

<b>Syntax</b>	<b>:STORe:STATIstics?</b>
<b>Description</b>	Queries the current recording setup statistics values
<b>Parameter</b>	None
<b>Explanation</b>	Shows the current state (ON OFF) and the current value of the recording statistics
<b>Return Format</b>	<Boolean>,<Nrf>
<b>Example</b>	<pre>-&gt; :STORe:STATIstics? &lt;-  OFF,1.0</pre>

## 8.20 :STORe:STATIstics {ON|OFF|<Nrf>} | {{ON|OFF},<Nrf>}

<b>Syntax</b>	<b>:STORe:STATIstics {ON OFF &lt;Nrf&gt;}   {{ON OFF},&lt;Nrf&gt;}</b>												
<b>Description</b>	Sets the waveform posttime												
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>Statistics state or value</td> <td>Literal   Nrf</td> <td>ON   OFF   1–max seconds</td> <td>None</td> </tr> <tr> <td>Statistics value</td> <td>Nrf   None</td> <td>1–max seconds</td> <td>None</td> </tr> </tbody> </table>	Name	Type	Range	Default	Statistics state or value	Literal   Nrf	ON   OFF   1–max seconds	None	Statistics value	Nrf   None	1–max seconds	None
Name	Type	Range	Default										
Statistics state or value	Literal   Nrf	ON   OFF   1–max seconds	None										
Statistics value	Nrf   None	1–max seconds	None										
<b>Explanation</b>	Sets the current state and/or value of the recording setup statistics. The state should be ON or OFF. The value is numeric in seconds. This command accepts one or two parameters (see examples)												
<b>Example</b>	<pre>-&gt; :STORe:STATIstics OFF -&gt; :STORe:STATIstics? &lt;-  OFF,1.0  -&gt; :STORe:STATIstics 1.1           //enables statistcs↵ ↵state -&gt; :STORe:STATIstics? &lt;-  ON,1.1  -&gt; :STORe:STATIstics OFF,0.0 -&gt; :STORe:STATIstics? &lt;-  OFF,0.0</pre>												

## 8.21 :STORe:AUTOStart?

<b>Syntax</b>	<b>:STORe:AUTOStart?</b>
<b>Description</b>	Queries the current start measurement automatically state
<b>Parameter</b>	None
<b>Explanation</b>	Shows the current state (ON OFF) of start measurement automatically
<b>Return Format</b>	<Boolean>
<b>Example</b>	<pre>-&gt; :STORe:AUTOStart? &lt;-  OFF</pre>

## 8.22 :STORe:AUTOStart <Boolean>

<b>Syntax</b>	<b>:STORe:AUTOStart {ON 1 OFF 0}</b>											
<b>Description</b>	Sets the start measurement automatically state											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>{ON 1 OFF 0}</td> <td>Boolean</td> <td>ON 1 OFF 0</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	{ON 1 OFF 0}	Boolean	ON 1 OFF 0	None
Name	Type	Range	Default									
{ON 1 OFF 0}	Boolean	ON 1 OFF 0	None									
<b>Explanation</b>	Sets the start measurement automatically state to enabled or disabled											
<b>Example</b>	<pre>-&gt; :STORe:AUTOStart ON -&gt; :STORe:AUTOStart? &lt;-  ON</pre>											

## 8.23 :STORe:RACquisition?

<b>Syntax</b>	<b>:STORe:RACquisition?</b>
<b>Description</b>	Queries the current restart acquisition on measurement start state
<b>Parameter</b>	None
<b>Explanation</b>	Shows the current state (ON OFF) of restart acquisition on measurement start
<b>Return Format</b>	<Boolean>
<b>Example</b>	<pre>-&gt; :STORe:RACquisition? &lt;-  OFF</pre>

## 8.24 :STORe:RACQquisition <Boolean>

<b>Syntax</b>	<b>:STORe:RACQquisition {ON 1 OFF 0}</b>											
<b>Description</b>	Sets the restart acquisition on measurement start flag											
<b>Parameter</b>	<table border="1" style="width: 100%;"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>{ON 1 OFF 0}</td> <td>Boolean</td> <td>ON 1 OFF 0</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	{ON 1 OFF 0}	Boolean	ON 1 OFF 0	None
	Name	Type	Range	Default								
	{ON 1 OFF 0}	Boolean	ON 1 OFF 0	None								
<b>Explanation</b>	Sets the restart acquisition on measurement start to enabled or disabled											
<b>Example</b>	<pre>-&gt; :STORe:RACQquisition ON -&gt; :STORe:RACQquisition? &lt;- ON</pre>											

## 8.25 :STORe:SAFTer?

<b>Syntax</b>	<b>:STORe:WAVEform:SAFTer?</b>
<b>Description</b>	Queries the current stop measurement after values
<b>Parameter</b>	None
<b>Explanation</b>	Shows the current state (ON OFF) and the stop measurement after value
<b>Return Format</b>	<Boolean>,<Nrf>
<b>Example</b>	<pre>-&gt; :STORe:SAFTer? &lt;- OFF,1.0</pre>

## 8.26 :STORe:SAFTer {ON|OFF|<Nrf>} | {{ON|OFF},<Nrf>}

<b>Syntax</b>	<b>:STORe:SAFTer {ON OFF &lt;Nrf&gt;}   {{ON OFF},&lt;Nrf&gt;}</b>			
<b>Description</b>	Sets the current stop measurement after state and/or value			
<b>Parameter</b>	<b>Name</b>	<b>Type</b>	<b>Range</b>	<b>Default</b>
	Stop measurement after state or value	Literal   Nrf	ON   OFF   1–max seconds	None
	Stop measurement after value	Nrf   None	1–max seconds	None
<b>Explanation</b>	Sets the current state and/or value of stop measurement after. The state should be ON or OFF. The value is numeric in seconds. This command accepts one or two parameters (see examples)			
<b>Example</b>	<pre> -&gt; :STORe:SAFTer OFF -&gt; :STORe:SAFTer? &lt;- OFF,1.0 -&gt; :STORe:SAFTer 2.0           //sets the pretimed     enabled and to 1sec -&gt; :STORe:SAFTer? &lt;- ON,2.0 -&gt; :STORe:SAFTer OFF,9.0 -&gt; :STORe:SAFTer? &lt;- OFF,9.0 </pre>			

## 8.27 :STORe:ADVanced?

<b>Syntax</b>	<b>STORe:ADVanced?</b>
<b>Description</b>	Queries the current state of the individual channel configuration
<b>Parameter</b>	None
<b>Explanation</b>	Shows the current state (ON OFF) of the individual channel configuration
<b>Return Format</b>	<Boolean>
<b>Example</b>	<pre> -&gt; :STORe:ADVanced? &lt;- OFF </pre>

## 8.28 :STORe:ADVanced <Boolean>

<b>Syntax</b>	<b>:STORe:ADVanced {ON 1 OFF 0}</b>			
<b>Description</b>	Sets the individual channel configuration state			
<b>Parameter</b>				
	<b>Name</b>	<b>Type</b>	<b>Range</b>	<b>Default</b>
	{ON 1 OFF 0}	Boolean	ON 1 OFF 0	None
<b>Explanation</b>	Sets the individual channel configuration to enabled or disabled			
<b>Example</b>	<pre> -&gt; :STORe:ADVanced ON -&gt; :STORe:ADVanced? &lt;- ON                     </pre>			





---

## CHANNEL LIST ACCESS

### 9.1 :CHANNELlist:NAMes?

<b>Syntax</b>	<b>:CHANNELlist:NAMes?</b>
<b>Description</b>	Queries the long <i>name</i> and <i>ids</i> of all OXYGEN channels
<b>Parameter</b>	None
<b>Explanation</b>	Returns a list of elements alternating the numeric <i>channel-id</i> and the <i>channel-name</i>
<b>Return Format</b>	(<String >, <String>)[,< String >, <String>][,...]]   NONE
<b>Example</b>	<pre>-&gt; :CHANNEL:NAMes? &lt;- ("5770799963931934732", "AI 2/1 Sim"),     ↪ ("5770799963931934733", "AI 2/2 Sim")</pre>

## 9.2 :CHANNELlist:IDs?

<b>Syntax</b>	<b>:CHANNELlist:IDs? [&lt;channelname&gt;[,&lt;channelname&gt;[,...]]]</b>											
<b>Description</b>	Queries the ids of all or selected channels. For selection use the channels long name as parameter.											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;Channel-Name&gt;</td> <td>ASCII string</td> <td></td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<Channel-Name>	ASCII string		None
	Name	Type	Range	Default								
<Channel-Name>	ASCII string		None									
<b>Explanation</b>	Returns a list of <i>channel-ids</i> .											
<b>Return Format</b>	(<String>[,<String >[, ...]])   NONE											
<b>Example</b>	<pre> -&gt; :CHANNEL:IDs? &lt;- "5770799963931934732", "5770799963931934733" ... -&gt; :CHANNEL:ID? "AI 2/6 Sim" &lt;- :CHANNEL:ID "9310347796068433946" ... -&gt; :CHANNEL:IDs? "AI 2/1 Sim", "AI 2/2 Sim" &lt;- :CHANNEL:ID "7565484415439077397", ↵ ↵ "7565484415439077398" </pre>											

## 9.3 :CHANNELlist:ITEM<ChannelID>:ATTR

The following channel config items (properties) should be possible to access (if available for channel type):

Key	Explanation	Type
<b>Neon/Active</b>	Indicates that the selected channel is enabled	Boolean
<b>Neon/LongName</b>	Channel name and device	String
<b>Neon/Name</b>	Channel name	String
<b>Neon/PhysicalScaleFactor</b>	Scale factor for the selected channel	Floating Point
<b>Neon/PhysicalScaleOffset</b>	Scale offset for the selected channel	Floating Point
<b>Neon/Stored</b>	Indicates whether the channel data will be recorded to the datafile	ENUM
<b>Range</b>	Physical measurement range	RANGE
<b>SampleRate</b>	Sample rate (sample frequency) of an OXYGEN channel	SCALAR
<b>Unit</b>	Physical measurement unit	String
<b>Used</b>	Indicates that the selected channel is enabled	Boolean

You can obtain all possible config items with the query `:CHANNELlist:ITEM<ChannelID>:ATTR:NAMEs?` described below.

Since a single config item describes a different aspect of an OXYGEN channels, they can have different types. This can be a single value of a fundamental type (Boolean) or a compound of values with different types (SCALAR). See the next table to evaluate OXYGEN config item types:

Config Item Type	SCPI parameter type	Example
<b>Boolean</b>	<Boolean>	ON or TRUE OFF or FALSE
<b>Signed Integer</b>	Numeric <Nr1> or <Nr2> or <Nr3>	1, -5
<b>Unsigned Integer</b>	Numeric <Nr1> or <Nr2> or <Nr3>	1, 1.0E-5
<b>Floating point</b>	Numeric <Nrf>	3.14, 0.001, 1.0E-5, -5E3
<b>String</b>	<String>	"AI 2/1 Sim"
<b>ENUM</b>	<String> Literal,<String>,<String>	ENUM,"Channel- Stored","Auto"
<b>SCALAR</b>	Extended Numeric <Nrf> <Nrf>,<String> Literal,<Numeric>,<String>	0.01Hz 0.01,"Hz" SCALAR,0.0,"ms"
<b>RANGE</b>	Extended Numeric <Nrf>,<Nrf> Literal,<Numeric>,<String>,<Nu- meric>,<String>	0V,10.0V RANGE,-10.0,"V",10.0,"V"
<b>RATIO</b> (RationalScalar)	<Nr1>,<Nr1>,<String> Literal,<Numeric>,<Nu- meric>,<String>,<String>,<String>	RATIO,1,3,"V"
<b>CHANNEL_ID</b>	<String>	CHAN- NEL_ID,"199432"
<b>CHANNEL_ID_VECTOR</b>	<String>,...,<String>	CHAN- NEL_ID_VEC- TOR,"199432","199435"

For receiving the value of a single item use queries :CHANNEL-list:ITEM<ChannelID>:ATTR:VAL? or :CHANNELlist:PROPErty?.

Please note that the spectrum of config items can vary with different channel types. Some of the config items are read only and cannot be changed.

:: test: ":CHANNELlist:ITEM<ChannelID>:ATTR:NAMES?"

### 9.4 :CHANNELlist:ITEM<ChannelID>:ATTR:NAMes?

<b>Syntax</b>	<b>:CHANNELlist:ITEM&lt;ChannelID&gt;:ATTR:NAMes?</b>
<b>Description</b>	Queries the names (keys) of properties (config-items) of a given channel
<b>Parameter</b>	None
<b>Explanation</b>	Returns a list of available attribute names
<b>Return Format</b>	(<String > [,<String >, ...])   NONE
<b>Example</b>	<pre>-&gt; :CHANNEL:ITEM8316741119689883648:ATTR:NAMes? &lt;- "ChannelType", "DefaultName", "Neon/Name", "Neon/ ↳Stored"</pre>

### 9.5 :CHANNELlist:ITEM<ChannelID>:ATTR:VAL?

<b>Syntax</b>	<b>:CHANNELlist:ITEM&lt;ChannelID&gt;:ATTR:VAL? &lt;PropertyName&gt;</b>															
<b>Description</b>	Queries a specific property (config-item) of an channel															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;ChannelID&gt;</td> <td>Integer</td> <td>0-18446744073709551615</td> <td>None</td> </tr> <tr> <td>&lt;Parameter-Name&gt;</td> <td>ASCII string</td> <td></td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<ChannelID>	Integer	0-18446744073709551615	None	<Parameter-Name>	ASCII string		None
	Name	Type	Range	Default												
	<ChannelID>	Integer	0-18446744073709551615	None												
<Parameter-Name>	ASCII string		None													
<b>Explanation</b>	Returns the name of the property as string and the value of the property in the most appropriate format															
<b>Return Format</b>	ON   OFF   (<String>,<String>)   (<String>,<NRf>,<String>)   (<String>,<NRf>)   (<String>,<NRf>,<NRf>)   (<String>,<NR1>)   (<String>,<NR1>,<NR1>)   (<String>,<NRf>,<String>,<NRf>,<String>)   NONE															
<b>Example</b>	<pre>-&gt; :CHANNEL:ITEM5770799963931934732:ATTR:VAL? "Range" &lt;- (RANGE,-10.0,"V",10.0,"V")</pre>															

## 9.6 :CHANNELlist:ITEM<ChannelID>:REMove

<b>Syntax</b>	<b>:CHANNELlist:ITEM&lt;ChannelID&gt;:REMove</b>			
<b>Description</b>	Deletes the software channel			
<b>Parameter</b>				
	Name	Type	Range	Default
	<ChannelID>	Integer	0-18446744073709551615	None
<b>Explanation</b>	Deletes the software channel, for example those created using the ADDchannel command.			
<b>Example</b>	-> :CHANNEL:ITEM17555881209848922113:REM			

## 9.7 :CHANNELlist:ADDchannel?

<b>Syntax</b>	<pre>:CHANNELlist:ADDchannel? &lt;ModuleName&gt; [&lt;InputChannelID&gt;,&lt;InputChannelID&gt;][,...]]] :CHANNELlist:ADDchannel &lt;ModuleName&gt; [&lt;InputChannelID&gt;,&lt;InputChannelID&gt;][,...]]]</pre>												
<b>Description</b>	Create a new software channel												
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;Module-Name&gt;</td> <td>ASCII String</td> <td>Oxygen Id</td> <td>None</td> </tr> <tr> <td>&lt;InputChannelID&gt;</td> <td>ASCII String</td> <td>Oxygen Id</td> <td>None</td> </tr> </tbody> </table>	Name	Type	Range	Default	<Module-Name>	ASCII String	Oxygen Id	None	<InputChannelID>	ASCII String	Oxygen Id	None
Name	Type	Range	Default										
<Module-Name>	ASCII String	Oxygen Id	None										
<InputChannelID>	ASCII String	Oxygen Id	None										
<b>Explanation</b>	<p>Creates a new channel of the specified type. The specified &lt;InputChannelID&gt; elements are forwarded to the target module. The query syntax returns a list containing the ids of all created channels.</p> <p>The following types are currently supported:</p> <table border="1"> <thead> <tr> <th>&lt;ModuleName&gt;</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>"formula"</td> <td>Creates a single formula channel. Any specified input channels will be summed up to create a default formula. If no input channels are specified the channel will be in an error state until the property "FORMULA/Formula" is set to a valid value.</td> </tr> </tbody> </table>	<ModuleName>	Description	"formula"	Creates a single formula channel. Any specified input channels will be summed up to create a default formula. If no input channels are specified the channel will be in an error state until the property "FORMULA/Formula" is set to a valid value.								
<ModuleName>	Description												
"formula"	Creates a single formula channel. Any specified input channels will be summed up to create a default formula. If no input channels are specified the channel will be in an error state until the property "FORMULA/Formula" is set to a valid value.												
<b>Return Format (for query)</b>	(<String>[,<String>[,...]])   NONE												
<b>Example</b>	<pre>-&gt; :CHANNEL:ADD? "formula" &lt;- ("17555881209848922113")  -&gt; :CHANNEL:ADD "formula", "1755120948922142", ↵ ↵ "17555881209848922113"</pre>												



## 9.8 :CHANNELlist:PROPerTy?

<b>Syntax</b>	<b>:CHANNELlist:PROPerTy? &lt;ChannelID&gt;,&lt;PropertyName&gt;</b>			
<b>Description</b>	Queries a specific property (config-item) of an OXYGEN channel			
<b>Parameter</b>	<b>Name</b>	<b>Type</b>	<b>Range</b>	<b>Default</b>
	<ChannelID>	ASCII string	Oxygen Id	None
	<Property-Name>	ASCII string		None
<b>Explanation</b>	Returns the value(s) of the property in the most appropriate format. For finding possible channel IDs you can use the SCPI query :CHANNELlist:NAMes? or :CHANNELlist:IDs?. To evaluate all possible property names see query :CHANNELlist:ITEM<ChannelID>:ATTR:NAMes?.			
<b>Return Format</b>	ON   OFF   (<String>,<String>)   (<String>,<NRf>,<String>)   (<String>,<NRf>)   (<String>,<NRf>,<NRf>)   (<String>,<NR1>)   (<String>,<NR1>,<NR1>)   (<String>,<NRf>,<String>,<NRf>,<String>)   NONE			
<b>Example</b>	<pre>-&gt; :CHANNEL:PROPerTy? "5770799963931934732", "Range" &lt;- (RANGE,-10.0, "V", 10.0, "V")</pre>			

## 9.9 :CHANNELlist:PROPerTy

<b>Syntax</b>	<b>:CHANNELlist:PROPerTy &lt;ChannelID&gt;,&lt;Property-Name&gt;,&lt;PARAM&gt;[,&lt;PARAM&gt;[,...]]</b>																			
<b>Description</b>	Set the value[s] of a specific property (config-item) of a given channel																			
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;ChannelID&gt;</td> <td>ASCII string</td> <td>Oxygen Id</td> <td>None</td> </tr> <tr> <td>&lt;Property-Name&gt;</td> <td>ASCII string</td> <td></td> <td>None</td> </tr> <tr> <td>&lt;PARAM&gt;</td> <td>Numeric, ASCII string, Mnemonic</td> <td></td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<ChannelID>	ASCII string	Oxygen Id	None	<Property-Name>	ASCII string		None	<PARAM>	Numeric, ASCII string, Mnemonic		None
Name	Type	Range	Default																	
<ChannelID>	ASCII string	Oxygen Id	None																	
<Property-Name>	ASCII string		None																	
<PARAM>	Numeric, ASCII string, Mnemonic		None																	
<b>Explanation</b>	Set the value[s] of a channel specific config item. The number and the type of the value[s] depends from the type of the config item. For finding possible channel IDs you can use the SCPI query :CHANNELlist:NAMEs? or :CHANNELlist:IDs?. To evaluate all possible property names see query :CHANNELlist:ITEM<ChannelID>:ATTR:Names?. Use the CHANNELlist:CONSTRAINT query below to see restrictions for the item. Please note that some of the attributes are readonly and therefore cannot be set via SCPI.																			
<b>Example</b>	<pre> -&gt; :CHANNEL:PROP? "15451287354374881280", "Unit" &lt;- :CHANNEL:PROP "v" -&gt; :CHANNEL:PROP "15451287354374881280", "Unit", "A" -&gt; :CHANNEL:PROP? "15451287354374881280", "Unit" &lt;- :CHANNEL:PROP "A" ... -&gt; :CHANNEL:PROP? "15451287354374881280", "Neon/ ↳Stored" &lt;- :CHANNEL:PROP (ENUM, "ChannelStored", "No") -&gt; :CHANNEL:PROP "15451287354374881280", "Neon/ ↳Stored", "Auto" -&gt; :CHANNEL:PROP? "15451287354374881280", "Neon/ ↳Stored" &lt;- :CHANNEL:PROP (ENUM, "ChannelStored", "Auto") -&gt; :CHANNEL:PROP "15451287354374881280", "Neon/ ↳Stored", ENUM, "ChannelStored", "No" -&gt; :CHANNEL:PROP? "15451287354374881280", "Neon/ ↳Stored" &lt;- :CHANNEL:PROP (ENUM, "ChannelStored", "No") </pre>																			

## 9.10 :CHANNELlist:CONSTRaint?

<b>Syntax</b>	<b>:CHANNELlist:CONSTRaint? &lt;ChannelID&gt;,&lt;PropertyName&gt;</b>															
<b>Description</b>	Queries the constraints of a specific property (config-item) of a given channel															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;ChannelID&gt;</td> <td>ASCII string</td> <td>0-18446744073709551615</td> <td>None</td> </tr> <tr> <td>&lt;Property-Name&gt;</td> <td>ASCII string</td> <td></td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<ChannelID>	ASCII string	0-18446744073709551615	None	<Property-Name>	ASCII string		None
	Name	Type	Range	Default												
	<ChannelID>	ASCII string	0-18446744073709551615	None												
<Property-Name>	ASCII string		None													
<b>Explanation</b>	Returns the constraint(s) of a channel property in the most appropriate format. For finding possible channel IDs you can use the :CHANNELlist:NAMes? or :CHANNELlist:IDs? query. To evaluate all possible property names see query :CHANNELlist:ITEM<ChannelID>:ATTR:Names?.															
<b>Return Format</b>	(<String>,<String>)   (<String>,<NRf>,<String>)   (<String>,<NRf>)   (<String>,<NRf>,<NRf>)   (<String>,<NR1>)   (<String>,<NR1>,<NR1>)   (<String>,<NRf>,<String>,<NRf>,<String>)   NONE															
<b>Example</b>	<pre>-&gt; :CHANNEL:CONSTR? "3348707789336739861", "InputType" &lt;- :CHANNEL:CONSTR (STRING, "SE"), (STRING, "DIFF")</pre>															

## 9.11 :CHANNELlist:TIMing:HIGHest?

<b>Syntax</b>	<b>:CHANNELlist:TIMing:HIGHest?</b>
<b>Description</b>	Queries the highest timing interval from all used and valid sync channels
<b>Parameter</b>	None
<b>Explanation</b>	Returns the timing interval of the channel with the lowest sample rate in seconds
<b>Return Format</b>	<NRf> NONE
<b>Example</b>	<pre>-&gt; :CHANNELlist:TIMing:HIGHest? &lt;- :CHANNEL:TIM:HIGH 1.0E-3</pre>

### 9.12 :CHANNELlist:TIMing:LOWest?

<b>Syntax</b>	<b>:CHANNELlist:TIMing:LOWest?</b>
<b>Description</b>	Queries the lowest timing interval from all used and valid sync channels
<b>Parameter</b>	None
<b>Explanation</b>	Returns the timing interval of the channel with the highest sample rate in seconds
<b>Return Format</b>	<NRf> NONE
<b>Example</b>	<pre>-&gt; :CHANNELlist:TIMing:LOWest? &lt;- CHANNEL:TIM:LOW 1.0E-4</pre>

### 9.13 :CHANNELlist:SATuration:VALue?

<b>Syntax</b>	<b>:CHANNELlist:SATuration:VALue? {ALL   &lt;ChannelID&gt;}</b>												
<b>Description</b>	Queries the saturation of all or a selected OXYGEN scalar channel												
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>ALL</td> <td>Literal</td> <td></td> <td>None</td> </tr> <tr> <td>&lt;ChannelID&gt;</td> <td>ASCII string</td> <td>Oxygen Id</td> <td>None</td> </tr> </tbody> </table>	Name	Type	Range	Default	ALL	Literal		None	<ChannelID>	ASCII string	Oxygen Id	None
Name	Type	Range	Default										
ALL	Literal		None										
<ChannelID>	ASCII string	Oxygen Id	None										
<b>Explanation</b>	Returns the saturation in % since acquisition start.												
<b>Return Format</b>	<NRf> NONE												
<b>Example</b>	<pre>-&gt; :CHANNELlist:IDs? "AI 2/1 Sim" &lt;- :CHANNEL:ID "17770359696083910677" ... -&gt; :CHANNELlist:SATuration:VALue? ↪ "17770359696083910677" &lt;- :CHANNEL:SAT:VAL 79.999988079070334 ... -&gt; :CHANNELlist:SATuration:VALue? ALL &lt;- :CHANNEL:SAT:VAL ("4045076881718902815", 1. ↪ 420911867502358E-2), ("4045076881718902816", 97. ↪ 500000002468752), ("4045076881718902819", 1.420911867502358E-2), ↪ ("17770359696083910677", 79.999988079070334)</pre>												

## 9.14 :CHANNELlist:SATuration:RESet

<b>Syntax</b>	<b>:CHANNELlist:SATuration:RESet</b>
<b>Description</b>	Resets the saturation of all OXYGEN scalar channels
<b>Parameter</b>	None
<b>Explanation</b>	Rejects the saturation value of all OXYGEN scalar channels and restarts saturation calculation again.
<b>Example</b>	<pre>-&gt; :CHANNELlist:IDs? "AI 2/1 Sim" &lt;- :CHANNEL:ID "17770359696083910677" ... -&gt; :CHANNELlist:SATuration:VALue?↵ ↵"17770359696083910677" &lt;- :CHANNEL:SAT:VAL 79.999988079070334 ... -&gt; :CHANNELlist:SATuration:RESet -&gt; :CHANNELlist:SATuration:VALue?↵ ↵"17770359696083910677" &lt;- :CHANNEL:SAT:VAL 10.942228000000001</pre>

### 9.15 :CHANNELlist:FIRResponse

<b>Syntax</b>	<b>:CHANNELlist:FIRResponse? {&lt;ChannelID&gt;[,&lt;size&gt; [&lt;logarithmic frequency&gt;,&lt;logarithmic amplitude&gt;]]}</b>																							
<b>Description</b>	Queries the Finite Impuls Response of a FIR group channel																							
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;ChannelID&gt;</td> <td>ASCII string</td> <td>Oxygen Id</td> <td>None</td> </tr> <tr> <td>&lt;Size&gt;</td> <td>Integer</td> <td>10 - 2000</td> <td>1000</td> </tr> <tr> <td>&lt;use logarithmic frequency&gt;</td> <td>Boolean</td> <td>ON or TRUE OFF or FALSE</td> <td>TRUE</td> </tr> <tr> <td>&lt;use logarithmic amplitude (db)&gt;</td> <td>Boolean</td> <td>ON or TRUE OFF or FALSE</td> <td>TRUE</td> </tr> </tbody> </table>				Name	Type	Range	Default	<ChannelID>	ASCII string	Oxygen Id	None	<Size>	Integer	10 - 2000	1000	<use logarithmic frequency>	Boolean	ON or TRUE OFF or FALSE	TRUE	<use logarithmic amplitude (db)>	Boolean	ON or TRUE OFF or FALSE	TRUE
Name	Type	Range	Default																					
<ChannelID>	ASCII string	Oxygen Id	None																					
<Size>	Integer	10 - 2000	1000																					
<use logarithmic frequency>	Boolean	ON or TRUE OFF or FALSE	TRUE																					
<use logarithmic amplitude (db)>	Boolean	ON or TRUE OFF or FALSE	TRUE																					
<b>Explanation</b>	Returns a size number of frequency/amplitude pairs.																							
<b>Return Format</b>	<(NR3,NR3),(NR3,NR3),...> NONE																							
<b>Example</b>	<pre> -&gt; :CHANNELlist:IDs? "FIR_1" &lt;- :CHANNEL:ID "6693480426557669376" ... -&gt; :CHANNELlist:FIR? "6693480426557669376" &lt;- :CHANNEL:FIR (2.5E+2,-9.757726E-3),(3.1473135E+2, ↪1.7361388E-2),(3.962233E+2,9.5682509E-2),... ... -&gt; :CHANNELlist:FIR? "6693480426557669376",10 &lt;- :CHANNEL:FIR (2.5E+2,-9.757726E-3),(3.1473135E+2, ↪1.7361388E-2),(3.962233E+2,9.5682509E-2),... ... -&gt; :CHANNELlist:FIR? "6693480426557669376",100,OFF,ON &lt;- :CHANNEL:FIR (0.0E+0,10.0E-1),(2.5E+1,9. ↪9526759E-1),(5.0E+1,9.9768966E-1),(7.5E+1,9. ↪9899028E-1), (1.0E+2,9.9419968E-1),(1.25E+2,1.0003403E+0),(1.5E+2, ↪9.9625236E-1),(1.75E+2,9.9527457E-1), (2.0E+2,1.0019049E+0),(2.25E+2,9.9261725E-1),(2.5E+2, ↪9.9887723E-1),(2.75E+2,1.0009722E+0), (3.0E+2,9.8937606E-1),(3.25E+2,1.005167E+0),(3.5E+2, ↪9.9566524E-1),... ... </pre>																							

## 9.16 Channel Actions

For OXYGEN channels some actions can be invoked, which results more or less in an config update for the channel. Not all actions can be invoked on each channel at any time. This depends on the channel type and its mode. With the following SCPI subsystems the corresponding action can be invoked or queried.

The following table shows the available action subsystems for OXYGEN channels

Action	Description
<b>Channel Zero</b> <ul style="list-style-type: none"> <li>• :CHANNELlist:ACTion:ZERO</li> <li>• :CHANNELlist:ITEM&lt;ChannelID&gt;:ACTion:ZERO</li> </ul>	The zero action calculates the channel scaling offset
<b>Bridge Balance</b> <ul style="list-style-type: none"> <li>• :CHANNELlist:ACTion:BALance</li> <li>• :CHANNELlist:ITEM&lt;ChannelID&gt;:ACTion:BALance</li> </ul>	The bridge balance action calculates the sensor offset in bridge mode
<b>Shunt ON</b> <ul style="list-style-type: none"> <li>• :CHANNELlist:ACTion:SHUNT:ON</li> <li>• :CHANNELlist:ITEM&lt;ChannelID&gt;:ACTion:SHUNT:ON</li> </ul>	The shunt on action enables the bridge shunt
<b>Shunt OFF</b> <ul style="list-style-type: none"> <li>• :CHANNELlist:ACTion:SHUNT:OFF</li> <li>• :CHANNELlist:ITEM&lt;ChannelID&gt;:ACTion:SHUNT:OFF</li> </ul>	The shunt off action disables the bridge shunt

For each action an available query and an exec command is registered.

### 9.17 :CHANNELlist:ITEM<ChannelID>:ACTion:<action>:AVAI?

<b>Syntax</b>	<b>:CHANNELlist:ITEM&lt;ChannelID&gt;:ACTion:&lt;action&gt;:AVAI</b>											
<b>Description</b>	Queries the current <action> availability of the given OXYGEN channel											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;ChannelID&gt;</td> <td>Integer</td> <td>0-18446744073709551615</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<ChannelID>	Integer	0-18446744073709551615	None
	Name	Type	Range	Default								
	<ChannelID>	Integer	0-18446744073709551615	None								
<b>Explanation</b>	Checks if an <action> for the corresponding channel is able to invoke.											
<b>Return Format</b>	ON   OFF   NONE											
<b>Example</b>	<pre>-&gt; :CHANNELlist:ITEM5310588377010012183:ACTion:SHUNT:OFF:AVAI? ↪ //full command &lt;- OFF ↪ //action is not available</pre>											

### 9.18 :CHANNELlist:ITEM<ChannelID>:ACTion:<action>:EXEc

<b>Syntax</b>	<b>:CHANNELlist:ITEM&lt;ChannelID&gt;:ACTion:&lt;action&gt;:EXEc</b>											
<b>Description</b>	Execute the <action> of the given OXYGEN channel											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;ChannelID&gt;</td> <td>Integer</td> <td>0-18446744073709551615</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<ChannelID>	Integer	0-18446744073709551615	None
	Name	Type	Range	Default								
	<ChannelID>	Integer	0-18446744073709551615	None								
<b>Explanation</b>	Invoke the <action> on the corresponding channel. This is the default command of the action subsystem.											
<b>Example</b>	<pre>-&gt; :CHANNELlist:ITEM5310588377010012183:ACTion:ZERO:EXEc ↪ //full command ... -&gt; :CHANNEL:ITEM5770799963931934732:ACTION:ZERO ↪ //used as default -&gt; :CHANNEL:ITEM5310588377010012183:ACT:SHUNT:OFF ↪ //used as default</pre>											



### 9.19 CHANNELlist:ACTion:<action>:AVAI? <channelname\_or\_id>[,<channelname\_or\_id>[,...]]

<b>Syntax</b>	:CHANNELlist:ACTion:<action>:AVAI? <channelname_or_id>[,<channelname_or_id>[,...]]											
<b>Description</b>	Queries the action state for several channels											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;channelname_or_id&gt;</td> <td>ASCII string</td> <td>Oxygen channel name or Id</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<channelname_or_id>	ASCII string	Oxygen channel name or Id	None
Name	Type	Range	Default									
<channelname_or_id>	ASCII string	Oxygen channel name or Id	None									
<b>Explanation</b>	Checks if an <action> for the corresponding channels is able to invoke.											
<b>Return Format</b>	(<String >, ON   OFF)[,<String >, ON   OFF],...   NONE											
<b>Example</b>	<pre>-&gt; :CHANNELlist:ACTion:ZERO:AVAI? ↪ "5310588377010012160", "5310588377010012183" &lt;- ("5310588377010012160", OFF), ("5310588377010012183", ↪ ON)</pre>											

### 9.20 CHANNELlist:ACTion:<action>:EXEC <channelname\_or\_id>[,<channelname\_or\_id>[,...]]

<b>Syntax</b>	CHANNELlist:ACTion:<action>:EXEC <channelname_or_id>[,<channelname_or_id>[,...]]											
<b>Description</b>	Executes the action on selected channels											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;channelname_or_id&gt;</td> <td>ASCII string</td> <td>Oxygen channel name or Id</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<channelname_or_id>	ASCII string	Oxygen channel name or Id	None
Name	Type	Range	Default									
<channelname_or_id>	ASCII string	Oxygen channel name or Id	None									
<b>Explanation</b>	The zero action resets the offset of the linear scaling function for a set of channels.											
<b>Example</b>	<pre>-&gt; :CHANNELlist:ACTion:ZERO:EXEC ↪ "16823196399452553216", "AI 1/2 Sim" ... -&gt; :CHANNEL:ACT:ZERO "5310588377010012160", ↪ "5310588377010012183"</pre>											

## MEASUREMENT VALUES

### 10.1 :RATE {<num>[<unit>]|NONE}

<b>Syntax</b>	<b>:RATE {&lt;num&gt;[&lt;unit&gt;] NONE}</b>											
<b>Description</b>	Set numeric data aggregation time for output											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>{&lt;num&gt;[&lt;unit&gt;]   NONE}</td> <td>Nrf,Literal</td> <td>1-5000ms   0.001-5s   NONE</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	{<num>[<unit>]   NONE}	Nrf,Literal	1-5000ms   0.001-5s   NONE	None
Name	Type	Range	Default									
{<num>[<unit>]   NONE}	Nrf,Literal	1-5000ms   0.001-5s   NONE	None									
<b>Explanation</b>	<p>Set numeric data aggregation for output. If “NONE” specified, the aggregation time is off and the last hold value will be returned on :NUM:NORM:VAL?</p> <p>The resolution is milliseconds.</p> <p>Since revision 1.6: Data aggregation is implemented for scalar values only. More precisely, the last hold value will be returned for array channels.</p>											
<b>Example</b>	-> :RATE 500ms											

### 10.2 :RATE?

<b>Syntax</b>	<b>:RATE?</b>
<b>Description</b>	Queries the numeric data aggregation time for output
<b>Parameter</b>	
<b>Explanation</b>	The returned numeric is without unit according to the SCPI reference. But, the implicit unit is [s].
<b>Return Format</b>	<NRf> NONE
<b>Example</b>	<pre>-&gt; :RATE 500ms -&gt; :RATE? &lt;- 5.0E-1</pre>

### 10.3 :NUMeric:NORMal:ITEMS <channel>[,<channel>[,...]]

<b>Syntax</b>	<b>:NUMeric:NORMal:ITEMS &lt;channel&gt; [,&lt;channel&gt; [,...]]</b>			
<b>Description</b>	Set numeric data output items			
<b>Parameter</b>	<b>Name</b>	<b>Type</b>	<b>Range</b>	<b>Default</b>
	<channel>	ASCII String	Oxygen Channel Name	None
	"ABS-TIME"	ASCII String	Include Absolute Time Channel "YYYY-MM-DDTHH:mm:ss.sss"	
	"REL-TIME"	ASCII String	Include Relative Time Channel <NRf> Seconds since Acquisition Start	
<b>Explanation</b>	Set numeric data output items starting from index 1 on directly			
<b>Example</b>	<pre>-&gt; :NUMeric:NORMal:ITEMS "Channel-Name1", "U1_ ↳tRMS@PowerGroup"</pre>			

### 10.4 :NUMeric:NORMal:ITEMS?

<b>Syntax</b>	<b>:NUMeric:NORMal:ITEMS?</b>
<b>Description</b>	Query numeric data output items
<b>Parameter</b>	None
<b>Explanation</b>	Get the full list of channels in the numeric data output system
<b>Return Format</b>	String[, [String[,...]]]
<b>Example</b>	<pre>-&gt; :NUMeric:NORMal:ITEMS "ABS-TIME", "U1_ ↳tRMS@PowerGroup" -&gt; :NUMeric:NORMal:ITEMS? -&gt; "ABS-TIME", "U1_tRMS@PowerGroup"</pre>

## 10.5 :NUMeric:NORMAL:ITEM<x> <channel>

<b>Syntax</b>	<b>:NUMeric:NORMAL:ITEM&lt;x&gt; &lt;channel&gt;</b>											
<b>Description</b>	Set output item to specified channel name or system channel											
<b>Parameter</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Name</th> <th style="width: 25%;">Type</th> <th style="width: 30%;">Range</th> <th style="width: 20%;">Default</th> </tr> </thead> <tbody> <tr> <td>&lt;channel&gt;</td> <td>String</td> <td>Oxygen Channel Name or a system channel ("ABS-TIME", "REL-TIME")</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<channel>	String	Oxygen Channel Name or a system channel ("ABS-TIME", "REL-TIME")	None
	Name	Type	Range	Default								
<channel>	String	Oxygen Channel Name or a system channel ("ABS-TIME", "REL-TIME")	None									
<b>Explanation</b>												
<b>Example</b>	-> :NUMeric:NORMAL:ITEM1 "U1_tRMS@PowerGroup"											

## 10.6 :NUMeric:NORMAL:ITEM<x>?

<b>Syntax</b>	<b>:NUMeric:NORMAL:ITEM&lt;x&gt;?</b>
<b>Description</b>	Query channel name of output item
<b>Parameter</b>	None
<b>Explanation</b>	
<b>Return Format</b>	String
<b>Example</b>	<pre> -&gt; :NUMeric:NORMAL:ITEM1 "U1_tRMS@PowerGroup" -&gt; :NUMeric:NORMAL:ITEM1? &lt;- "U1_tRMS@PowerGroup" </pre>

## 10.7 :NUMeric:NORMal:CLEar {ALL|<NUM>[,<NUM>]}

<b>Syntax</b>	<b>:NUMeric:NORMal:CLEar {ALL &lt;num&gt;[,&lt;num&gt;]}</b>															
<b>Description</b>	Clear (Set to NONE) given items															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;num&gt;</td> <td>Integer</td> <td>1 - 32768</td> <td>None</td> </tr> <tr> <td>ALL</td> <td>Literal</td> <td></td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<num>	Integer	1 - 32768	None	ALL	Literal		None
	Name	Type	Range	Default												
	<num>	Integer	1 - 32768	None												
	ALL	Literal		None												
<b>Explanation</b>	Clear (Set to NONE) given items. If "ALL" is specified the hole list is cleared															
<b>Example</b>	<pre>-&gt; :NUMeric:NORMal:ITEMS "ABS-TIME", "U1_ ↳tRMS@PowerGroup" - -&gt; :NUMeric:NORMal:CLEar ALL -&gt; :NUMeric:NORMal:ITEMS? &lt;- 0</pre>															

## 10.8 :NUMeric:NORMal:DELeTe <NUM>[,<NUM>]

<b>Syntax</b>	<b>:NUMeric:NORMal:DELeTe {&lt;num&gt;[,&lt;num&gt;]}</b>											
<b>Description</b>	Delete Items, shift remaining to left											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;num&gt;</td> <td>Integer</td> <td>1 - 32768</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<num>	Integer	1 - 32768	None
	Name	Type	Range	Default								
	<num>	Integer	1 - 32768	None								
<b>Explanation</b>												
<b>Example</b>	<pre>-&gt; :NUMeric:NORMal:ITEMS "ABS-TIME", "U1_ ↳tRMS@PowerGroup" -&gt; :NUMeric:NORMal:DELeTe 1 -&gt; :NUMeric:NORMAL:ITEM1? &lt;- "U1_tRMS@PowerGroup"</pre>											

## 10.9 :NUMeric:NORMal:NUMber {<num>|ALL}

<b>Syntax</b>	<b>:NUMeric:NORMal:NUMber {&lt;num&gt; ALL}</b>															
<b>Description</b>	Sets the number of items to be transferred beginning from index 0															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;num&gt;</td> <td>Integer</td> <td>1 - 32768</td> <td>15</td> </tr> <tr> <td>ALL</td> <td>Literal</td> <td></td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<num>	Integer	1 - 32768	15	ALL	Literal		None
	Name	Type	Range	Default												
	<num>	Integer	1 - 32768	15												
	ALL	Literal		None												
<b>Explanation</b>																
<b>Example</b>	<pre>-&gt; :NUMeric:NORMal:ITEMS "ABS-TIME", "U1_ ↳tRMS@PowerGroup" -&gt; :NUMeric:NORMal:NUMber 2</pre>															

## 10.10 :NUMeric:NORMal:NUMber?

<b>Syntax</b>	<b>:NUMeric:NORMal:NUMber?</b>
<b>Description</b>	Queries the number of items to be transferred
<b>Parameter</b>	None
<b>Explanation</b>	Queries the number of items to be transferred
<b>Return Format</b>	Nr1
<b>Example</b>	<pre>-&gt; :NUMeric:NORMal:ITEMS "ABS-TIME", "U1_ ↳tRMS@PowerGroup" -&gt; :NUMeric:NORMal:NUMber 1 -&gt; :NUMeric:NORMal:NUMber? &lt;- 1</pre>

## 10.11 :NUMeric:NORMal:DIM<x> { <i\_max> | <i\_list> | MAX }

<b>Syntax</b>	<b>:NUMeric:NORMal:DIM&lt;x&gt; { &lt;i_max&gt;   &lt;i_list&gt;   MAX }</b>																			
<b>Description</b>	Selects indices of array channels to be included in output of :NUMeric:NORMal:VALue?																			
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;i_max&gt;</td> <td>Integer</td> <td>1 - 32768</td> <td>None</td> </tr> <tr> <td>&lt;i_list&gt;</td> <td>Numeric list</td> <td></td> <td>None</td> </tr> <tr> <td>MAX</td> <td>Literal</td> <td></td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<i_max>	Integer	1 - 32768	None	<i_list>	Numeric list		None	MAX	Literal		None
Name	Type	Range	Default																	
<i_max>	Integer	1 - 32768	None																	
<i_list>	Numeric list		None																	
MAX	Literal		None																	
<b>Explanation</b>	<p>By specifying the single value parameter &lt;I_MAX&gt;, you limit the number of elements of array channels to the first &lt;I_MAX&gt; values.</p> <p>If you want to include a specific set of indices, you can provide an &lt;I_LIST&gt;.</p> <p>In order to reset the selected indices to all available indices, use the parameter mnemonic MAX.</p> <p>If &lt;I_MAX&gt; or entries from &lt;I_LIST&gt; lie outside the range supported by the array channel, only indices in the allowed range will be saved, and an error will be generated.</p> <p>Any call to this command will clear the effect of a previously issued :NUM:NORM:DIM command for the selected item.</p> <p>Note that the indexing of array channels starts at 1.</p>																			
<b>Example</b>	<pre> -&gt; :NUMeric:NORMal:ITEM1 "AI 1/1" -&gt; :NUMeric:NORMal:ITEM3 "U1_hRMS@PowerGroup" -&gt; :NUMeric:NORMal:DIM1? &lt;- 1 -&gt; :NUMeric:NORMal:DIM2? &lt;- 1 -&gt; :NUMeric:NORMal:DIM3? &lt;- 128 -&gt; :NUMeric:NORMal:DIM3 10 -&gt; :NUMeric:NORMal:DIM3? &lt;- 10 -&gt; :NUMeric:NORMal:DIM3 MAX -&gt; :NUMeric:NORMal:DIM3? &lt;- 128 -&gt; :NUMeric:NORMal:DIM3 (1:10,50,60:70) -&gt; :NUMeric:NORMal:DIM3? &lt;- (1:10,50,60:70) </pre>																			

## 10.12 :NUMeric:NORMal:DIM<x>?

<b>Syntax</b>	<b>:NUMeric:NORMal:DIM&lt;x&gt;?</b>
<b>Description</b>	Queries the dimension of the data to be transferred for one item
<b>Parameter</b>	None
<b>Explanation</b>	<ul style="list-style-type: none"> <li>• for each scalar channel, 1 will be returned;</li> <li>• for each NONE item, 1 will be returned;</li> <li>• for array channels, the output format will vary depending on how the item was set up with :NUM:DIM&lt;X&gt;: <ul style="list-style-type: none"> <li>– if no upper bound was set, the number of items of the array channel will be returned</li> <li>– if an upper bound was set, that value will be returned</li> <li>– if a range of indices was selected, that range will be returned</li> </ul> </li> </ul>
<b>Return Format</b>	<NR1> <NumericList>
<b>Example</b>	Cf. :NUMeric:NORMal:DIM<x> command.

## 10.13 :NUMeric:NORMal:DIMS?

<b>Syntax</b>	<b>:NUMeric:NORMal:DIMS?</b>
<b>Description</b>	Queries the dimension of the data to be transferred for each item
<b>Parameter</b>	None
<b>Explanation</b>	The list returned will contain one entry for each item in the format used for :NUM:NORM:DIM<x>?
<b>Return Format</b>	[<NR1> <NumericList>[,...]]
<b>Example</b>	Cf. :NUMeric:NORMal:DIM<x> command.



## 10.14 :NUMeric:NORMal:FORMat {ASCII|BIN\_INTEL|BIN\_MOTOROLA}

<b>Syntax</b>	<b>:NUMeric:NORMal:FORMat {ASCII BIN_INTEL BIN_MOTOROLA}</b>											
<b>Description</b>	Sets the requested output format											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>Format</td> <td>ASCII literal(s)</td> <td> <ul style="list-style-type: none"> <li>• ASCII</li> <li>• BIN_INTEL</li> <li>• BIN_MOTOROLA</li> </ul> </td> <td>ASCII</td> </tr> </tbody> </table>				Name	Type	Range	Default	Format	ASCII literal(s)	<ul style="list-style-type: none"> <li>• ASCII</li> <li>• BIN_INTEL</li> <li>• BIN_MOTOROLA</li> </ul>	ASCII
	Name	Type	Range	Default								
Format	ASCII literal(s)	<ul style="list-style-type: none"> <li>• ASCII</li> <li>• BIN_INTEL</li> <li>• BIN_MOTOROLA</li> </ul>	ASCII									
<b>Explanation</b>	<p>When format is set to ASCII, all numbers are returned in an ASCII format (NR3 for values, timestamps in NR2 format or as Strings)</p> <p>When format is set to BIN_INTEL or BIN_MOTOROLA, all values are returned in a single array of IEEE 754 float32 values (in the corresponding byte order). See <i>Arbitrary Block</i> for more information. Values that are not convertible to float32 values are reported as NaN.</p>											
<b>Example</b>	-> :NUMeric:NORMal:FORMat BIN_INTEL											

## 10.15 :NUMeric:NORMal:FORMat?

<b>Syntax</b>	<b>:NUMeric:NORMal:FORMat?</b>
<b>Description</b>	Queries the currently configured output format
<b>Parameter</b>	None
<b>Explanation</b>	Returns the currently configured output format
<b>Return Format</b>	{ASCII BIN_INTEL BIN_MOTOROLA}
<b>Example</b>	<pre>-&gt; :NUMeric:NORMal:FORM ASCII -&gt; :NUMeric:NORMal:FORM? &lt;- ASCII</pre>

## 10.16 :NUMeric:NORMal:VALue? [<NUM>]

<b>Syntax</b>	<b>:NUMeric:NORMal:VALue? [&lt;num&gt;]</b>											
<b>Description</b>	Query numeric values of the preset item list											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;num&gt;</td> <td>Integer</td> <td>1...32768: Only return the value of the entry at index &lt;num&gt; in ITEMS.</td> <td>Returns a flat list containing the values of all entries in ITEMS.</td> </tr> </tbody> </table>				Name	Type	Range	Default	<num>	Integer	1...32768: Only return the value of the entry at index <num> in ITEMS.	Returns a flat list containing the values of all entries in ITEMS.
	Name	Type	Range	Default								
<num>	Integer	1...32768: Only return the value of the entry at index <num> in ITEMS.	Returns a flat list containing the values of all entries in ITEMS.									
<b>Explanation</b>	<p>If no index is specified, all values in the list will be returned in a comma separated list (for ASCII format) up to the number of items to be transferred.</p> <p>Since 1.6: The list returned by this query will be flat, i.e. values for array channels will be included as a sequence of values. Use the :NUM:NORM:DIMS? query to retrieve the number of elements that can be expected for each item.</p> <p>If an index is specified the value for the item at index is returned, regardless if the index is higher than the number of items to be transferred.</p> <p>Since 1.20: When the format is BIN_INTEL or BIN_MOTOROLA, instead of comma separated ASCII characters, on block of binary &lt;Arbitrary Block&gt; is returned. The data layout is the same as for ASCII.</p>											
<b>Return Format</b>	{ASCII BIN_INTEL BIN_MOTOROLA}											
<b>Example</b>	<pre>-&gt; :NUMeric:NORMal:ITEMS "REL-TIME", "U1_ ↳tRMS@PowerGroup" -&gt; :NUMeric:NORMal:VALue? &lt;- 15.2,2.4553 -&gt; :NUMeric:NORMal:VALue? 2 &lt;- 2.4553 -&gt; :NUMeric:NORMal:ITEM3 "U1_frMS@PowerGroup" -&gt; :NUMeric:NORMal:VALue? 3 &lt;- 2.3451,-1.2425,...,4.0124</pre>											



## EXTERNAL DATA LOGGING (ELOG)

The ELOG subsystem allows the SCPI user to retrieve synchronized access to multiple channels via statistics calculations.

First, the subsystem needs to be configured. Possible parameters are a channel list, aggregation calculations, aggregation duration as well as result formats and timestamp formats.

After configuration, the user can start the computations and fetch all values. By continuously requesting data records, gap-less readout is possible. Data is kept available inside Oxygen for at least 20 seconds before fetching of old samples is no longer possible.

ELOG cannot be configured during recording or armed triggers state.

### 11.1 :ELOG:ITEMs <channel>[,<channel>[,...]]

<b>Syntax</b>	<b>:ELOG:ITEMs &lt;channel&gt; [,&lt;channel&gt; [,...]]</b>											
<b>Description</b>	Defines the ordered list of requested channels											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;channel&gt;</td> <td>ASCII String</td> <td>Oxygen Channel Name</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<channel>	ASCII String	Oxygen Channel Name	None
Name	Type	Range	Default									
<channel>	ASCII String	Oxygen Channel Name	None									
<b>Explanation</b>	<p>The user can specify a list of multiple scalar channels. The channels need not share the same sample-rate or raw numeric format. If a channel is not compatible, an error is generated, and the channel is not added to the list. If one of the registered channels becomes unused or is changed, the whole dataset will be discarded and the ELOG system will be in an error state. Use :ELOG:STATE? to get more details.</p> <p>The following channel types are supported: Analog, Counter, Math, Statistics, Filter, Power, AI Async Pad, CAN Signals, GPS-Position/Velocity/Heading/SecondsOfDay/Distance and Acceleration. For proper operation, a minimum sample rate of 5Hz is suggested.</p>											
<b>Example</b>	-> :ELOG:ITEMs "U1_tRMS@PowerGroup", "AI 1/2"											

## 11.2 :ELOG:ITEMs?

<b>Syntax</b>	<b>:ELOG:ITEMs?</b>
<b>Description</b>	Returns the current list of requested channels
<b>Parameter</b>	None
<b>Explanation</b>	
<b>Return Format</b>	<ASCII String>[, <ASCII String>[, ...]]   NONE
<b>Example</b>	<pre>-&gt; :ELOG:ITEMs? &lt;- "U1_tRMS@PowerGroup", "AI 1/2"</pre>

## 11.3 :ELOG:PERiod <Duration>

<b>Syntax</b>	<b>:ELOG:PERiod &lt;Duration&gt;</b>											
<b>Description</b>	Sets the duration (in seconds) of the data aggregation time of statistics from source channels											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;duration&gt;</td> <td>NRf</td> <td>&gt; 0</td> <td>0.1 seconds</td> </tr> </tbody> </table>				Name	Type	Range	Default	<duration>	NRf	> 0	0.1 seconds
Name	Type	Range	Default									
<duration>	NRf	> 0	0.1 seconds									
<b>Explanation</b>	The duration period is the number of seconds where statistics values such as MIN/AVG/... are collected from the source channels (ITEMS). The minimum duration is limited by the frequency of the source channel. Values below the inverse of the source channel frequency will lead to an error state											
<b>Example</b>	Set 2.5Hz (0.4) or 1kHz (0.001) minimum source channel frequency <pre>-&gt; :ELOG:PERiod 0.4 ... -&gt; :ELOG:PERiod 0.001</pre>											

## 11.4 :ELOG:PERiod?

<b>Syntax</b>	<b>:ELOG:PERiod?</b>
<b>Description</b>	Queries the numeric data aggregation time for output (in seconds)
<b>Parameter</b>	None
<b>Explanation</b>	
<b>Return Format</b>	<NR2>
<b>Example</b>	<pre>-&gt; :ELOG:PER 0.5 -&gt; :ELOG:PER? &lt;- 0.5</pre>

## 11.5 :ELOG:CALCulations {AVG|MIN|MAX|RMS}[, {AVG|MIN|MAX|RMS}[,...]]

<b>Syntax</b>	<b>:ELOG:CALCulations {AVG MIN MAX RMS}[, {AVG MIN MAX RMS}[,...]]</b>											
<b>Description</b>	Sets a list of requested statistical calculations for all channels											
<b>Parameter</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Name</th> <th style="width: 20%;">Type</th> <th style="width: 40%;">Range</th> <th style="width: 20%;">Default</th> </tr> </thead> <tbody> <tr> <td>Modes</td> <td>ASCII literal(s)</td> <td> <ul style="list-style-type: none"> <li>• AVG Average values</li> <li>• MIN Minimum values</li> <li>• MAX Maximum values</li> <li>• RMS root-mean-square values</li> </ul> </td> <td>AVG</td> </tr> </tbody> </table>				Name	Type	Range	Default	Modes	ASCII literal(s)	<ul style="list-style-type: none"> <li>• AVG Average values</li> <li>• MIN Minimum values</li> <li>• MAX Maximum values</li> <li>• RMS root-mean-square values</li> </ul>	AVG
	Name	Type	Range	Default								
Modes	ASCII literal(s)	<ul style="list-style-type: none"> <li>• AVG Average values</li> <li>• MIN Minimum values</li> <li>• MAX Maximum values</li> <li>• RMS root-mean-square values</li> </ul>	AVG									
<b>Explanation</b>	One or multiple calculations are performed for each ITEMS channel. When fetching data, calculations are returned in the same order as requested in CALC.											
<b>Example</b>	<pre>-&gt; :ELOG:CALCulations AVG,MAX,RMS</pre>											

## 11.6 :ELOG:CALCulations?

### 11.7 :ELOG:PERiod?

<b>Syntax</b>	<b>:ELOG:PERiod?</b>
<b>Description</b>	Queries the list of configured calculation modes
<b>Parameter</b>	None
<b>Explanation</b>	
<b>Return Format</b>	{AVG MIN MAX RMS}[,{AVG MIN MAX RMS}[,...]]
<b>Example</b>	<pre>-&gt; :ELOG:CALC MIN,MAX -&gt; :ELOG:CALC? &lt;- MIN,MAX</pre>

### 11.8 :ELOG:FORMat {ASCII|BIN\_INTEL|BIN\_MOTOROLA}

<b>Syntax</b>	<b>:ELOG:FORMat {ASCII BIN_INTEL BIN_MOTOROLA}</b>											
<b>Description</b>	Sets the requested output format											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>Format</td> <td>ASCII literal(s)</td> <td> <ul style="list-style-type: none"> <li>• ASCII</li> <li>• BIN_INTEL</li> <li>• BIN_MOTOROLA</li> </ul> </td> <td>ASCII</td> </tr> </tbody> </table>				Name	Type	Range	Default	Format	ASCII literal(s)	<ul style="list-style-type: none"> <li>• ASCII</li> <li>• BIN_INTEL</li> <li>• BIN_MOTOROLA</li> </ul>	ASCII
	Name	Type	Range	Default								
Format	ASCII literal(s)	<ul style="list-style-type: none"> <li>• ASCII</li> <li>• BIN_INTEL</li> <li>• BIN_MOTOROLA</li> </ul>	ASCII									
<b>Explanation</b>	<p>When format is set to ASCII, all numbers are returned in the NR3 format (timestamps in NR2 format)</p> <p>When format is set to BIN_INTEL or BIN_MOTOROLA, all values are returned in IEEE 754 float32 formats (in the corresponding byte order). See FETCH for more details about record layouts</p>											
<b>Example</b>	<pre>-&gt; :ELOG:FORM BIN_INTEL</pre>											

## 11.9 :ELOG:FORMat?

<b>Syntax</b>	<b>:ELOG:FORMat?</b>
<b>Description</b>	Queries the list of configured calculation modes
<b>Parameter</b>	None
<b>Explanation</b>	Returns the currently configured output format
<b>Return Format</b>	{ASCII BIN_INTEL BIN_MOTOROLA}
<b>Example</b>	<pre>-&gt; :ELOG:FORM? -&gt; :ELOG:FORM? &lt;- ASCII</pre>

## 11.10 :ELOG:TIMestamp {OFF|REL|ABS|ELOG}

<b>Syntax</b>	<b>:ELOG:TIMestamp {OFF REL ABS ELOG}</b>											
<b>Description</b>	Sets the requested timestamp format											
<b>Parameter</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Name</th> <th style="width: 20%;">Type</th> <th style="width: 40%;">Range</th> <th style="width: 20%;">Default</th> </tr> </thead> <tbody> <tr> <td>Mode</td> <td>ASCII literal(s)</td> <td> <ul style="list-style-type: none"> <li>• OFF No timestamp</li> <li>• REL Seconds since acquisition start</li> <li>• ABS Absolute timestamp as ISO date/time</li> <li>• ELOG Seconds since the current ELOG session was started</li> </ul> </td> <td>ASCII</td> </tr> </tbody> </table>				Name	Type	Range	Default	Mode	ASCII literal(s)	<ul style="list-style-type: none"> <li>• OFF No timestamp</li> <li>• REL Seconds since acquisition start</li> <li>• ABS Absolute timestamp as ISO date/time</li> <li>• ELOG Seconds since the current ELOG session was started</li> </ul>	ASCII
	Name	Type	Range	Default								
Mode	ASCII literal(s)	<ul style="list-style-type: none"> <li>• OFF No timestamp</li> <li>• REL Seconds since acquisition start</li> <li>• ABS Absolute timestamp as ISO date/time</li> <li>• ELOG Seconds since the current ELOG session was started</li> </ul>	ASCII									
<b>Explanation</b>	Enables or disables timestamp reporting when requesting data through FETCH. Relative times (REL) are reported as seconds since acquisition start, ELOG times are seconds since the first sample from the current ELOG session (after calling :ELOG:START). Absolute times are only reported in ASCII output format and have the following format: YYYY-MM-DDThh:mm:ss.xxxxxx											
<b>Example</b>	-> :ELOG:TIM REL											



### 11.11 :ELOG:TIMestamp?

<b>Syntax</b>	<b>:ELOG:TIMestamp?</b>
<b>Description</b>	Queries the currently configured timestamp format
<b>Parameter</b>	None
<b>Explanation</b>	
<b>Return Format</b>	{OFF REL ABS ELOG}
<b>Example</b>	-> :ELOG:TIM REL -> :ELOG:TIM? <- REL

### 11.12 :ELOG:START

<b>Syntax</b>	<b>:ELOG:START</b>
<b>Description</b>	Initializes and starts ELOG data gathering. Samples values can be fetched as early as one aggregation period after START. After calling START, no ELOG configuration is possible.
<b>Parameter</b>	None
<b>Explanation</b>	
<b>Example</b>	-> :ELOG:START



### 11.13 :ELOG:FETCh? [<NUM>]

<b>Syntax</b>	<b>:ELOG:FETCh [&lt;max_records&gt;]</b>																		
<b>Description</b>	Fetches MAX_RECORDS records or less from the internal ELOG buffer. If no parameter is given, all available records are returned. FETCh is only possible after START.																		
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>Max_records</td> <td>NR1</td> <td>1 .. MAX</td> <td>MAX</td> </tr> </tbody> </table>				Name	Type	Range	Default	Max_records	NR1	1 .. MAX	MAX							
Name	Type	Range	Default																
Max_records	NR1	1 .. MAX	MAX																
<b>Explanation</b>	<p>Fetch returns values at record level granularity. There is one record for each timestamp. Each record consists of values from all channels (ITEMs) and all calculation modes (CALC). If no data is available, NONE is returned. ERROR indicates an error.</p> <p>In ASCII mode, the result is a list of NRf values, all values of one record are returned consecutively, possibly prepended by the timestamp value (see example).</p> <p>In binary mode, for every channel-calculation, a separate Arbitrary Block with up to MAX_RECORDS float32 values is returned. When using timestamps, an Arbitrary Block with float32 timestamp values is returned as the first result.</p> <p>In the following examples, we assume the following settings, which configure a record of four values:</p> <pre>-&gt; :ELOG:ITEMs "CH0", "CH1" -&gt; :ELOG:CALC AVG, MIN</pre>																		
<b>Example</b>	<table border="1"> <thead> <tr> <th>Timestamp</th> <th>CH0.AVG</th> <th>CH0.MIN</th> <th>CH1.AVG</th> <th>CH1.MIN</th> </tr> </thead> <tbody> <tr> <td>0.1</td> <td>1.5</td> <td>1</td> <td>10.5</td> <td>10</td> </tr> <tr> <td>0.2</td> <td>2.5</td> <td>2</td> <td>20.5</td> <td>20</td> </tr> </tbody> </table> <pre>-&gt; :ELOG:FORM ASCII; :ELOG:TIM OFF; :ELOG:START -&gt; :ELOG:FETCh? 2 &lt;- 1.5, 1, 10.5, 10, 2.5, 2, 20.5, 20 -&gt; :ELOG:STOP ... -&gt; :ELOG:FORM ASCII; :ELOG:TIM REL; :ELOG:START -&gt; :ELOG:FETCh? 2 &lt;- 0.1, 1.5, 1, 10.5, 10, 0.2, 2.5, 2, 20.5, 20 -&gt; :ELOG:STOP ... -&gt; :ELOG:FORM BIN_INTEL; :ELOG:TIM REL; :ELOG:START -&gt; :ELOG:FETCh? 2 &lt;- {0.1, 0.2}, {1.5, 2.5}, {1, 2}, {10.5, 20.5}, {10, ↪ 20} -&gt; :ELOG:STOP</pre>				Timestamp	CH0.AVG	CH0.MIN	CH1.AVG	CH1.MIN	0.1	1.5	1	10.5	10	0.2	2.5	2	20.5	20
Timestamp	CH0.AVG	CH0.MIN	CH1.AVG	CH1.MIN															
0.1	1.5	1	10.5	10															
0.2	2.5	2	20.5	20															

## 11.14 :ELOG:STOP

<b>Syntax</b>	<b>:ELOG:STOP</b>
<b>Description</b>	Stops buffering data after a previous START call, clears the data buffer and disables fetching.
<b>Parameter</b>	None
<b>Explanation</b>	The configuration is not changed; therefore, data acquisition can be restarted by calling START again.
<b>Example</b>	-> :ELOG:STOP

## 11.15 :ELOG:RESet

<b>Syntax</b>	<b>:ELOG:RESet</b>
<b>Description</b>	Resets the ELOG configuration and stops fetching if started with START. All parameters are set to their default values and the channel list (ITEMs) is cleared.
<b>Parameter</b>	None
<b>Explanation</b>	Same behavior as when calling *RST, but limited to the ELOG subsystem.
<b>Example</b>	-> :ELOG:RESet

## 11.16 :ELOG:STATE?

<b>Syntax</b>	<b>:ELOG:STATE?</b>
<b>Description</b>	Returns the internal state of the ELOG subsystem. This command is not needed for regular operations but is useful for information purposes. Possible states are: CONFIG, RUNNING and INVALID
<b>Parameter</b>	None
<b>Explanation</b>	The ELOG subsystem uses states internally. Initially, ELOG is in configuration state (CONFIG) which accepts commands like ITEMS, PER, CALC, FORM and TIM. After calling START, the subsystem is switched into a data gathering state (RUNNING) which no longer allows to modify the configuration. The configuration state can be entered again by either calling STOP or RESet. While fetching data from Oxygen through ELOG, the measurement configuration in Oxygen must not be changed. If a change of the configuration is detected, ELOG will stop fetching data and an error is reported when calling FETCH. The state query will return the INVALID state in this case
<b>Return Format</b>	ASCII literal
<b>Example</b>	-> :ELOG:STATE? <- RUNNING

## DATA STREAMING (DSTREAM)

The DSTREAM subsystem allows the SCPI user to configure and control fast data streaming via a TCP network protocol. This documentation describes the SCPI commands to control the subsystem. Details about the TCP transfer and examples can be found in a separate documentation.

The following SCPI commands can be used to configure and control one or more streaming groups (distinguished via a GRP number). Each group has its own list of channels and a TCP port where data is provided. Each group can be configured and started individually. Some commands provide an ALL option to start/stop all configured groups.

Each streaming group has its own internal state. It can be queried using the STATE? query. The following list describes the allowed operations in each state:

**INVALID:** An invalid state means the streaming group does not exist, it will be automatically created when calling configuration commands such as ITEMS and PORT

**CONFIG:** In the configuration state, ITEMS and PORT commands can be used to configure the group. The INIT command will initialize the subsystem.

**INITIALIZED:** The streaming group is initialized (e.g. has an open TCP port) and is waiting for connections and the START command. If a configuration command such as ITEMS or PORT is executed, the state is set back to CONFIG.

**RUNNING:** The streaming group is actively sending data. Call STOP to enter the INITIALIZED state again.

**ERROR:** If one of the channels to register is unused, the whole dataset will be discarded and the DSTREAM system will be in an error state. Use `:DStream:STATE?` to get more details. It can be reset using the RESet command.

## 12.1 :DStream:ITEMs[<GRP>] <channel>[,<channel>[,...]]

<b>Syntax</b>	<b>:DStream:ITEMs[&lt;GRP&gt;] &lt;channel&gt;[,&lt;channel&gt; [,...]]</b>															
<b>Description</b>	Defines the ordered list of requested channels for one stream group GRP															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;GRP&gt;</td> <td>NR1</td> <td>Group number &gt;= 1</td> <td>1</td> </tr> <tr> <td>&lt;channel&gt;</td> <td>ASCII String</td> <td>Oxygen Channel Name</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<GRP>	NR1	Group number >= 1	1	<channel>	ASCII String	Oxygen Channel Name	None
	Name	Type	Range	Default												
	<GRP>	NR1	Group number >= 1	1												
<channel>	ASCII String	Oxygen Channel Name	None													
<b>Explanation</b>	<p>The user can specify a list of multiple channels. If one channel does not exist, is invalid or disabled, no channel is set for the streaming group (see system error log for more details). In addition, the following channel types are not supported: Digital channels, video channels and Rosette group channels.  </p>															
<b>Example</b>	<p>Set two channels for stream group 2: AI 1/1 and AI 1/2        -&gt; :DST:ITEMs2 "AI 1/1", "AI 1/2"</p>															

## 12.2 :DStream:ITEMs[<GRP>]?

<b>Syntax</b>	<b>:DStream:ITEMs[&lt;GRP&gt;]?</b>											
<b>Description</b>	Returns the current list of requested channels for one channel group GRP											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;GRP&gt;</td> <td>NR1</td> <td>Group number &gt;= 1</td> <td>1</td> </tr> </tbody> </table>				Name	Type	Range	Default	<GRP>	NR1	Group number >= 1	1
	Name	Type	Range	Default								
	<GRP>	NR1	Group number >= 1	1								
<b>Explanation</b>	Returns NONE if no items are set											
<b>Return Format</b>	<ASCII String>[, <ASCII String>[, ...]]   NONE											
<b>Example</b>	<p>Query the items of stream group 2:        -&gt; :DST:ITEMs2?        &lt;- "AI 1/1", "AI 1/2"</p>											

### 12.3 :DStream:PORT[<GRP>] <PORT>

<b>Syntax</b>	:DStream:PORT[<GRP>] <PORT>															
<b>Description</b>	Configures the port number of the TCP server for a streaming group GRP															
<b>Parameter</b>	<table border="1" style="width: 100%;"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;GRP&gt;</td> <td>NR1</td> <td>Group number &gt;= 1</td> <td>1</td> </tr> <tr> <td>&lt;PORT&gt;</td> <td>NR1</td> <td>1 .. 65536</td> <td>10003</td> </tr> </tbody> </table>				Name	Type	Range	Default	<GRP>	NR1	Group number >= 1	1	<PORT>	NR1	1 .. 65536	10003
	Name	Type	Range	Default												
	<GRP>	NR1	Group number >= 1	1												
	<PORT>	NR1	1 .. 65536	10003												
<b>Explanation</b>	The port must be a valid TCP port. It must not be used by any other TCP server on the local system but can be shared by multiple streaming groups of the OXYGEN instance.															
<b>Example</b>	Set the TCP port for stream group 2 to 10005: -> :DST:PORT2 10005															

### 12.4 :DStream:PORT[<GRP>]?

<b>Syntax</b>	:DStream:PORT[<GRP>]?											
<b>Description</b>	Queries the currently configured TCP port											
<b>Parameter</b>	<table border="1" style="width: 100%;"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;GRP&gt;</td> <td>NR1</td> <td>Group number &gt;= 1</td> <td>1</td> </tr> </tbody> </table>				Name	Type	Range	Default	<GRP>	NR1	Group number >= 1	1
	Name	Type	Range	Default								
	<GRP>	NR1	Group number >= 1	1								
<b>Explanation</b>	Returns the NR1 numeric value of the configured TCP port											
<b>Return Format</b>	<Nr1>											
<b>Example</b>	-> :DST:PORT2? <- 10005											



## 12.5 :DStream:INIT [<GRP> | ALL]

<b>Syntax</b>	<b>:DStream:INIT [&lt;GRP&gt;   ALL]</b>															
<b>Description</b>	Initializes one or all data streaming groups and opens the TCP port															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;GRP&gt;</td> <td>NR1</td> <td>Group number &gt;= 1</td> <td>1</td> </tr> <tr> <td>ALL</td> <td>Literal</td> <td></td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<GRP>	NR1	Group number >= 1	1	ALL	Literal		None
	Name	Type	Range	Default												
	<GRP>	NR1	Group number >= 1	1												
	ALL	Literal		None												
<b>Explanation</b>	The server will listen for new connection on the configured port															
<b>Example</b>	Initializes the streaming group 1: -> :DST:INIT 1															

## 12.6 :DStream:START [<GRP> | ALL]

<b>Syntax</b>	<b>:DStream:START [&lt;GRP&gt;   ALL]</b>															
<b>Description</b>	Starts streaming of one or all data streaming groups															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;GRP&gt;</td> <td>NR1</td> <td>Group number &gt;= 1</td> <td>1</td> </tr> <tr> <td>ALL</td> <td>Literal</td> <td></td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<GRP>	NR1	Group number >= 1	1	ALL	Literal		None
	Name	Type	Range	Default												
	<GRP>	NR1	Group number >= 1	1												
	ALL	Literal		None												
<b>Explanation</b>	Only possible after calling INIT for the specified group(s).															
<b>Example</b>	Start streaming for all configured streaming groups: -> :DST:START ALL															

## 12.7 :DStream:STOP [<GRP> | ALL]

<b>Syntax</b>	<b>:DStream:STOP [&lt;GRP&gt;   ALL]</b>															
<b>Description</b>	Stops streaming of one or all data streaming groups															
<b>Parameter</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Name</th> <th style="width: 25%;">Type</th> <th style="width: 25%;">Range</th> <th style="width: 25%;">Default</th> </tr> </thead> <tbody> <tr> <td>&lt;GRP&gt;</td> <td>NR1</td> <td>Group number &gt;= 1</td> <td>1</td> </tr> <tr> <td>ALL</td> <td>Literal</td> <td></td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<GRP>	NR1	Group number >= 1	1	ALL	Literal		None
	Name	Type	Range	Default												
	<GRP>	NR1	Group number >= 1	1												
	ALL	Literal		None												
<b>Explanation</b>	Only possible after calling START for the specified group(s). After stopping, the streaming group is in INITIALIZED state.															
<b>Example</b>	Stop streaming for all configured streaming groups: -> :DST:STOP ALL															

## 12.8 :DStream:DELeTe [<GRP> | ALL]

<b>Syntax</b>	<b>:DStream:DELeTe [&lt;GRP&gt;   ALL]</b>															
<b>Description</b>	Deletes a configured streaming group															
<b>Parameter</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Name</th> <th style="width: 25%;">Type</th> <th style="width: 25%;">Range</th> <th style="width: 25%;">Default</th> </tr> </thead> <tbody> <tr> <td>&lt;GRP&gt;</td> <td>NR1</td> <td>Group number &gt;= 1</td> <td>1</td> </tr> <tr> <td>ALL</td> <td>Literal</td> <td></td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<GRP>	NR1	Group number >= 1	1	ALL	Literal		None
	Name	Type	Range	Default												
	<GRP>	NR1	Group number >= 1	1												
	ALL	Literal		None												
<b>Explanation</b>	The streaming group must be in config state. All ITEMS and port settings are deleted.															
<b>Example</b>	Delete streaming group 1: -> :DST:DEL 1															

## 12.9 : DStream:RESet

<b>Syntax</b>	<b>:DStream:RESet</b>
<b>Description</b>	Resets the data streaming subsystem
<b>Parameter</b>	None
<b>Explanation</b>	Stops the server if started, and resets the configuration to its default settings
<b>Example</b>	-> :DST:RES

## 12.10 : DStream:STATe[<GRP>]?

<b>Syntax</b>	<b>:DStream:STATe[&lt;GRP&gt;]?</b>								
<b>Description</b>	Queries the internal state of the data streaming subsystem								
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;GRP&gt;</td> <td>NR1</td> <td>Group number &gt;= 1</td> <td>1</td> </tr> </tbody> </table>	Name	Type	Range	Default	<GRP>	NR1	Group number >= 1	1
Name	Type	Range	Default						
<GRP>	NR1	Group number >= 1	1						
<b>Explanation</b>	<p><b>Possible states are:</b></p> <ul style="list-style-type: none"> <li>INVALID: Streaming group does not exist</li> <li>CONFIG: Configuration state (allows settings of ITEMS and PORT)</li> <li>INITIALIZED: The streaming group has already been initialized</li> <li>RUNNING: The server is started and cannot be configured anymore</li> <li>ERROR: An error has occurred; the user needs to call RESet</li> <li>UNLICENSED: A valid license for the data streaming subsystem was not found</li> </ul>								
<b>Return Format</b>	<ASCII String>[, <ASCII String>[, ...]]   NONE								
<b>Example</b>	<p>Query the state of the streaming group 1 (default value):</p> <pre>-&gt; :DST:STATe? &lt;- CONFIG</pre>								

### 12.11 : DStream:TRIG[<GRP>] {ON|OFF}

<b>Syntax</b>	<b>:DStream:TRIG[&lt;GRP&gt;] {ON OFF}</b>															
<b>Description</b>	Configure a streaming group to send data only when a trigger is active															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;GRP&gt;</td> <td>NR1</td> <td>Group number &gt;= 1</td> <td>1</td> </tr> <tr> <td>{ON OFF}</td> <td>Boolean</td> <td>{ON OFF}</td> <td>OFF</td> </tr> </tbody> </table>				Name	Type	Range	Default	<GRP>	NR1	Group number >= 1	1	{ON OFF}	Boolean	{ON OFF}	OFF
	Name	Type	Range	Default												
	<GRP>	NR1	Group number >= 1	1												
	{ON OFF}	Boolean	{ON OFF}	OFF												
<b>Explanation</b>	When this property is enabled for a streaming group, data will only be streamed during a triggered measurement, while any waveform recording trigger is active.															
<b>Example</b>	Enable the triggered mode of streaming group 1 (default value): -> :DST:TRIG ON															

### 12.12 : DStream:TRIG[<GRP>]?

<b>Syntax</b>	<b>:DStream:TRIG[&lt;GRP&gt;]?</b>											
<b>Description</b>	Query the triggered property for a given streaming group											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;GRP&gt;</td> <td>NR1</td> <td>Group number &gt;= 1</td> <td>1</td> </tr> </tbody> </table>				Name	Type	Range	Default	<GRP>	NR1	Group number >= 1	1
	Name	Type	Range	Default								
	<GRP>	NR1	Group number >= 1	1								
<b>Explanation</b>	<p><b>Returns the current state of the triggered property for a stream group, possible values are</b></p> <ul style="list-style-type: none"> <li>• ON (triggered mode is active)</li> <li>• OFF (data is streamed continuously)</li> </ul>											
<b>Example</b>	Query the triggered mode of streaming group 1 (default value): -> :DST:TRIG? <- ON											

### 12.13 :DStream:REPLAY[<GRP>] {LIVE | BULK}

<b>Syntax</b>	<b>:DStream:REPLAY {LIVE   BULK}</b>															
<b>Description</b>	Sets the datasream REPLAY state															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;GRP&gt;</td> <td>NR1</td> <td>Group number &gt;= 1</td> <td>1</td> </tr> <tr> <td>{LIVE   BULK}</td> <td>Literal</td> <td>{LIVE BULK}</td> <td>BULK</td> </tr> </tbody> </table>				Name	Type	Range	Default	<GRP>	NR1	Group number >= 1	1	{LIVE   BULK}	Literal	{LIVE BULK}	BULK
	Name	Type	Range	Default												
	<GRP>	NR1	Group number >= 1	1												
	{LIVE   BULK}	Literal	{LIVE BULK}	BULK												
<b>Explanation</b>	Sets the data streaming REPLAY mode. Possible values are LIVE or BULK															
<b>Example</b>	<pre>-&gt; : DStream:REPLAY BULK -&gt; : DStream:REPLAY? &lt;- BULK</pre>															

### 12.14 : DStream:REPLAY[<GRP>]?

<b>Syntax</b>	<b>:DStream:REPLAY?</b>											
<b>Description</b>	Query the dstream REPLAY state											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;GRP&gt;</td> <td>NR1</td> <td>Group number &gt;= 1</td> <td>1</td> </tr> </tbody> </table>				Name	Type	Range	Default	<GRP>	NR1	Group number >= 1	1
	Name	Type	Range	Default								
	<GRP>	NR1	Group number >= 1	1								
<b>Explanation</b>	<p><b>Returns the current REPLAY state for a stream group, possible values are</b></p> <ul style="list-style-type: none"> <li>• LIVE (data streaming as in live mode)</li> <li>• BULK (streaming continuously as fast as possible)</li> </ul>											
<b>Example</b>	<pre>-&gt; :DST:REPLAY? &lt;- LIVE</pre>											

### 12.15 :DStream:INTERVAL[<GRP>] <INTV>

<b>Syntax</b>	<b>:DStream:INTERVAL &lt;INTV&gt;</b>															
<b>Description</b>	Configure a streaming group to send data blocks with a given time interval															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;GRP&gt;</td> <td>NR1</td> <td>Group number &gt;= 1</td> <td>1</td> </tr> <tr> <td>INTV</td> <td>NR1</td> <td>100 .. 1000</td> <td>100</td> </tr> </tbody> </table>				Name	Type	Range	Default	<GRP>	NR1	Group number >= 1	1	INTV	NR1	100 .. 1000	100
	Name	Type	Range	Default												
	<GRP>	NR1	Group number >= 1	1												
	INTV	NR1	100 .. 1000	100												
<b>Explanation</b>	When this property is set for a streaming group, data will be sent in blocks of the given time interval.															
<b>Example</b>	<pre>-&gt; : DStream:INTERVAL 200 -&gt; : DStream:INTERVAL? &lt;- 200</pre>															

### 12.16 :DStream:INTERVAL[<GRP>]?

<b>Syntax</b>	<b>:DStream:INTERVAL?</b>											
<b>Description</b>	Query the interval property for a given streaming group											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;GRP&gt;</td> <td>NR1</td> <td>Group number &gt;= 1</td> <td>1</td> </tr> </tbody> </table>				Name	Type	Range	Default	<GRP>	NR1	Group number >= 1	1
	Name	Type	Range	Default								
	<GRP>	NR1	Group number >= 1	1								
<b>Explanation</b>	Returns the current value of the interval property for a stream group, possible values are between 100 and 1000 milliseconds.											
<b>Example</b>	<pre>-&gt; :DST:INTERVAL? &lt;- 100</pre>											



## EXPORT COMMANDS

### 13.1 :EXP:DIRectory

<b>Syntax</b>	<b>:EXP:DIRectory?</b>
<b>Description</b>	Queries the current export folder
<b>Parameter</b>	None
<b>Explanation</b>	Returns the current directory of the OXYGEN export files. If the current folder is empty, the OXYGEN export default directory is used.
<b>Return Format</b>	String
<b>Example</b>	<pre>-&gt; :EXP:DIR? &lt;- :EXP:DIR "d:/temp"</pre>

### 13.2 :EXP:DIRectory "path"

<b>Syntax</b>	<b>:EXP:DIRectory "path"</b>											
<b>Description</b>	Sets the current export folder											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;path&gt;</td> <td>ASCII String</td> <td></td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<path>	ASCII String		None
Name	Type	Range	Default									
<path>	ASCII String		None									
<b>Explanation</b>	Sets the OXYGEN export folder. An error code is set if the input parameter is wrong or ill formed.											
<b>Example</b>	<pre>-&gt; :EXP:DIR "c:/temp" -&gt; :EXP:DIR? &lt;- :EXP:DIR "c:/temp"</pre>											



### 13.3 :EXPort:AUTO?

<b>Syntax</b>	<b>:EXPort:AUTO?</b>
<b>Description</b>	Queries the auto export flag
<b>Parameter</b>	None
<b>Explanation</b>	Returns the current OXYGEN export flag.
<b>Return Format</b>	<Boolean>
<b>Example</b>	<pre>-&gt; :EXP:AUTO? &lt;- :EXP:AUTO ON</pre>

### 13.4 :EXPort:AUTO {ON|OFF}

<b>Syntax</b>	<b>:EXPort:AUTO {ON OFF}</b>											
<b>Description</b>	Sets the OXYGEN auto export flag.											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>switch</td> <td>Literal</td> <td> <ul style="list-style-type: none"> <li>• ON: enables auto export</li> <li>• OFF: disables auto export</li> </ul> </td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	switch	Literal	<ul style="list-style-type: none"> <li>• ON: enables auto export</li> <li>• OFF: disables auto export</li> </ul>	None
Name	Type	Range	Default									
switch	Literal	<ul style="list-style-type: none"> <li>• ON: enables auto export</li> <li>• OFF: disables auto export</li> </ul>	None									
<b>Explanation</b>	<p>Enables/disables the OXYGEN auto export. When enabled OXYGEN exports selected data to the export folder, after the recording stop event (see :STORE:STOP command above).</p> <p>An error code is set if input parameter is wrong or ill formed-</p>											
<b>Example</b>	<pre>-&gt; :EXP:AUTO ON -&gt; :EXP:AUTO? &lt;- :EXP:AUTO ON</pre>											

## 13.5 :EXPort:ITEMS Commands and Queries

### 13.6 :EXPort:ITEMS[:LIST]?

<b>Syntax</b>	<b>:EXPort:ITEMS[:LIST]?</b>
<b>Description</b>	Queries the selected channels for export
<b>Parameter</b>	None
<b>Explanation</b>	Returns a list of elements alternating the channel id and the channel name. Note: this is the default query in the export items subsystem (see examples)
<b>Return Format</b>	(<String>, <String>)[,<String>, <String>][,...]   NONE
<b>Example</b>	<pre> -&gt; :EXPort:ITEMS:LIST? &lt;- ("12942219420566028311", "AI 1/1 Sim"),     ↪ ("12942219420566028312", "AI 1/2 Sim") ... -&gt; :EXPort:ITEMS?                               //use as ↪     ↪ default query &lt;- ("12942219420566028311", "AI 1/1 Sim"),     ↪ ("12942219420566028312", "AI 1/2 Sim") </pre>

### 13.7 :EXPort:ITEMS[:LIST] <channel>[,<channel>[,...]]

<b>Syntax</b>	<b>:EXPort:ITEMS[:LIST] &lt;channel&gt;[,&lt;channel&gt;[,...]]</b>											
<b>Description</b>	Setup the export selected channels list											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;channel&gt;</td> <td>ASCII String</td> <td>Oxygen Long Channel Name or Channel ID</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<channel>	ASCII String	Oxygen Long Channel Name or Channel ID	None
	Name	Type	Range	Default								
	<channel>	ASCII String	Oxygen Long Channel Name or Channel ID	None								
<b>Explanation</b>	The user can specify a list of multiple channels to set as selected export channels. If one channel does not exist, is invalid or disabled, no channel is set for the export (see system error log for more details).											
<b>Example</b>	<pre>Set two channels for export selection list: AI 1/1 and AI 1/2 -&gt; :EXPort:ITEMS:LIST "AI 1/1 Sim",    ↪ "14597010808647974936" -&gt; :EXPort:ITEMS:LIST? &lt;- ("14597010808647974935", "AI 1/1 Sim"),    ↪ ("14597010808647974936", "AI 1/2 Sim") ... -&gt; :EXPort:ITEMS "AI 1/3 Sim", "AI 1/4 Sim"    ↪ //use as default query -&gt; :EXPort:ITEMS? &lt;- ("14597010808647974937", "AI 1/3 Sim"),    ↪ ("14597010808647974938", "AI 1/4 Sim")</pre>											

### 13.8 :EXPort:ITEMS:AVAI?

<b>Syntax</b>	<b>:EXPort:ITEMS:AVAI?</b>
<b>Description</b>	Queries all channels which are available for export
<b>Parameter</b>	None
<b>Explanation</b>	Returns a list of elements alternating the channel id and the channel name
<b>Return Format</b>	(<String>, <String>)[,<String>, <String>[,...]]   NONE
<b>Example</b>	<pre>-&gt; :EXPort:ITEMS:AVAI? &lt;- ("14597010808647974935", "AI 1/1 Sim"),    ↪ ("14597010808647974936", "AI 1/2 Sim"),    ↪ ("14597010808647974937", "AI 1/3 Sim")...</pre>

### 13.9 :EXPort:ITEMS:CLEAr

<b>Syntax</b>	<b>:EXPort:ITEMS:CLEAr</b>
<b>Description</b>	Clear export channels list
<b>Parameter</b>	None
<b>Explanation</b>	Remove all selected export channels from the export channels list
<b>Example</b>	<pre> -&gt; :EXPort:ITEMS? &lt;- ("14597010808647974937", "AI 1/3 Sim"),     ↪ ("14597010808647974938", "AI 1/4 Sim") -&gt; :EXPort:ITEMS:CLEAr -&gt; :EXPort:ITEMS? &lt;- NONE         </pre>

### 13.10 :EXPort:ITEMS:ADD <channel>[,<channel>[,...]]

<b>Syntax</b>	<b>:EXPort:ITEMS:ADD &lt;channel&gt;[,&lt;channel&gt;[,...]]</b>											
<b>Description</b>	Adds one or more channels to the export channels list											
<b>Parameter</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">Name</th> <th style="width: 25%;">Type</th> <th style="width: 25%;">Range</th> <th style="width: 25%;">Default</th> </tr> </thead> <tbody> <tr> <td>&lt;channel&gt;</td> <td>ASCII String</td> <td>Oxygen Long Channel Name or Channel ID</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<channel>	ASCII String	Oxygen Long Channel Name or Channel ID	None
Name	Type	Range	Default									
<channel>	ASCII String	Oxygen Long Channel Name or Channel ID	None									
<b>Explanation</b>	<p>The user can specify a list of multiple channels to add these to the selected export channels.</p> <p>If one channel does not exist, is invalid or disabled, no channel is added to the export channels list (see system error log for more details).</p>											
<b>Example</b>	<p>Add channel "AI 1/1 Sim" to the export selection list</p> <pre> -&gt; :EXPort:ITEMS:LIST? &lt;- ("8709117245814472727", "AI 1/1 Sim") -&gt; :EXPort:ITEMS:ADD "AI 1/2 Sim" -&gt; :EXPort:ITEMS? &lt;- ("8709117245814472727", "AI 1/1 Sim"),     ↪ ("8709117245814472728", "AI 1/2 Sim")         </pre>											

### 13.11 :EXPort:ITEMS:DELeTe <channel>[,<channel>[,...]]

<b>Syntax</b>	<b>:EXPort:ITEMS:DELeTe &lt;channel&gt;[,&lt;channel&gt;[,...]]</b>											
<b>Description</b>	Deletes one or more channels from the export channels list											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;channel&gt;</td> <td>ASCII String</td> <td>Oxygen Long Channel Name or Channel ID</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<channel>	ASCII String	Oxygen Long Channel Name or Channel ID	None
	Name	Type	Range	Default								
	<channel>	ASCII String	Oxygen Long Channel Name or Channel ID	None								
<b>Explanation</b>	<p>The user can specify a list of multiple channels to remove these from the selected export channels.</p> <p>If one channel does not exist, is invalid or disabled, no channel is removed from the channels export list (see system error log for more details).</p>											
<b>Example</b>	<pre>Remove channel "AI 1/1 Sim" from the export selection list -&gt; :EXPort:ITEMS:LIST? &lt;- (("8709117245814472727","AI 1/1 Sim"),     ↪ ("8709117245814472728","AI 1/2 Sim") -&gt; :EXPort:ITEMS:DELeTe "AI 1/1 Sim" -&gt; :EXPort:ITEMS? &lt;- ("8709117245814472728","AI 1/2 Sim")</pre>											

## MARKER COMMANDS

### 14.1 :MARKer:ADD <label>[,<description>|<time>|,<description>,<time>]]

<b>Syntax</b>	<b>:MARKer:ADD label&gt;[,&lt;description&gt; ,&lt;time&gt; ,&lt;description&gt;,&lt;time&gt;]]</b>																			
<b>Description</b>	Add markers to current measurement																			
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;label&gt;</td> <td>ASCII String</td> <td></td> <td>None</td> </tr> <tr> <td>&lt;description&gt;</td> <td>ASCII String</td> <td></td> <td>None</td> </tr> <tr> <td>&lt;time&gt;</td> <td>Nr2</td> <td></td> <td>Current time</td> </tr> </tbody> </table>				Name	Type	Range	Default	<label>	ASCII String		None	<description>	ASCII String		None	<time>	Nr2		Current time
Name	Type	Range	Default																	
<label>	ASCII String		None																	
<description>	ASCII String		None																	
<time>	Nr2		Current time																	
<b>Explanation</b>	Add a single marker at position <TIME> (in seconds from acquisition start) with <LABEL> and description <DESCRIPTION>. If no <TIME>-parameter is specified in the parameter list, the marker is added at the current time.																			
<b>Example</b>	<pre>-&gt; :MARK:ADD "Label 1","Description of Marker 1" -&gt; :MARK:ADD "Label 1",0.0 -&gt; :MARK:ADD "Label 1","Description of Marker 1",1.   ↳2345   &lt;- 0</pre>																			



## SYNCHRONISATION QUERIES

### 15.1 :SYNC:STATE?

<b>Syntax</b>	<b>:SYNC:STATE?</b>
<b>Description</b>	Query the current synchronization hardware state
<b>Parameter</b>	None
<b>Explanation</b>	Returns synchronization hardware state <ul style="list-style-type: none"> <li>• IN_SYNC: hardware is synced (corresponds to the green sync symbol)</li> <li>• SYNCING: hardware is syncing (corresponds to the yellow sync symbol)</li> <li>• OUT_OF_SYNC: current state is not synced (corresponds to the red sync symbol)</li> <li>• SYNC_UNAVAILABLE: internal sync source is used (corresponds to the grey sync symbol)</li> <li>• ERROR: sync validation error or no valid sync hardware found (corresponds to the red sync symbol)</li> </ul>
<b>Return Format</b>	Literal
<b>Example</b>	<pre>-&gt; :SYNC:STATE? &lt;- :SYNC:STATE IN_SYNC</pre>

### 15.2 :SYNC:ENCLOSURES[:LIST]?

<b>Syntax</b>	<b>:SYNC:ENCLOSURES[:LIST]?</b>
<b>Description</b>	Returns the most important information about all enclosures in the system.
<b>Parameter</b>	None
<b>Explanation</b>	Returns Name, Serial number and Nodename for all enclosures.
<b>Return Format</b>	(<String>,<String>,<String>)[, ...]
<b>Example</b>	<pre>-&gt; :SYNC:ENCLOSURES? &lt;- :SYNC:ENCLOSURES ("DEWE3-A4", "8008098", ""),     ↪ ("DEWE3-A4", "12323", "shienar")</pre>



### 15.3 :SYNC:ENClosure[<ENR>]:NAME?

<b>Syntax</b>	<b>:SYNC:ENClosure[&lt;ENR&gt;]:NAME?</b>											
<b>Description</b>	Returns the name of the selected enclosure											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;ENR&gt;</td> <td>NR1</td> <td>Enclosure number in the range [1, enclosure_count]</td> <td>1</td> </tr> </tbody> </table>				Name	Type	Range	Default	<ENR>	NR1	Enclosure number in the range [1, enclosure_count]	1
	Name	Type	Range	Default								
	<ENR>	NR1	Enclosure number in the range [1, enclosure_count]	1								
<b>Explanation</b>	Returns the name.											
<b>Return Format</b>	<String>											
<b>Example</b>	<pre>-&gt; :SYNC:ENC1:NAME? &lt;- :SYNC:ENC1:NAME "DEWE3-A4"</pre>											

### 15.4 :SYNC:ENClosure[<ENR>]:SERIAL?

<b>Syntax</b>	<b>:SYNC:ENClosure[&lt;ENR&gt;]:SERIAL?</b>											
<b>Description</b>	Returns the serial number of the selected enclosure											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;ENR&gt;</td> <td>NR1</td> <td>Enclosure number in the range [1, enclosure_count]</td> <td>1</td> </tr> </tbody> </table>				Name	Type	Range	Default	<ENR>	NR1	Enclosure number in the range [1, enclosure_count]	1
	Name	Type	Range	Default								
	<ENR>	NR1	Enclosure number in the range [1, enclosure_count]	1								
<b>Explanation</b>	Returns the string representation of the serial number											
<b>Return Format</b>	<String>											
<b>Example</b>	<pre>-&gt; :SYNC:ENC1:SERI? &lt;- :SYNC:ENC1:SERI "1234123"</pre>											

### 15.5 :SYNC:ENClosure[<ENR>]:NODEName?

<b>Syntax</b>	<b>:SYNC:ENClosure[&lt;ENR&gt;]:NODEName?</b>											
<b>Description</b>	Returns the name of the parent node of the selected enclosure											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;ENR&gt;</td> <td>NR1</td> <td>Enclosure number in the range [1, enclosure_count]</td> <td>1</td> </tr> </tbody> </table>				Name	Type	Range	Default	<ENR>	NR1	Enclosure number in the range [1, enclosure_count]	1
	Name	Type	Range	Default								
	<ENR>	NR1	Enclosure number in the range [1, enclosure_count]	1								
<b>Explanation</b>	Returns the name of the parent node, or an empty string if the enclosure is owned by the local node.											
<b>Return Format</b>	<String>											
<b>Example</b>	<pre>-&gt; :SYNC:ENC1:NODEN? &lt;- :SYNC:ENC1:NODEN "shienar"</pre>											

### 15.6 :SYNC:ENClosure[<GRP>]:IN[<INR>]:MODE?

<b>Syntax</b>	<b>:SYNC:ENClosure[&lt;GRP&gt;]:IN[&lt;INR&gt;]:MODE?</b>															
<b>Description</b>	Returns the current mode of the synchronization input															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;ENR&gt;</td> <td>NR1</td> <td>Enclosure number in the range [1, enclosure_count]</td> <td>1</td> </tr> <tr> <td>&lt;INR&gt;</td> <td>NR1</td> <td>The only supported value is 1</td> <td>1</td> </tr> </tbody> </table>				Name	Type	Range	Default	<ENR>	NR1	Enclosure number in the range [1, enclosure_count]	1	<INR>	NR1	The only supported value is 1	1
	Name	Type	Range	Default												
	<ENR>	NR1	Enclosure number in the range [1, enclosure_count]	1												
<INR>	NR1	The only supported value is 1	1													
<b>Explanation</b>	Returns the currently selected sync mode for the enclosure															
<b>Return Format</b>	<String>															
<b>Example</b>	<pre>-&gt; :SYNC:ENC1:IN:MODE? &lt;- :SYNC:ENC1:IN:MODE "GPS"</pre>															

## 15.7 :SYNC:ENClosure[<GRP>]:OUTPUTS[:LIST]?

<b>Syntax</b>	<b>:SYNC:ENClosure[&lt;GRP&gt;]:OUTPUTS[:LIST]?</b>											
<b>Description</b>	Returns information about the sync output connectors of the selected enclosure											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;ENR&gt;</td> <td>NR1</td> <td>Enclosure number in the range [1, enclosure_count]</td> <td>1</td> </tr> </tbody> </table>				Name	Type	Range	Default	<ENR>	NR1	Enclosure number in the range [1, enclosure_count]	1
	Name	Type	Range	Default								
	<ENR>	NR1	Enclosure number in the range [1, enclosure_count]	1								
<b>Explanation</b>	Returns the name, connector name and selected mode of the sync output											
<b>Return Format</b>	(<String>,<String>,<String>)[, ...] or NONE if there is no sync connector.											
<b>Example</b>	<pre> -&gt; :SYNC:ENC:OUTPUTS:LIST? &lt;- :SYNC:ENC1:OUTPUTS ("SyncOut2", "SYNC", "TRION/     ↪DEWE2SyncOut"), ("SyncOutAux", "AUX", "None"),     ↪("SyncOutPtp", "PTP", "None")  -&gt; :SYNC:ENC2:OUTPUTS? &lt;- :SYNC:ENC2:OUTPUTS NONE </pre>											

## 15.8 :SYNC:ENClosure[<GRP>]:OUT[<ONR>]:NAME?

<b>Syntax</b>	<b>:SYNC:ENClosure[&lt;GRP&gt;]:OUT[&lt;ONR&gt;]:NAME?</b>															
<b>Description</b>	Returns the name of the specified synchronization output															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;ENR&gt;</td> <td>NR1</td> <td>Enclosure number in the range [1, enclosure_count]</td> <td>1</td> </tr> <tr> <td>&lt;ONR&gt;</td> <td>NR1</td> <td>Index of the sync output port</td> <td>1</td> </tr> </tbody> </table>				Name	Type	Range	Default	<ENR>	NR1	Enclosure number in the range [1, enclosure_count]	1	<ONR>	NR1	Index of the sync output port	1
	Name	Type	Range	Default												
	<ENR>	NR1	Enclosure number in the range [1, enclosure_count]	1												
<ONR>	NR1	Index of the sync output port	1													
<b>Explanation</b>	Returns the name of the sync output															
<b>Return Format</b>	<String>															
<b>Example</b>	<pre> -&gt; :SYNC:ENC1:OUT3:NAME? &lt;- :SYNC:ENC1:OUT3:NAME "SyncOutPtp" </pre>															

### 15.9 :SYNC:ENClosure[<GRP>]:OUT[<ONR>]:CONNector?

<b>Syntax</b>	<b>:SYNC:ENClosure[&lt;GRP&gt;]:OUT[&lt;ONR&gt;]:CONNector?</b>			
<b>Description</b>	Returns the connector name of the specified synchronization output			
<b>Parameter</b>	<b>Name</b>	<b>Type</b>	<b>Range</b>	<b>Default</b>
	<ENR>	NR1	Enclosure number in the range [1, enclosure_count]	1
	<ONR>	NR1	Index of the sync output port	1
<b>Explanation</b>	Returns the name of the sync output connector			
<b>Return Format</b>	<String>			
<b>Example</b>	<pre>-&gt; :SYNC:ENC1:OUT3:CONN? &lt;- :SYNC:ENC1:OUT3:CONN "PTP"</pre>			

### 15.10 :SYNC:ENClosure[<GRP>]:OUT[<ONR>]:MODE?

<b>Syntax</b>	<b>:SYNC:ENClosure[&lt;GRP&gt;]:OUT[&lt;ONR&gt;]:MODE?</b>			
<b>Description</b>	Returns the current mode of the specified synchronization output connector			
<b>Parameter</b>	<b>Name</b>	<b>Type</b>	<b>Range</b>	<b>Default</b>
	<ENR>	NR1	Enclosure number in the range [1, enclosure_count]	1
	<ONR>	NR1	Index of the sync output port	1
<b>Explanation</b>	Returns the currently selected sync mode for the connector			
<b>Return Format</b>	<String>			
<b>Example</b>	<pre>-&gt; :SYNC:ENC1:OUT2:MODE? &lt;- :SYNC:ENC1:OUT2:MODE "None"</pre>			



## MEASUREMENT SCREEN COMMANDS

### 16.1 :SCReen:INSTRuments:OUTputchannel:STARt

<b>Syntax</b>	<b>:SCReen:INSTRuments:OUTputchannel:STARt</b>
<b>Description</b>	Start streaming to analog out channels
<b>Parameter</b>	None
<b>Explanation</b>	Starts streaming on the default stream output instrument if it is in the Paused or Stopped states. The default instrument is the oldest output channel instrument instance, on the measurement screen with the lowest index, that is configured for replay mode.
<b>Example</b>	-> :SCReen:INSTRuments:OUTputchannel:STARt

### 16.2 :SCReen:INSTRuments:OUTputchannel:PAUSE

<b>Syntax</b>	<b>:SCReen:INSTRuments:OUTputchannel:PAUSE</b>
<b>Description</b>	Start streaming to analog out channels
<b>Parameter</b>	None
<b>Explanation</b>	Stop sending data on the default stream output instrument that is in the Started state.
<b>Example</b>	-> :SCReen:INSTRuments:OUTputchannel:PAUSE

### 16.3 :SCReen:INSTRuments:OUTputchannel:STOP

<b>Syntax</b>	<b>:SCReen:INSTRuments:OUTputchannel:STOP</b>
<b>Description</b>	Start streaming to analog out channels
<b>Parameter</b>	None
<b>Explanation</b>	Stop sending data on the default stream output instrument that is in the Started state. Then reset the playback cursor to the start position
<b>Example</b>	-> :SCReen:INSTRuments:OUTputchannel:STOP

### 16.4 :SCReen:INSTRuments:OUTputchannel:STATe?

<b>Syntax</b>	<b>:SCReen:INSTRuments:OUTputchannel:STATe?</b>
<b>Description</b>	Query the current state of the output channel instrument.
<b>Parameter</b>	None
<b>Explanation</b>	<p>Returns the instrument state as a string:</p> <ul style="list-style-type: none"> <li>• <b>Not_found</b> (no stream output instrument found on the measurement screens)</li> <li>• <b>Configuration_error</b> (instrument is not correctly set up and cannot be used)</li> <li>• <b>Blocked</b> (another instrument instance is currently playing)</li> <li>• <b>Started</b> (the instrument is streaming dmd data)</li> <li>• <b>Paused</b> (the instrument does not stream)</li> <li>• <b>Stopped</b> (the instrument does not stream, stream will start at the beginning)</li> <li>• <b>Active</b> (the instrument continuously sends data; not used for stream output)</li> <li>• <b>Error</b></li> </ul>
<b>Return format</b>	<String>
<b>Example</b>	<pre>-&gt; :SCReen:INSTRuments:OUTputchannel:START -&gt; :SCReen:INSTRuments:OUTputchannel:STATe? &lt;- Started</pre>

## 16.5 :SCReen:SAVE "PATH"

<b>Syntax</b>	<b>:CReen:SAVE [&lt;path&gt;]</b>											
<b>Description</b>	Save the current screen to a png file											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;path&gt;</td> <td>String</td> <td>A valid filename of the png file. If no absolute path is specified, the file is stored in the OXYGEN export folder</td> <td>A file starting with "screenshot" and with screen number and the current timestamp is generated in the OXYGEN export folder</td> </tr> </tbody> </table>				Name	Type	Range	Default	<path>	String	A valid filename of the png file. If no absolute path is specified, the file is stored in the OXYGEN export folder	A file starting with "screenshot" and with screen number and the current timestamp is generated in the OXYGEN export folder
	Name	Type	Range	Default								
<path>	String	A valid filename of the png file. If no absolute path is specified, the file is stored in the OXYGEN export folder	A file starting with "screenshot" and with screen number and the current timestamp is generated in the OXYGEN export folder									
<b>Explanation</b>	Saves the current OXYGEN instruments screen to a png file.											
<b>Example</b>	<pre> -&gt; :SCReen:SAVE //stores the current ↪screen with number to file (e.g. "c:/export/ ↪screenshot_1_20230713_120948.png") ... -&gt; :SCReen:SAVE "xyz" //stores the current ↪screen to file (e.g. "c:/export/xyz.png") ... -&gt; :SCReen:SAVE "c:/temp/xyz" //stores the current ↪screen to "c:/temp/xyz.png" </pre>											



## 16.6 :SCReen:ITEM<ScreenNumber>:SAVE "PATH"

<b>Syntax</b>	<b>:SCReen:ITEM&lt;ScreenNumber&gt;:SAVE [&lt;path&gt;]</b>											
<b>Description</b>	Save screen to a png file											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;path&gt;</td> <td>String</td> <td>A valid filename of the png file. If no absolute path is specified, the file is stored in the OXYGEN export folder</td> <td>A file starting with "screenshot" and with the &lt;ScreenNumber&gt; and the current timestamp is generated in the OXYGEN export folder</td> </tr> </tbody> </table>				Name	Type	Range	Default	<path>	String	A valid filename of the png file. If no absolute path is specified, the file is stored in the OXYGEN export folder	A file starting with "screenshot" and with the <ScreenNumber> and the current timestamp is generated in the OXYGEN export folder
	Name	Type	Range	Default								
<path>	String	A valid filename of the png file. If no absolute path is specified, the file is stored in the OXYGEN export folder	A file starting with "screenshot" and with the <ScreenNumber> and the current timestamp is generated in the OXYGEN export folder									
<b>Explanation</b>	Saves the OXYGEN instruments screen with the given <ScreenNumber> to a png file. The <ScreenNumber> should be a valid OXYGEN instrument screen (1 - 32768)											
<b>Example</b>	<pre> -&gt; :SCReen:ITEM1:SAVE                               //stores screen ↪number 1 to file (e.g. "c:/export/screenshot_1_ ↪20230713_120948.png") ... -&gt; :SCReen:ITEM2:SAVE "xyz"                          //stores screen ↪number 2 to file (e.g. "c:/export/xyz.png") ... -&gt; :SCReen:ITEM3:SAVE "c:/temp/xyz" //stores screen ↪number 3 to "c:/temp/xyz.png" </pre>											

## MEASUREMENT REPORT COMMANDS

### 17.1 :REPort:SAVE[:ALL] "PATH"

<b>Syntax</b>	<b>:REPort:SAVE:ALL [&lt;path&gt;]</b>											
<b>Description</b>	Save all report pages to a pdf file											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;path&gt;</td> <td>String</td> <td>A valid filename of the pdf file. If no absolute path is specified, the file is stored in the OXYGEN export folder.</td> <td>A file starting with "report" and the current timestamp is generated in the OXYGEN export folder. All OXYGEN reports are stored.</td> </tr> </tbody> </table>				Name	Type	Range	Default	<path>	String	A valid filename of the pdf file. If no absolute path is specified, the file is stored in the OXYGEN export folder.	A file starting with "report" and the current timestamp is generated in the OXYGEN export folder. All OXYGEN reports are stored.
Name	Type	Range	Default									
<path>	String	A valid filename of the pdf file. If no absolute path is specified, the file is stored in the OXYGEN export folder.	A file starting with "report" and the current timestamp is generated in the OXYGEN export folder. All OXYGEN reports are stored.									
<b>Explanation</b>	Save all possible OXYGEN reports to a pdf file. "ALL" is the preferred command of scpi group "REPort:SAVE" and does not have to be spelled out											
<b>Example</b>	<pre> -&gt; :REPort:SAVE:ALL           //stores all OXYGEN ↪report pages to file (e.g. "c:/export/report_ ↪20230713_120948.pdf") ... -&gt; :REPort:SAVE               //same command as ↪:REPort:SAVE:ALL ... -&gt; :REPort:SAVE:ALL "xyz"    //stores the current ↪report to file (e.g. "c:/export/xyz.pdf") ... -&gt; :REPort:SAVE "c:/temp/xyz" //stores the current ↪report to file "c:/temp/xyz.pdf" </pre>											

## 17.2 :REPort:ITEM<PageNumber>:SAVE "PATH"

<b>Syntax</b>	<b>:REPort:ITEM&lt;PageNumber&gt;:SAVE [&lt;path&gt;]</b>											
<b>Description</b>	Save a single report page to a pdf file											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;path&gt;</td> <td>String</td> <td>A valid filename of the pdf file. If no absolute path is specified, the file is stored in the OXYGEN export folder</td> <td>A file starting with "re- port" and the current timestamp is generated in the OXYGEN export folder</td> </tr> </tbody> </table>				Name	Type	Range	Default	<path>	String	A valid filename of the pdf file. If no absolute path is specified, the file is stored in the OXYGEN export folder	A file starting with "re- port" and the current timestamp is generated in the OXYGEN export folder
	Name	Type	Range	Default								
<path>	String	A valid filename of the pdf file. If no absolute path is specified, the file is stored in the OXYGEN export folder	A file starting with "re- port" and the current timestamp is generated in the OXYGEN export folder									
<b>Explanation</b>	Saves the OXYGEN report with the given <ReportNumber> to a pdf file. The <ReportNumber> should be a valid OXYGEN report (1 - 32768)											
<b>Example</b>	<pre> -&gt; :REPort:ITEM1:SAVE //stores page_ ↪number 1 to file (e.g. "c:/export/report_20230713_ ↪120948.pdf") ... -&gt; :REPort:ITEM2:SAVE "xyz" //stores page_ ↪number 2 to file (e.g. "c:/export/xyz.pdf") ... -&gt; :REPort:ITEM3:SAVE "c:/temp/xyz" //stores page_ ↪number 3 to "c:/temp/xyz.pdf" </pre>											

## MEASUREMENT HEADER DATA

With the Measurement Header Data the user can tag stored recordings with specific user defined data lines. The lines are managed via the Oxygen System Setup.

The following SCPI commands allows to retrieve and manipulate the measurement header data lines from the System Setup.

### 18.1 :HEADer:ADD <key>,<description>

<b>Syntax</b>	(1) :HEADer:ADD <key>, <description> (2) :HEADer:ADD TEXT, <key>, <description> (3) :HEADer:ADD NUMERIC_CONSTANT, <key>, <VALUE>																			
<b>Description</b>	Add a new measure header data line																			
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;KEY&gt;</td> <td>ASCII String</td> <td></td> <td>None</td> </tr> <tr> <td>&lt;DESCRIP- TION&gt;</td> <td>ASCII String</td> <td></td> <td>None</td> </tr> <tr> <td>&lt;VALUE&gt;</td> <td>Nrf</td> <td></td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<KEY>	ASCII String		None	<DESCRIP- TION>	ASCII String		None	<VALUE>	Nrf		None
Name	Type	Range	Default																	
<KEY>	ASCII String		None																	
<DESCRIP- TION>	ASCII String		None																	
<VALUE>	Nrf		None																	
<b>Explanation</b>	Command (1) + (2): Adds a new measurement header data line entry with the specified key Command (3) Adds a named constant with the specified value. Constants need to have unique names and can only edited while not in a measurement.																			
<b>Example</b>	<pre>-&gt; :HEAD:GET? "Header 1" -&gt; :HEAD:ADD "Header 1","Description in Line 1" -&gt; :HEAD:GET? "Header 1" &lt;- :HEAD:GET "Description in Line 1"</pre>																			

## 18.2 :HEADer:GET? <key>

<b>Syntax</b>	<b>:HEADer:GET? &lt;key&gt;</b>			
<b>Description</b>	Query a measurement data line			
<b>Parameter</b>				
	<b>Name</b>	<b>Type</b>	<b>Range</b>	<b>Default</b>
	<KEY>	ASCII String		None
<b>Explanation</b>	Returns the content (description) of the header data line with the specified key.			
<b>Return Format</b>	String			
<b>Example</b>	<pre>-&gt; :HEADer:GET? "Key 1" &lt;- :HEAD:GET "HEAD DATA Line 1"</pre>			

## 18.3 :HEADer:KEYs?

<b>Syntax</b>	<b>:HEADer:KEYs?</b>
<b>Description</b>	Query all keys of the measurement header data
<b>Parameter</b>	None
<b>Explanation</b>	Returns a list of all keys (names) of all measurement header data lines.
<b>Return Format</b>	[String[,String[...]]   None
<b>Example</b>	<pre>-&gt; :HEAD:ADD "Header 1","Description in Line 1" -&gt; :HEAD:ADD TEXT,"Header 2","Description in Line 2" -&gt; :HEAD:KEYs? &lt;- :HEAD:KEY "Header 1","Header 2"</pre>

### 18.4 :HEADer:SET <key>,<description>

<b>Syntax</b>	<p>(1) :HEADer:SET &lt;key&gt;,&lt;description&gt;                  (2) :HEADer:SET TEXT, &lt;key&gt;, &lt;description&gt;                  (3) :HEADer:SET NUMERIC_CONSTANT, &lt;key&gt;, &lt;VALUE&gt;</p>																
<b>Description</b>	Sets a content to a measure header data line																
<b>Parameter</b>	<table border="1"> <thead> <tr> <th data-bbox="448 651 660 703">Name</th> <th data-bbox="660 651 874 703">Type</th> <th data-bbox="874 651 1171 703">Range</th> <th data-bbox="1171 651 1383 703">Default</th> </tr> </thead> <tbody> <tr> <td data-bbox="448 703 660 748">&lt;KEY&gt;</td> <td data-bbox="660 703 874 748">ASCII String</td> <td data-bbox="874 703 1171 748"></td> <td data-bbox="1171 703 1383 748">None</td> </tr> <tr> <td data-bbox="448 748 660 831">&lt;DESCRIP-TION&gt;</td> <td data-bbox="660 748 874 831">ASCII String</td> <td data-bbox="874 748 1171 831"></td> <td data-bbox="1171 748 1383 831">None</td> </tr> <tr> <td data-bbox="448 831 660 882">&lt;VALUE&gt;</td> <td data-bbox="660 831 874 882">Nrf</td> <td data-bbox="874 831 1171 882"></td> <td data-bbox="1171 831 1383 882">None</td> </tr> </tbody> </table>	Name	Type	Range	Default	<KEY>	ASCII String		None	<DESCRIP-TION>	ASCII String		None	<VALUE>	Nrf		None
Name	Type	Range	Default														
<KEY>	ASCII String		None														
<DESCRIP-TION>	ASCII String		None														
<VALUE>	Nrf		None														
<b>Explanation</b>	<p>Command (1) + (2): Set the description value of the measurement header data line with the key &lt;key&gt;.</p> <p>Command (3) Set the value of the specified constant. Constants need to have unique names and can only edited while not in a measurement.</p>																
<b>Example</b>	<pre>-&gt; :HEAD:GET? "Header 1" &lt;- :HEAD:GET "Description in Line 1" -&gt; :HEAD:SET "Header 1","A new description in line 1" -&gt; :HEAD:GET? "Header 1" &lt;- :HEAD:GET "A new description in line 1"</pre>																

## 18.5 :HEADer:DELeTe <key>[,<key>[,...]]

<b>Syntax</b>	<b>:HEADer:DELeTe &lt;key&gt;,&lt;key&gt;,...</b>			
<b>Description</b>	Delete measurement header data line(s)			
<b>Parameter</b>				
	<b>Name</b>	<b>Type</b>	<b>Range</b>	<b>Default</b>
	<KEY>	ASCII String		None
<b>Explanation</b>	This command allows to delete one or more measurement header data lines. Constants cannot be deleted during measurement.			
<b>Example</b>	<pre> -&gt; :HEAD:GET? "Header 1" &lt;- :HEAD:GET "Description in Line 1" -&gt; :HEAD:KEYs? &lt;- :HEAD:KEY "Header 1","Header 2","Header 3" -&gt; :HEAD:DELeTe "Header 1","Header 3" -&gt; :HEAD:KEYs? &lt;- :HEAD:KEY "Header 2" </pre>			

## 18.6 :HEADer:VALues?

<b>Syntax</b>	<b>:HEADer:VALues?</b>
<b>Description</b>	Query the measurement header data
<b>Parameter</b>	
<b>Explanation</b>	Returns a list of all key value fields stored in the measurement header data. The third element is the data type, currently either TEXT or NUMERIC_CONSTANT. New elements may be added to the fields in the future depending on the type.
<b>Return Format</b>	(String, String),(String, String),...
<b>Example</b>	<pre> -&gt; :HEAD:VALues? &lt;- :HEAD:VAL ("Header 1","Description in line 1", ↪TEXT), ("Header 2","Description in Line 2",TEXT) </pre>

## UTILITY COMMANDS

### 19.1 :SYSTem:VERSion?

<b>Syntax</b>	<b>:SYSTem:VERSion?</b>
<b>Description</b>	Queries the implemented SCPI Standard Version
<b>Parameter</b>	None
<b>Explanation</b>	Queries the implemented SCPI Standard Version
<b>Return Format</b>	ASCII String
<b>Example</b>	<pre>-&gt; :SYSTem:VERSion? &lt;- "1999.0"</pre>

### 19.2 :SYSTem:HELP:HEADers?

<b>Syntax</b>	<b>:SYSTem:HELP:HEADers?</b>
<b>Description</b>	Queries the implemented commands as list
<b>Parameter</b>	None
<b>Explanation</b>	Queries the implemented commands as list
<b>Return Format</b>	#<num_int><num_char><Arbitrary Data>
<b>Example</b>	<pre>-&gt; :SYSTem:HELP:HEADers? &lt;- #41063 &lt;- :SYSTem:ERRor:ALL?/qonly/ &lt;- :SYSTem:ERRor:CODE:ALL?/qonly/ &lt;- :SYSTem:ERRor:CODE[:NEXT]?/qonly/ &lt;- :SYSTem:ERRor:COUNT?/qonly/ &lt;- ...</pre>





## TRIGGER EVENTS

Note that commands to set trigger event parameters are not valid if the OXYGEN is within a measurement.

### 20.1 :TRIGger[:GET]?

<b>Syntax</b>	<b>:TRIGger:GET?</b>
<b>Description</b>	Queries the number, enabled state and the name of all stored trigger events
<b>Parameter</b>	None
<b>Explanation</b>	Returns a list of elements alternating the numeric event-number, the event-enabled state and the event-name. Note: this is the default query in the trigger subsystem (see examples)
<b>Return Format</b>	(<Integer>, <ON OFF>, <String>)[,<Integer>, <ON   OFF>, <String>][,...]]   NONE
<b>Example</b>	<pre> -&gt; :TRIGger:GET? &lt;- (1,ON,"Event 1"),(2,ON,"My Event") ... -&gt; :TRIG? //use as default ↪query &lt;- (1,ON,"Event 1"),(2,ON,"My Event") </pre>

## 20.2 :TRIGger:RESet

<b>Syntax</b>	<b>:TRIGger:RESet</b>
<b>Description</b>	Delete all trigger events
<b>Parameter</b>	None
<b>Explanation</b>	This command removes all trigger events
<b>Example</b>	<pre>-&gt; :TRIGger:RESet -&gt; :TRIGger? &lt;- NONE</pre>

## 20.3 :TRIGger:ADDevent

<b>Syntax</b>	<b>:TRIGger:ADDevent</b>											
<b>Description</b>	Add a new trigger event											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;event name&gt;</td> <td>ASCII String</td> <td>optional</td> <td>"Event &lt;next event-number&gt;"</td> </tr> </tbody> </table>				Name	Type	Range	Default	<event name>	ASCII String	optional	"Event <next event-number>"
Name	Type	Range	Default									
<event name>	ASCII String	optional	"Event <next event-number>"									
<b>Explanation</b>	Add a new single trigger event to the trigger subsystem. If no event-name is specified then default name "Event <next event-number>" is assigned											
<b>Example</b>	<pre>-&gt; :TRIGger? &lt;- :NONE ↪ //no Events -&gt; :TRIGger:ADDevent -&gt; :TRIGger:ADDevent "My Event" -&gt; :TRIG? &lt;- (1,ON,"Event 1"),(2,ON,"My Event")</pre>											

## 20.4 :TRIGger:EvEnt<context> Commands and Queries

The Oxygen trigger subsystem facilitates one or more trigger events, to control the OXYGEN recording setup. Each event can be identified by a corresponding event number and an event name. The event number is used as context for several commands and queries and directs each command to the dedicated event (e.g. :TRIGger:EvEnt<event-number>:...). If no number is specified, event number 1 is assumed.

Each event is constructed with one or more condition(s) and one or more action(s). For each condition and each action a corresponding number is dedicated. These numbers also can be used as

context numbers for specific condition and action commands (e.g. :TRIGger:EVent<Number>:CONDi-tion<condition-number>:...).

The following commands and queries can be used to query and setup all available Oxygen trigger events.

## 20.5 :TRIGger:EVent<event-number>[:SETup]?

<b>Syntax</b>	<b>TRIGger:EVent&lt;event-number&gt;:SETup?</b>
<b>Description</b>	Queries the state, name, conditions and actions of a specific trigger event
<b>Parameter</b>	None
<b>Explanation</b>	Returns the state, name and a list of all conditions and all actions specified for the event with the corresponding event number. Note: this is the default query in the trigger event subsystem (see example).
<b>Return Format</b>	<ON   OFF>,<String>[,<Condition>[,...]] [,<Action>[,...]]   NONE
<b>Example</b>	<pre> -&gt; :TRIGger:EVent2:SETup? &lt;- ON, "Event 2"     ↪ //Event 2 without any condition or action ... -&gt; :TRIG:EV?     ↪ //use as default query, event number 1 is assumed     ↪ "1886450731343413248" //no action configured         </pre>

## 20.6 :TRIGger:EVent<event-number>[:SETup] {<String>} | {{ON|OFF},<String>}

<b>Syntax</b>	:TRIGger:EVent<event-number>:SETup {<String>}   {{ON OFF},<String>}			
<b>Description</b>	Sets the state and/or name and state of the corresponding event			
<b>Parameter</b>	<b>Name</b>	<b>Type</b>	<b>Range</b>	<b>Default</b>
	Event state or name	Literal   <String>	ON   OFF   <String>>	None
	Event name	<String>		None
<b>Explanation</b>	Sets the current state and/or name of the corresponding event. The state should be ON or OFF. Note: this is the default command in the trigger event subsystem (see example).			
<b>Example</b>	<pre> -&gt; :TRIGger:EVent2:SETup? &lt;- OFF, "Event 2" //Event_ ↳without any condition or action -&gt; TRIGger:EVent2:SETup "My Event" //set_ ↳name and the state enabled -&gt; :TRIG:EV2? &lt;- ON, "My Event" -&gt; :TRIG:EV2 OFF, "My Event 2" -&gt; :TRIG:EV2? &lt;- OFF, "My Event 2" </pre>			

## 20.7 :TRIGger:EVent<event-number>:VALId?

<b>Syntax</b>	:TRIGger:EVent<event-number>:VALId?
<b>Description</b>	Queries validation state of the corresponding event
<b>Parameter</b>	None
<b>Explanation</b>	Returns TRUE if the event has a valid configuration. FALSE is returned if the event does not exist, a condition is missing, or any of the conditions and actions are mismatched.
<b>Return Format</b>	TRUE   FALSE
<b>Example</b>	<pre> -&gt; :TRIGger:EVent1:VALId? &lt;- TRUE -&gt; :TRIG:EV2:VALI? &lt;- FALSE </pre>

## 20.8 :TRIGger:EVent<event-number>:DELeTe

<b>Syntax</b>	<b>:TRIGger:EVent&lt;event-number&gt;:DELeTe</b>
<b>Description</b>	Deletes the trigger event with the corresponding event number
<b>Parameter</b>	None
<b>Explanation</b>	Deletes the trigger event with all conditions and actions. Note: All subsequent events are reordered with a new event number. If no event number is specified, event 1 is deleted.
<b>Example</b>	<pre> -&gt; :TRIGger? &lt;- (1,ON,"Event 1"),(2,ON,"Event 2"),(3,ON,"Event 3") -&gt; :TRIGger:EVent2:DELeTe -&gt; :TRIGger? &lt;- (1,ON,"Event 1"),(2,ON,"Event 3") -&gt; :TRIGger:EVent2:DELeTe -&gt; :TRIGger? &lt;- (1,ON,"Event 1") </pre>

## 20.9 :TRIGger:EVent<event-number>:ADDCondition

<b>Syntax</b>	<b>:TRIGger:EVent&lt;event-number&gt;:ADDCondition</b>
<b>Description</b>	Add a new trigger event condition to the event
<b>Parameter</b>	None
<b>Explanation</b>	Add a new single condition to the corresponding trigger event. If no event-number is specified as context event number 1 is assumed. The constructed condition is a high level condition with its own default parameters. For changing the condition type or parameters see the condition setup commands below.
<b>Example</b>	<pre> -&gt; :TRIGger:EVent1? &lt;- ON,"Event 1" -&gt; :TRIGger:EVent1:ADDCondition -&gt; :TRIGger:EVent1? &lt;- ON,"Event 1", (CONDITION,HIGHLEVEL,0.0,OFF) / ↪ /event 1 with one high level condition </pre>

## 20.10 :TRIGger:EVent<event-number>:ADDAction

<b>Syntax</b>	<b>:TRIGger:EVent&lt;event-number&gt;:ADDAction</b>
<b>Description</b>	Add a new trigger event action to the event
<b>Parameter</b>	None
<b>Explanation</b>	Add a new single action to the corresponding trigger event. If no event-number is specified as context event number 1 is assumed. The constructed action is a recording action with start recording as default parameter. For changing the action type or parameters see the action setup commands below.
<b>Example</b>	<pre> -&gt; :TRIGger:EVent1? &lt;- ON, "Event 1" -&gt; :TRIGger:EVent1:ADDAction -&gt; :TRIGger:EVent1? &lt;- ON, "Event 1", (ACTION, RECORDING, START) //event_ ↪1 with one recording action </pre>

## 20.11 :TRIGger:EVent<event-number>:CONDition<condition-number>:GET?

<b>Syntax</b>	<b>:TRIGger:EVent&lt;event-number&gt;:CONDition&lt;condition-number&gt;:GET?</b>
<b>Description</b>	Queries the condition type and parameter of the corresponding trigger event condition
<b>Parameter</b>	None
<b>Explanation</b>	Returns the condition type and all of the specific parameters for this type. Note: this is the default query in the trigger event condition subsystem (see sample example).
<b>Return Format</b>	<p>((&lt;condition-type&gt;[,&lt;parameter 1&gt;[,...]])   NONE</p> <ul style="list-style-type: none"> <li>• High level condition: (CONDITION,HIGHLEVEL,&lt;nrf&gt;,&lt;literal&gt; &lt;nrf&gt;,String&gt;[,&lt;String&gt;[,...]])</li> <li>• Low level condition: (CONDITION,LOWLEVEL,&lt;nrf&gt;,&lt;literal&gt; &lt;nrf&gt;,String&gt;[,&lt;String&gt;[,...]])</li> <li>• In window condition: (CONDITION,INWINDOW,&lt;nrf&gt;,&lt;nrf&gt;,&lt;literal&gt; &lt;nrf&gt;,&lt;literal&gt; &lt;nrf&gt;,String&gt;[,&lt;String&gt;[,...]])</li> <li>• Out window condition: (CONDITION,OUTWINDOW,&lt;nrf&gt;,&lt;nrf&gt;,&lt;literal&gt; &lt;nrf&gt;,&lt;literal&gt; &lt;nrf&gt;,String&gt;[,&lt;String&gt;[,...]])</li> <li>• Keyboard condition: (CONDITION,KEYBOARD,&lt;literal&gt;,&lt;String&gt;)</li> <li>• Time condition: (CONDITION,TIME,&lt;literal&gt; &lt;String&gt;,&lt;literal&gt; &lt;nrf&gt;,&lt;literal&gt; &lt;nrf&gt;)</li> </ul> <p>For a detailed parameter name and description of all trigger event conditions see the condition setup commands below.</p>
<b>Example</b>	<pre>-&gt; :TRIGger:EVent1:CONDition1:GET? &lt;- NONE -&gt; :TRIGger:EVent1:ADDCondition -&gt; :TRIGger:EVent1:CONDition1? // ↪use as default query &lt;- (CONDITION,HIGHLEVEL,0.0,OFF) -&gt; :TRIG:EV:COND? // ↪short form &lt;- (CONDITION,HIGHLEVEL,0.0,OFF)</pre>



## 20.12 :TRIGger:EvEnt<event-number>:CONDition<condition-number>:VALId?

<b>Syntax</b>	<b>:TRIGger:EvEnt&lt;event-number&gt;:CONDition&lt;condition-number&gt;:VALId?</b>
<b>Description</b>	Queries validation state of the condition
<b>Parameter</b>	None
<b>Explanation</b>	Returns TRUE if the condition has a valid configuration. FALSE is returned if the condition does not exist, or one or more parameters are missing or mismatched (e.g. no channel assigned).
<b>Return Format</b>	TRUE   FALSE
<b>Example</b>	<pre>-&gt; :TRIGger:EvEnt1:CONDition1:VALId? &lt;- TRUE</pre>

## 20.13 :TRIGger:EvEnt<event-number>:CONDition<condition-number>:DELeTe

<b>Syntax</b>	<b>:TRIGger:EvEnt&lt;event-number&gt;:CONDition&lt;condition-number&gt;:DELeTe</b>
<b>Description</b>	Deletes the trigger condition with corresponding condition number
<b>Parameter</b>	None
<b>Explanation</b>	Deletes the trigger condition with the corresponding condition number from the corresponding event. Note: All subsequent conditions are reordered with a new condition number. If no condition number is specified, condition 1 is deleted.
<b>Example</b>	<pre>-&gt; :TRIGger:EvEnt1? &lt;- ON, "Event 1", (CONDITION, HIGHLEVEL, 0.0, OFF),   ↳ (CONDITION, KEYBOARD, SINGLE, "Shift+C") -&gt; :TRIGger:EvEnt1:CONDition1:DELeTe -&gt; :TRIGger:EvEnt1? &lt;- ON, "Event 1", (CONDITION, KEYBOARD, SINGLE, "Shift+C")</pre>

20.14 :TRIGger:Event<event-number>:CONDition<condition-number>:HIGHlevel:SETup  
 <nrf>,<literal>|<nrf>,<String>[,<String>[,...]]

<b>Syntax</b>	:TRIGger:Event<event-number>:CONDition<condition-number>:HIGHlevel:SETup <nrf>,<literal> <nrf>,<String>[,<String>[,...]]																			
<b>Description</b>	Converts corresponding condition to a high level condition																			
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>Threshold</td> <td>&lt;nrf&gt;</td> <td>channel range min max</td> <td></td> </tr> <tr> <td>Rearm level</td> <td>&lt;Boolean&gt;   &lt;nrf&gt;</td> <td>ON   OFF   range min - threshold value</td> <td></td> </tr> <tr> <td>Channel ID List</td> <td colspan="3">&lt;String&gt;[,&lt;String&gt;[,&lt;String&gt;[,...]]</td> </tr> </tbody> </table>				Name	Type	Range	Default	Threshold	<nrf>	channel range min max		Rearm level	<Boolean>   <nrf>	ON   OFF   range min - threshold value		Channel ID List	<String>[,<String>[,<String>[,...]]		
Name	Type	Range	Default																	
Threshold	<nrf>	channel range min max																		
Rearm level	<Boolean>   <nrf>	ON   OFF   range min - threshold value																		
Channel ID List	<String>[,<String>[,<String>[,...]]																			
<b>Explanation</b>	Converts the corresponding condition to a high level condition and sets its parameter. Note: the rearm level param should be ON or OFF or a numeric value (see channel range). If this parameter is a numeric value the rearm level is enabled automatically.																			
<b>Example</b>	<pre> -&gt; :TRIGger:EVent1:CONDition1:GET? &lt;- NONE -&gt; :TRIGger:EVent1:ADDCondition -&gt; :TRIGger:EVent1:CONDition1? // ↪ same GET as default query &lt;- (CONDITION,HIGHLEVEL,0.0,OFF) -&gt; :TRIGger:EVent1:CONDition1:HIGHlevel:SETup 1,0.5, ↪ "17342517731483713537", "17342517731483713538" -&gt; :TRIGger:EVent1:CONDition1? &lt;- (CONDITION,HIGHLEVEL,1.0,5.0E-1, ↪ "17342517731483713537", "17342517731483713538") -&gt; :TRIG:EV1:COND1:HIGH 1,0.5, "17342517731483713537", ↪ "17342517731483713538" //short command -&gt; :TRIGger:EVent1:CONDition1? &lt;- (CONDITION,HIGHLEVEL,1.0,5.0E-1, ↪ "17342517731483713537", "17342517731483713538")         </pre>																			

## 20.15 :TRIGger:Event<event-number>:CONDition<condition-number>:LOWlevel:SETup <nrf>,<literal>|<nrf>,<String>[,<String>[,...]]

<b>Syntax</b>	:TRIGger:Event<event-number>:CONDition<condition-number>:LOWlevel:SETup <nrf>,<literal> <nrf>,<String>[,<String>[,...]]																			
<b>Description</b>	Converts corresponding condition to a low level condition																			
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>Threshold</td> <td>&lt;nrf&gt;</td> <td>channel range min max</td> <td></td> </tr> <tr> <td>Rearm level</td> <td>&lt;Boolean&gt;   &lt;nrf&gt;</td> <td>ON   OFF   threshold value - range max</td> <td></td> </tr> <tr> <td>Channel ID List</td> <td colspan="3">&lt;String&gt;[,&lt;String&gt;[,&lt;String&gt;[,...]]</td> </tr> </tbody> </table>				Name	Type	Range	Default	Threshold	<nrf>	channel range min max		Rearm level	<Boolean>   <nrf>	ON   OFF   threshold value - range max		Channel ID List	<String>[,<String>[,<String>[,...]]		
Name	Type	Range	Default																	
Threshold	<nrf>	channel range min max																		
Rearm level	<Boolean>   <nrf>	ON   OFF   threshold value - range max																		
Channel ID List	<String>[,<String>[,<String>[,...]]																			
<b>Explanation</b>	Converts the corresponding condition to a low level condition and sets its parameter. Note: the rearm level param should be ON or OFF or a numeric value (see channel range). If this parameter is a numeric value the rearm level is enabled automatically.																			
<b>Example</b>	<pre> -&gt; :TRIGger:EVent1:CONDition1:GET? &lt;- NONE -&gt; :TRIGger:EVent1:ADDCondition -&gt; :TRIGger:EVent1:CONDition1? &lt;- (CONDITION,HIGHLEVEL,0.0,OFF) -&gt; :TRIGger:EVent1:CONDition1:LOWlevel:SETup 0.5,1.0, ↪ "17342517731483713537", "17342517731483713538" -&gt; :TRIGger:EVent1:CONDition1? &lt;- (CONDITION,LOWLEVEL,5.0E-1,1.0, ↪ "17342517731483713537", "17342517731483713538") -&gt; :TRIG:EV1:COND1:LOW 0.5,1.0,"17342517731483713537", ↪ "17342517731483713538" //short command -&gt; :TRIGger:EVent1:CONDition1? &lt;- (CONDITION,LOWLEVEL,5.0E-1,1.0, ↪ "17342517731483713537", "17342517731483713538") </pre>																			

20.16 :TRIGger:EvEnt<event-number>:CONDition<condition-number>:INwindow:SETup  
 <nrf>,<nrf>,<literal>|<nrf>,<literal>|<nrf>,<String>[,<String>[,...]]

<b>Syntax</b>	:TRIGger:EvEnt<event-number>:CONDition<condition-number>:INwindow:SETup <nrf>,<nrf>,<literal> <nrf>,<literal> <nrf>,<String>[,<String>[,...]]																											
<b>Description</b>	Converts corresponding condition to a in window condition																											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>Lower level</td> <td>&lt;nrf&gt;</td> <td>range min - upper level</td> <td></td> </tr> <tr> <td>Upper level</td> <td>&lt;nrf&gt;</td> <td>lower level - range max</td> <td></td> </tr> <tr> <td>Rearm lower level</td> <td>&lt;Boolean&gt;   &lt;nrf&gt;</td> <td>ON   OFF   threshold value - range max</td> <td></td> </tr> <tr> <td>Rearm upper level</td> <td>&lt;Boolean&gt;   &lt;nrf&gt;</td> <td>ON   OFF   threshold value - range max</td> <td></td> </tr> <tr> <td>Channel ID List</td> <td>&lt;String&gt;[,&lt;String&gt;[,&lt;String&gt;[,...]]</td> <td></td> <td></td> </tr> </tbody> </table>				Name	Type	Range	Default	Lower level	<nrf>	range min - upper level		Upper level	<nrf>	lower level - range max		Rearm lower level	<Boolean>   <nrf>	ON   OFF   threshold value - range max		Rearm upper level	<Boolean>   <nrf>	ON   OFF   threshold value - range max		Channel ID List	<String>[,<String>[,<String>[,...]]		
Name	Type	Range	Default																									
Lower level	<nrf>	range min - upper level																										
Upper level	<nrf>	lower level - range max																										
Rearm lower level	<Boolean>   <nrf>	ON   OFF   threshold value - range max																										
Rearm upper level	<Boolean>   <nrf>	ON   OFF   threshold value - range max																										
Channel ID List	<String>[,<String>[,<String>[,...]]																											
<b>Explanation</b>	Converts the corresponding condition to a in window condition and sets its parameters lower level, upper level and channel ids. For lower and upper level see the channel range min-max for relevant values.																											
<b>Example</b>	<pre> -&gt; :TRIGger:EvEnt1:CONDition1:GET? &lt;- NONE -&gt; :TRIGger:EvEnt1:ADDCondition -&gt; :TRIGger:EvEnt1:CONDition1? &lt;- (CONDITION,HIGHLEVEL,0.0,OFF) -&gt; :TRIGger:EvEnt1:CONDition1:INwindow:SETup ↪-0.5,0.5,-0.6,0.6,"5906189427729760280", ↪"5906189427729760281" -&gt; :TRIGger:EvEnt1:CONDition1? &lt;- (CONDITION,INWINDOW,-5.0E-1,5.0E-1,-1.0E-1,1.0E-1, ↪"5906189427729760280","5906189427729760281") -&gt; :TRIG:EV1:COND1:IN -0.5,0.5,-0.6,0.6, ↪"5906189427729760280","5906189427729760281" //↪ ↪short command -&gt; :TRIGger:EvEnt1:CONDition1? &lt;- (CONDITION,INWINDOW,-5.0E-1,5.0E-1,-1.0E-1,1.0E-1, ↪"5906189427729760280","5906189427729760281")         </pre>																											

## 20.17 :TRIGger:Event<event-number>:CONDition<condition-number>:OUTwindow:SETup <nrf>,<nrf>,<literal>|<nrf>,<literal>|<nrf>,<String>[,<String>[,...]]

<b>Syntax</b>	:TRIGger:Event<event-number>:CONDition<condition-number>:OUTwindow:SETup <nrf>,<nrf>,<literal> <nrf>,<literal> <nrf>,<String>[,<String>[,...]]																											
<b>Description</b>	Converts corresponding condition to a out window condition																											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>Lower level</td> <td>&lt;nrf&gt;</td> <td>range min - upper level</td> <td></td> </tr> <tr> <td>Upper level</td> <td>&lt;nrf&gt;</td> <td>lower level - range max</td> <td></td> </tr> <tr> <td>Rearm lower level</td> <td>&lt;Boolean&gt;   &lt;nrf&gt;</td> <td>ON   OFF   threshold value - range max</td> <td></td> </tr> <tr> <td>Rearm upper level</td> <td>&lt;Boolean&gt;   &lt;nrf&gt;</td> <td>ON   OFF   threshold value - range max</td> <td></td> </tr> <tr> <td>Channel ID List</td> <td>&lt;String&gt;[,&lt;String&gt;[,&lt;String&gt;[,...]]</td> <td></td> <td></td> </tr> </tbody> </table>				Name	Type	Range	Default	Lower level	<nrf>	range min - upper level		Upper level	<nrf>	lower level - range max		Rearm lower level	<Boolean>   <nrf>	ON   OFF   threshold value - range max		Rearm upper level	<Boolean>   <nrf>	ON   OFF   threshold value - range max		Channel ID List	<String>[,<String>[,<String>[,...]]		
Name	Type	Range	Default																									
Lower level	<nrf>	range min - upper level																										
Upper level	<nrf>	lower level - range max																										
Rearm lower level	<Boolean>   <nrf>	ON   OFF   threshold value - range max																										
Rearm upper level	<Boolean>   <nrf>	ON   OFF   threshold value - range max																										
Channel ID List	<String>[,<String>[,<String>[,...]]																											
<b>Explanation</b>	Converts the corresponding condition to a out window condition and sets its parameters lower level, upper level and channel ids. For lower and upper level see the channel range min-max for relevant values																											
<b>Example</b>	<pre> -&gt; :TRIGger:EVent1:CONDition1:GET? &lt;- NONE -&gt; :TRIGger:EVent1:ADDCondition -&gt; :TRIGger:EVent1:CONDition1? &lt;- (CONDITION,HIGHLEVEL,0.0,OFF) -&gt; :TRIGger:EVent1:CONDition1:OUTwindow:SETup -0.3,0. ↪3,ON,OFF,"5906189427729760280" -&gt; :TRIGger:EVent1:CONDition1? &lt;- (CONDITION,OUTWINDOW,-3.0E-1,3.0E-1,0.0,OFF, ↪"5906189427729760280") -&gt; :TRIG:EV1:COND1:OUT -0.3,0.3,ON,OFF, ↪"17342517731483713537" // short command -&gt; :TRIGger:EVent1:CONDition1? &lt;- (CONDITION,OUTWINDOW,-3.0E-1,3.0E-1,0.0,OFF, ↪"5906189427729760280") </pre>																											

## 20.18 :TRIGger:EVent<event-number>:CONDition<condition-number>:KEYBoard:SETup

<b>Syntax</b>	:TRIGger:EVent<event-number>:CONDition<condition-number>:KEYBoard:SETup<literal>,<String>															
<b>Description</b>	Converts corresponding condition to a keyboard condition															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>Pressed behaviour</td> <td>&lt;literal&gt;</td> <td>SINGLE   TOGGLE</td> <td></td> </tr> <tr> <td>Keyboard Key</td> <td>&lt;String&gt;</td> <td></td> <td></td> </tr> </tbody> </table>				Name	Type	Range	Default	Pressed behaviour	<literal>	SINGLE   TOGGLE		Keyboard Key	<String>		
Name	Type	Range	Default													
Pressed behaviour	<literal>	SINGLE   TOGGLE														
Keyboard Key	<String>															
<b>Explanation</b>	Converts the corresponding condition to a keyboard condition and sets its parameters pressed behaviour and keyboard key. Note: the keyboard key should be a valid key string with "Shift", "Ctrl", "Alt" and a valid key char (e.g. "Ctrl+Alt+K")															
<b>Example</b>	<pre> -&gt; :TRIGger:EVent1:CONDition1:GET? &lt;- NONE -&gt; :TRIGger:EVent1:ADDCondition -&gt; :TRIGger:EVent1:CONDition1? &lt;- (CONDITION,HIGHLEVEL,0.0,OFF) -&gt; :TRIGger:EVent1:CONDition1:KEYBoard:SETup TOGGLE, ↳"Ctrl+C" -&gt; :TRIGger:EVent1:CONDition1? &lt;- (CONDITION,KEYBOARD,TOGGLE,"Ctrl+C") -&gt; :TRIG:EV1:COND1:KEYB TOGGLE,"Ctrl+C" // short_ ↳command -&gt; :TRIGger:EVent1:CONDition1? &lt;- (CONDITION,KEYBOARD,TOGGLE,"Ctrl+C") </pre>															

## 20.19 :TRIGger:EVent<event-number>:CONDition<condition-number>:TIME:SETup

<b>Syntax</b>	:TRIGger:EVent<event-number>:CONDition<condition-number>:TIME:SETup <literal> <String>,<literal> <nrf>,<literal> <nrf>																			
<b>Description</b>	Converts corresponding condition to a time condition																			
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>First trigger time</td> <td>&lt;Boolean&gt;   &lt;String&gt;</td> <td>ON   OFF   "yyyy-MM-dd'T'HH:mm:ss"</td> <td></td> </tr> <tr> <td>Trigger interval time</td> <td>&lt;Boolean&gt;   &lt;nrf&gt;</td> <td>ON   OFF   seconds</td> <td></td> </tr> <tr> <td>Active for time</td> <td>&lt;Boolean&gt;   &lt;nrf&gt;</td> <td>ON   OFF   seconds</td> <td></td> </tr> </tbody> </table>				Name	Type	Range	Default	First trigger time	<Boolean>   <String>	ON   OFF   "yyyy-MM-dd'T'HH:mm:ss"		Trigger interval time	<Boolean>   <nrf>	ON   OFF   seconds		Active for time	<Boolean>   <nrf>	ON   OFF   seconds	
Name	Type	Range	Default																	
First trigger time	<Boolean>   <String>	ON   OFF   "yyyy-MM-dd'T'HH:mm:ss"																		
Trigger interval time	<Boolean>   <nrf>	ON   OFF   seconds																		
Active for time	<Boolean>   <nrf>	ON   OFF   seconds																		
<b>Explanation</b>	Converts the corresponding condition to a time condition and sets its parameters first trigger time, trigger interval and active for time. Note: the first trigger time should be ON or OFF or a valid ISO 8601 time string format. if a valid Iso string is given ON is assumed automatically.																			
<b>Example</b>	<pre> -&gt; :TRIGger:EVent1:CONDition1:GET? &lt;- NONE -&gt; :TRIGger:EVent1:ADDCondition -&gt; :TRIGger:EVent1:CONDition1? &lt;- (CONDITION,HIGHLEVEL,0.0,OFF) -&gt; :TRIGger:EVent1:CONDition1:TIME:SETup ↪ "2023-11-14T12:59:00",60,20 -&gt; :TRIGger:EVent1:CONDition1? &lt;- (CONDITION,TIME,"2023-11-14T12:59:00",60.0,20.0) -&gt; :TRIG:EV1:COND1:TIME "2023-11-14T12:59:00",60,20 / ↪ / short command -&gt; :TRIGger:EVent1:CONDition1? &lt;- (CONDITION,TIME,"2023-11-14T12:59:00",60.0,20.0) </pre>																			

## 20.20 :TRIGger:EvEnt<event-number>:ACTIon<action-number>:GET?

<b>Syntax</b>	<b>:TRIGger:EvEnt&lt;event-number&gt;:ACTIon&lt;action-number&gt;:GET?</b>
<b>Description</b>	Queries the action type and all parameters of the corresponding trigger event action
<b>Parameter</b>	None
<b>Explanation</b>	Returns the action type and all of the specific parameters for this type. Note: this is the default query in the trigger event action subsystem (see sample example).
<b>Return Format</b>	<p>([&lt;action-type&gt;[,&lt;parameter 1&gt;[,...]]])   NONE</p> <ul style="list-style-type: none"> <li>Recording action: (ACTION,RECORDING,START EVENT STOP PAUSE TOGGLE)</li> <li>Digital out action: (ACTION,DIGITALOUT,&lt;literal&gt; &lt;nrf&gt;,&lt;literal&gt; &lt;nrf&gt;,&lt;literal&gt;,&lt;String&gt;,&lt;String&gt;, ...)</li> <li>Alarm action: (ACTION,ALARM,&lt;literal&gt;,&lt;literal&gt; &lt;nrf&gt;,&lt;literal&gt; &lt;nrf&gt;,&lt;literal&gt;,&lt;String&gt;,&lt;String&gt;, ...)</li> <li>Marker action: (ACTION,MARKER,"Event 1",&lt;literal&gt;)</li> <li>Snapshot action:(ACTION,SNAPSHOT,&lt;literal&gt;,&lt;nrf&gt; &lt;String&gt;,&lt;String&gt;,&lt;String&gt;, ...)</li> <li>Arm action: (ACTION,ARM,&lt;literal&gt;)</li> </ul> <p>For a detailed parameter name and description of all trigger event actions see the action setup commands below.</p>
<b>Example</b>	<pre>-&gt; :TRIGger:EvEnt1:ACTIon1:GET? &lt;- NONE -&gt; :TRIGger:EvEnt1:ADDAction -&gt; :TRIGger:EvEnt1:ACTIon? &lt;- (ACTION,RECORDING,START)</pre>

## 20.21 :TRIGger:EvEnt<event-number>:ACTIon<action-number>:VALId?

<b>Syntax</b>	<b>:TRIGger:EvEnt&lt;event-number&gt;:ACTIon&lt;action-number&gt;:VALId?</b>
<b>Description</b>	Queries validation state of the corresponding action
<b>Parameter</b>	None
<b>Explanation</b>	Returns TRUE if the action has a valid configuration. FALSE is returned if the action does not exist, or one or more parameters are missing or mismatched (e.g. no channel assigned).
<b>Return Format</b>	TRUE   FALSE
<b>Example</b>	<pre>-&gt; :TRIGger:EvEnt1:ACTIon1:VALId? &lt;- TRUE</pre>



## 20.22 :TRIGger:EvEnt<event-number>:ACTIon<action-number>:DELEte

<b>Syntax</b>	<b>:TRIGger:EvEnt&lt;event-number&gt;:ACTIon&lt;action-number&gt;:DELEte</b>
<b>Description</b>	Deletes the trigger action with corresponding action number
<b>Parameter</b>	None
<b>Explanation</b>	Deletes the trigger action with the corresponding action number from the corresponding event. Note: All subsequent actions are reordered with a new action number. If no action number is specified, action 1 is deleted.
<b>Example</b>	

## 20.23 :TRIGger:EvEnt<event-number>:ACTIon<action-number>:RECOrding:SETup<literal>

<b>Syntax</b>	<b>:TRIGger:EvEnt&lt;event-number&gt;:ACTIon&lt;action-number&gt;:RECOrding:SETup&lt;literal&gt;</b>											
<b>Description</b>	Converts corresponding action to a recording action											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>Recording action</td> <td>&lt;literal&gt;</td> <td>START   EVENT   STOP   PAUSE   TOGGLE</td> <td></td> </tr> </tbody> </table>				Name	Type	Range	Default	Recording action	<literal>	START   EVENT   STOP   PAUSE   TOGGLE	
Name	Type	Range	Default									
Recording action	<literal>	START   EVENT   STOP   PAUSE   TOGGLE										
<b>Explanation</b>	Converts the corresponding action to a recording action and sets its action parameter. Note: this is the default command in the trigger event action recording subsystem (see sample example).											
<b>Example</b>	<pre> -&gt; :TRIGger:EvEnt1:ACTIon1:GET? &lt;- NONE -&gt; :TRIGger:EvEnt1:ADDAction -&gt; :TRIGger:EvEnt1:ACTIon1? &lt;- (ACTION, RECORDING, START) -&gt; :TRIGger:EvEnt1:ACTIon1:RECOrding:SETup EVENT -&gt; :TRIGger:EvEnt1:ACTIon1? &lt;- (ACTION, RECORDING, EVENT) -&gt; :TRIG:EV1:ACT1:REC EVENT // short command -&gt; :TRIGger:EvEnt1:ACTIon1? &lt;- (ACTION, RECORDING, EVENT) </pre>											

20.24 :TRIGger:EvEnt<event-number>:ACTion<action-number>:DIGOut:SETup  
 <literal>|<nrf>,<literal|<nrf>,<literal>,<String>[,<String>[,...]]

<b>Syntax</b>	<b>TRIGger:EvEnt&lt;event-number&gt;:ACTion&lt;action-number&gt;:DIGOut:SETup          &lt;literal&gt; &lt;nrf&gt;,&lt;literal &lt;nrf&gt;,&lt;literal&gt;,&lt;String&gt;[,&lt;String&gt;[,...]]</b>																							
<b>Description</b>	Converts corresponding action to a digital out action																							
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>Delay do channels</td> <td>&lt;Boolean&gt;   &lt;nrf&gt;</td> <td>ON   OFF   0-3600s</td> <td></td> </tr> <tr> <td>Auto reset do channels</td> <td>&lt;Boolean&gt;   &lt;nrf&gt;</td> <td>ON   OFF   0-3600s</td> <td></td> </tr> <tr> <td>Digital out level</td> <td>&lt;literal&gt;</td> <td>HIGH   LOW</td> <td></td> </tr> <tr> <td>Channel ID List</td> <td>&lt;String&gt;[,&lt;String&gt;[,&lt;String&gt;[,...]]</td> <td></td> <td></td> </tr> </tbody> </table>				Name	Type	Range	Default	Delay do channels	<Boolean>   <nrf>	ON   OFF   0-3600s		Auto reset do channels	<Boolean>   <nrf>	ON   OFF   0-3600s		Digital out level	<literal>	HIGH   LOW		Channel ID List	<String>[,<String>[,<String>[,...]]		
Name	Type	Range	Default																					
Delay do channels	<Boolean>   <nrf>	ON   OFF   0-3600s																						
Auto reset do channels	<Boolean>   <nrf>	ON   OFF   0-3600s																						
Digital out level	<literal>	HIGH   LOW																						
Channel ID List	<String>[,<String>[,<String>[,...]]																							
<b>Explanation</b>	Converts the corresponding action to a digital out action and sets its action parameters. Note: this is the default command in the trigger event action digital out subsystem (see sample example).																							
<b>Example</b>	<pre> -&gt; :TRIGger:EvEnt1:ACTion1:GET? &lt;- NONE -&gt; :TRIGger:EvEnt1:ADDAction -&gt; :TRIGger:EvEnt1:ACTion1? &lt;- (ACTION,RECORDING,START) -&gt; :TRIGger:EvEnt1:ACTion1:DIGout:SETup 1,60,LOW, ↪"17342517731483713537" -&gt; :TRIGger:EvEnt1:ACTion1? &lt;- (ACTION,DIGITALOUT,1.0,60.0,LOW, ↪"17342517731483713537") -&gt; :TRIG:EV1:ACT1:DIGO 1,60,LOW, ↪"17342517731483713537" // short command -&gt; :TRIGger:EvEnt1:ACTion1? &lt;- (ACTION,DIGITALOUT,1.0,60.0,LOW, ↪"17342517731483713537")         </pre>																							

## 20.25 :TRIGger:Event<event-number>:ACTion<action-number>:ALARm:SETup <literal>,<literal>|<nrf>,<literal|<nrf>,<literal>,<String>[,<String>[,...]]

<b>Syntax</b>	<b>TRIGger:Event&lt;event-number&gt;:ACTion&lt;action-number&gt;:ALARm:SETup          &lt;literal&gt;,&lt;literal&gt; &lt;nrf&gt;,&lt;literal &lt;nrf&gt;,&lt;literal&gt;,&lt;String&gt;[,&lt;String&gt;[,...]]</b>																											
<b>Description</b>	Converts corresponding action to a alarm action																											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>Add marker on Alarm</td> <td>&lt;Boolean&gt;</td> <td>ON   OFF</td> <td></td> </tr> <tr> <td>Auto reset do channels</td> <td>&lt;Boolean&gt;   &lt;nrf&gt;</td> <td>ON   OFF   0-3600s</td> <td></td> </tr> <tr> <td>Auto reset do channels</td> <td>&lt;Boolean&gt;   &lt;nrf&gt;</td> <td>ON   OFF   0-3600s</td> <td></td> </tr> <tr> <td>Digital out level</td> <td>&lt;literal&gt;</td> <td>HIGH   LOW</td> <td></td> </tr> <tr> <td>Channel ID List</td> <td>&lt;String&gt;[,&lt;String&gt;[,&lt;String&gt;[,&lt;String&gt;[,...]]]</td> <td></td> <td></td> </tr> </tbody> </table>				Name	Type	Range	Default	Add marker on Alarm	<Boolean>	ON   OFF		Auto reset do channels	<Boolean>   <nrf>	ON   OFF   0-3600s		Auto reset do channels	<Boolean>   <nrf>	ON   OFF   0-3600s		Digital out level	<literal>	HIGH   LOW		Channel ID List	<String>[,<String>[,<String>[,<String>[,...]]]		
Name	Type	Range	Default																									
Add marker on Alarm	<Boolean>	ON   OFF																										
Auto reset do channels	<Boolean>   <nrf>	ON   OFF   0-3600s																										
Auto reset do channels	<Boolean>   <nrf>	ON   OFF   0-3600s																										
Digital out level	<literal>	HIGH   LOW																										
Channel ID List	<String>[,<String>[,<String>[,<String>[,...]]]																											
<b>Explanation</b>	Converts the corresponding action to a alarm action and sets its action parameters. Note: this is the default command in the trigger event action alarm subsystem (see sample example).																											
<b>Example</b>	<pre> -&gt; :TRIGger:EVent1:ACTion1:GET? &lt;- NONE -&gt; :TRIGger:EVent1:ADDAction -&gt; :TRIGger:EVent1:ACTion1? &lt;- (ACTION,RECORDING,START) -&gt; :TRIGger:EVent1:ACTion1:ALARm:SETup ON,1,60,LOW, ↪"17342517731483713537" -&gt; :TRIGger:EVent1:ACTion1? &lt;- (ACTION,ALARM,ON,1.0,60.0,LOW, ↪"17342517731483713537") -&gt; :TRIG:EV1:ACT1:ALAR ON,1,60,LOW, ↪"17342517731483713537" // short command -&gt; :TRIGger:EVent1:ACTion1? &lt;- (ACTION,ALARM,ON,1.0,60.0,LOW, ↪"17342517731483713537") </pre>																											

## 20.26 :TRIGger:EVent<event-number>:ACTion<action-number>:MARKer:SETup <String>,<literal>

<b>Syntax</b>	:TRIGger:EVent<event-number>:ACTion<action-number>:MARKer:SETup <String>,<literal>															
<b>Description</b>	Converts corresponding action to a marker action															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>Marker Text</td> <td>&lt;String&gt;</td> <td></td> <td></td> </tr> <tr> <td>Append State</td> <td>&lt;literal&gt;</td> <td>ONACTIVE   ONINACTIVE   ONBOTH</td> <td></td> </tr> </tbody> </table>				Name	Type	Range	Default	Marker Text	<String>			Append State	<literal>	ONACTIVE   ONINACTIVE   ONBOTH	
Name	Type	Range	Default													
Marker Text	<String>															
Append State	<literal>	ONACTIVE   ONINACTIVE   ONBOTH														
<b>Explanation</b>	Converts the corresponding action to a marker action and sets its action parameters. Note: this is the default command in the trigger event marker subsystem (see sample example).															
<b>Example</b>	<pre> -&gt; :TRIGger:EVent1:ACTion1:GET? &lt;- NONE -&gt; :TRIGger:EVent1:ADDAction -&gt; :TRIGger:EVent1:ACTion1? &lt;- (ACTION,RECORDING,START) -&gt; :TRIGger:EVent1:ACTion1:MARKer:SETup "My Marker",     ↪ONBOTH -&gt; :TRIGger:EVent1:ACTion1? &lt;- (ACTION,MARKER,"My Marker",ONBOTH) -&gt; :TRIG:EV1:ACT1:MARK "My Marker",ONBOTH // short_     ↪command -&gt; :TRIGger:EVent1:ACTion1? &lt;- (ACTION,MARKER,"My Marker",ONBOTH) </pre>															

## 20.27 :TRIGger:EvEnt<event-number>:ACTion<action-number>:SNAPshot:SETup <literal>,<nrf>,<String>[,<String>[,...]]

<b>Syntax</b>	:TRIGger:EvEnt<event-number>:ACTion<action-number>:SNAPshot:SETup <literal>,<nrf> <String>,<String>[,<String>[,...]]			
<b>Description</b>	Converts corresponding action to a snapshot action			
<b>Parameter</b>	<b>Name</b>	<b>Type</b>	<b>Range</b>	<b>Default</b>
	Calculation Mode	<literal>	ACTUAL   MIN   RMS   PEAK   AVG   MAX   ACRMS	
	optional Snapshot Window	<nrf>	0.001 - 10s	Note: for ACTUAL calculation mode no window is needed
	Channel ID List	<String>[,<String>[,<String>[,...]]		
<b>Explanation</b>	Converts the corresponding action to a snapshot and sets its action parameters. Note: this is the default command in the trigger event action snapshot subsystem (see sample example).			
<b>Example</b>	<pre> -&gt; :TRIGger:EvEnt1:ACTion1:GET? &lt;- NONE -&gt; :TRIGger:EvEnt1:ADDAction -&gt; :TRIGger:EvEnt1:ACTion1? &lt;- (ACTION,RECORDING,START) -&gt; :TRIGger:EvEnt1:ACTion1:SNAPshot:SETup ACTUAL,   ↪ "7303155183163277312" -&gt; :TRIGger:EvEnt1:ACTion1? &lt;- (ACTION,SNAPSHOT,ACTUAL,"7303155183163277312") -&gt; :TRIG:EV1:ACT1:SNAP AVG,1.0,"7303155183163277312"↵   ↪ // short command -&gt; :TRIGger:EvEnt1:ACTion1? &lt;- (ACTION,SNAPSHOT,AVG,1.0,"7303155183163277312") </pre>			

20.28 :TRIGger:EVent<event-number>:ACTion<action-number>:ARM:SETup <literal>

<b>Syntax</b>	:TRIGger:EVent<event-number>:ACTion<action-number>:ARM:SETup <literal>											
<b>Description</b>	Converts corresponding action to an arm action											
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>Arm State</td> <td>&lt;literal&gt;</td> <td>ON   OFF</td> <td></td> </tr> </tbody> </table>				Name	Type	Range	Default	Arm State	<literal>	ON   OFF	
Name	Type	Range	Default									
Arm State	<literal>	ON   OFF										
<b>Explanation</b>	Converts the corresponding action to an arm and sets its action parameter. Note: this is the default command in the trigger event action arm subsystem (see sample example).											
<b>Example</b>	<pre> -&gt; :TRIGger:EVent1:ACTion1:GET? &lt;- NONE -&gt; :TRIGger:EVent1:ADDAction -&gt; :TRIGger:EVent1:ACTion1? &lt;- (ACTION,RECORDING,START) -&gt; :TRIGger:EVent1:ACTion1:ARM:SETup ON -&gt; :TRIGger:EVent1:ACTion1? &lt;- (ACTION,ARM,ON) -&gt; :TRIG:EV1:ACT1:ARM ON // short command -&gt; :TRIGger:EVent1:ACTion1? &lt;- (ACTION,ARM,ON) </pre>											



## ANALYSIS CONTROL

With the following commands and queries you can control the OXYGEN analysis mode.

Please note, when the OXYGEN analysis mode is active, several commands of other subsystems do not make sense and would return an execution error. The start/restart of oxygens acquisition is not possible in analysis mode. Several commands from the recording control would also return execution errors. The loading of setup files is not possible, but saving the setup of the current loaded dmd files is possible.

### 21.1 :ANALYSIS:ACTIVE?

<b>Syntax</b>	<b>:ANALYSIS:ACTIVE</b>
<b>Description</b>	Queries the OXYGEN analysis state
<b>Parameter</b>	None
<b>Explanation</b>	Return true if the OXYGEN is in the file analysis state.
<b>Return Format</b>	<Boolean>
<b>Example</b>	<pre>-&gt; :ANALYSIS:ACTIVE? &lt;- 0</pre>



## 21.2 :ANALysis:OPEN <file\_name>|<path>[,<file\_name>|<path>[,...]]

<b>Syntax</b>	<b>:ANALysis:OPEN &lt;file_name&gt; &lt;path&gt;[,&lt;file_name&gt; &lt;path&gt;[,...]]</b>															
<b>Description</b>	Load specified data files.															
<b>Parameter</b>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Range</th> <th>Default</th> </tr> </thead> <tbody> <tr> <td>&lt;file_name&gt;</td> <td>String</td> <td>The filename of the data file in the default data directory.</td> <td>None</td> </tr> <tr> <td>&lt;path&gt;</td> <td>String</td> <td>The absolute path of the setup file to load</td> <td>None</td> </tr> </tbody> </table>				Name	Type	Range	Default	<file_name>	String	The filename of the data file in the default data directory.	None	<path>	String	The absolute path of the setup file to load	None
	Name	Type	Range	Default												
	<file_name>	String	The filename of the data file in the default data directory.	None												
<path>	String	The absolute path of the setup file to load	None													
<b>Explanation</b>	<p>This command loads one or more measurement data files. If the specified file path is relative, the file is loaded from the current recording directory. The &lt;file_name&gt; does not need to include the file extension “.dmd”. If one file cannot be loaded, Oxygen switches back to live mode and returns one of the following error codes in the error queue.</p> <ul style="list-style-type: none"> <li>• Data corrupt or stale (-230)</li> <li>• Invalid format (-232)</li> <li>• Invalid version (-233)</li> <li>• File name not found (-256)</li> <li>• Settings conflict (-221)</li> <li>• Media protected (-258)</li> <li>• In all other error situations Execution error (-200) is returned</li> </ul>															
<b>Example</b>	<pre>-&gt; :ANALysis:OPEN "file1.dmd" -&gt; :ANALysis:ACTive? &lt;- 1 ... -&gt; :ANALysis:OPEN "file1.dmd", "file2.dmd" ... -&gt; :ANALysis:OPEN "d:/DATA/file1.dmd", "d:/DATA/file2. ↪dmd"</pre>															

### 21.3 :ANALysis:CLOSE

<b>Syntax</b>	<b>:ANALysis:CLOSE</b>
<b>Parameter</b>	None
<b>Description</b>	Closes all opened data files.
<b>Explanation</b>	This command closes all opened measurement data files. Oxygen returns to live mode.
<b>Example</b>	-> :ANALysis:CLOSE

### 21.4 :ANALysis:FILES?

<b>Syntax</b>	<b>:ANALysis:FILES?</b>
<b>Description</b>	Queries all opened data FILES.
<b>Parameter</b>	None
<b>Explanation</b>	Returns a list of all opened FILES. The returned filename is an absolute path.
<b>Return Format</b>	(<String>[,<String >, ...])   NONE
<b>Example</b>	-> :ANALysis:FILES? <- "d:/DATA/file1.dmd", "d:/DATA/file2.dmd"



## ERROR HANDLING

### 22.1 :SYSTem:ERRor[:NEXT]?

<b>Syntax</b>	<b>:SYSTem:ERRor[:NEXT]?</b>
<b>Description</b>	Queries the next element in the Error Queue
<b>Parameter</b>	None
<b>Explanation</b>	Queries the next element in the Error Queue
<b>Return Format</b>	<NR1>, String
<b>Example</b>	<pre>-&gt; :SYSTem:ERRor? &lt;- -102, "Syntax error" -&gt; :SYSTem:ERRor? &lt;- -108, "Parameter not allowed" -&gt; :SYSTem:ERRor? &lt;- 0, "No error"</pre>

### 22.2 :SYSTem:ERRor:ALL?

<b>Syntax</b>	<b>:SYSTem:ERRor:ALL?</b>
<b>Description</b>	Queries all entries the Error Queue
<b>Parameter</b>	None
<b>Explanation</b>	Queries all entries the Error Queue
<b>Return Format</b>	<NR1>, String, <NR1>, String, ...
<b>Example</b>	<pre>-&gt; :SYSTem:ERRor:ALL? &lt;- -102, "Syntax error", -108, "Parameter not allowed"</pre>

## 22.3 :SYSTem:ERRor:CODE[:NEXT]?

<b>Syntax</b>	<b>:SYSTem:ERRor:CODE:NEXT?</b>
<b>Description</b>	Queries next error code in the Error Queue
<b>Parameter</b>	None
<b>Explanation</b>	Queries next error code in the Error Queue
<b>Return Format</b>	<NR1>
<b>Example</b>	<pre>-&gt; :SYSTem:ERRor:CODE? &lt;- -102 -&gt; :SYSTem:ERRor:CODE? &lt;- -108</pre>

## 22.4 :SYSTem:ERRor:CODE:ALL?

<b>Syntax</b>	<b>:SYSTem:ERRor:CODE:ALL?</b>
<b>Description</b>	Queries all error codes in the Error Queue
<b>Parameter</b>	None
<b>Explanation</b>	Queries all error codes in the Error Queue
<b>Return Format</b>	<NR1>,<NR1>,...
<b>Example</b>	<pre>-&gt; :SYSTem:ERRor:CODE:ALL? &lt;- -102,-108</pre>

## 22.5 :SYSTem:ERRor:COUNT?

<b>Syntax</b>	<b>:SYSTem:ERRor:COUNT?</b>
<b>Description</b>	Queries the Error Queue number of elements
<b>Parameter</b>	None
<b>Explanation</b>	Queries the Error Queue number of elements
<b>Return Format</b>	<NR1>
<b>Example</b>	<pre>-&gt; :SYSTem:ERRor:COUNT? &lt;- 2</pre>

## 22.6 :SYSTem:ERRor:ENABle:ADD (<num>:<num>)

<b>Syntax</b>	<b>:SYSTem:ERRor:ENABle:ADD (&lt;num&gt;:&lt;num&gt;)</b>			
<b>Description</b>	Add a range of error codes to be queued in the Error Queue			
<b>Parameter</b>				
	<b>Name</b>	<b>Type</b>	<b>Range</b>	<b>Default</b>
	<num>	Integer	-32768 to 32768	None
<b>Explanation</b>	Add a range of error codes to be queued in the Error Queue.			
<b>Example</b>	<pre> -&gt; :SYSTem:ERRor:ENABle:LIST? &lt;- (-499:-100,1:32767) -&gt; :SYSTem:ERRor:ENABle:ADD (-1000:-900) -&gt; :SYSTem:ERRor:ENABle:LIST? &lt;- (-1000:-900,-499:-100,1:32767) </pre>			

## 22.7 :SYSTem:ERRor:ENABle:DELEte (<num>:<num>)

<b>Syntax</b>	<b>:SYSTem:ERRor:ENABle:DELEte (&lt;num&gt;:&lt;num&gt;)</b>			
<b>Description</b>	Delete a range of error codes to be queued in the Error Queue			
<b>Parameter</b>				
	<b>Name</b>	<b>Type</b>	<b>Range</b>	<b>Default</b>
	<num>	Integer	-32768 to 32768	None
<b>Explanation</b>	Delete a range of error codes to be queued in the Error Queue			
<b>Example</b>	<pre> -&gt; :SYSTem:ERRor:ENABle:LIST? &lt;- (-499:-100,1:32767) -&gt; :SYSTem:ERRor:ENABle:DELEte (-199:-100) -&gt; :SYSTem:ERRor:ENABle:LIST? &lt;- (-499:-200,1:32767) </pre>			

## 22.8 :SYSTem:ERRor:ENABle[:LIST]?

<b>Syntax</b>	<b>:SYSTem:ERRor:ENABle:LIST?</b>
<b>Description</b>	Queries the range of error codes to be queued in the Error Queue
<b>Parameter</b>	None
<b>Explanation</b>	Queries the range of error codes to be queued in the Error Queue
<b>Return Format</b>	(<NR1>:<NR1>,<NR1>:<NR1>)
<b>Example</b>	-> :SYSTem:ERRor:ENABle:LIST <- (-499:-100,1:32767)

## **ERROR CODES**

Use the `:SYSTEM:ERROR:ALL?` command to return all errors that have occurred. The errors returned have to following syntax `<Error Code>,<Error Message>`. The following table shows the error codes possible when interacting with OXYGEN software. SCPI defines the negative error codes, while the vendor specific error codes defined by DEWETRON GmbH are positive numbers (currently unused).



<b>Error Code</b>	<b>Error Message</b>	<b>Description</b>
0	No error	
100	Command error	
-102	Syntax error	Indicates that an unrecognized command or data type was encountered. For example, a string was received when the device does not accept strings.
-104	Data type error	The parser recognized a data element different than one allowed. For example, numeric or string data was expected but block data was encountered.
-108	Parameter not allowed	Indicates that less parameters were received than required for the header.
-109	Missing parameter	Indicates that more parameters were received than expected for the header
-113	Undefined header	Indicates the header is syntactically correct, but it is undefined for this specific device.
-114	Header suffix out of range	Indicates the value of a header suffix attached to a program mnemonic makes the header invalid.
-138	Suffix not allowed	Indicates that a suffix was encountered after a numeric element that does not allow suffixes.
-200	Execution error	General execution error.
-220	Parameter error	Indicates that a program data element related error occurred.
-221	Settings conflict	Indicates that a legal program data element was parsed but could not be executed due to the current device state.
-222	Data out of range	Indicates that a legal program data element was parsed but could not be executed because the interpreted value was outside the legal range defined by the devices.
-224	Illegal parameter value	Indicates that a program data element is ill-formed
-250	Mass storage error	Indicates that a mass storage error occurred.
-256	File name not found	Indicates that a legal program command or query could not be executed because the file name was not found on the media.
-294	Incompatible type	
-300	Device-specific error	
-350	Queue overflow	
-400	Query error	

## EXAMPLES

## 24.1 Fetch Online Measurement Data

```

-> *RST // Reset Device
-> :COMMunicate:HEADer 0 // Switch Off Header response
-> *IDN? // Query Identification
<- "DEWETRON,OXYGEN,0,2.5 RC1"
-> *VER? // Query Version Information
<- "SCPI,"1999.0",RC_SCPI,"1.5",OXYGEN,"2.5 RC1""
-> :SETUP:LOAD "scpi_test_setup.dms" // Load Measurement setup
-> :ACQU:STAT? // Query Acquisition state
<- Waiting_for_sync
-> :ACQU:STAT? // Query Acquisition state
<- Started
-> :RATE 500ms // Set Aggregation Rate to
↳500ms
-> :NUM:NORMal:ITEMs "ABS-TIME","U1_tRMS@PG1","I1_tRMS@PG1","P1_
↳t@PG1"
-> :NUM:NORMal:ITEMs? // Query Output Channels
<- "ABS-TIME","U1_tRMS@PG1","I1_tRMS@PG1","P1_t@PG1"
-> :NUM:NORMal:VAL?
<- "2017-08-28T13:17:26.9715+00:00",5.6568531E+1,5.6568531E+1,3.
↳1999988E+3

```

## 24.2 Store Measurement Data on Device

```

-> *RST // Reset Device
-> :COMMunicate:HEADer 0 // Switch Off Header response
-> *IDN? // Query Identification
<- "DEWETRON,OXYGEN,0,5.1.1"
-> *VER? // Query Version Information
<- "SCPI,"1999.0",RC_SCPI,"1.10",OXYGEN,"5.1.1""
-> :SETUP:LOAD "scpi_test_setup.dms" // Load Measurement setup
-> :ACQU:STAT? // Query Acquisition state
<- Waiting_for_sync
-> :ACQU:STAT? // Query Acquisition state
<- Started

```

```
-> :STORE:FILE:NAME "TEST_1"           // Set File Name for storing_
↳operation
-> :STORE:START                          // Start storing operation
-> :STORE:PAUSE                          // Pause storing operation
-> :STORE:START                          // Resume storing operation
<- :STORE:STOP                          // Stop storing operation
```

## 24.3 Set Channel Properties

### 24.3.1 Set a bool Item example

```
-> :CHANNEL:ID? "Sync Sim 0"           //get channel id
<- :CHANNEL:ID "10271021882991968256"
-> :CHANNEL:CONSTR? "10271021882991968256","Used" //check_
↳constraints
<- :CHANNEL:CONSTR (BOOL,OFF), (BOOL,ON)
-> CHANNEL:PROP? "10271021882991968256","Used" //get value
<- :CHANNEL:PROP (BOOL,OFF)
-> :CHANNEL:PROP "10271021882991968256","Used",ON //set value
-> :CHANNEL:PROP? "10271021882991968256","Used"
<- :CHANNEL:PROP (BOOL,ON)
-> :CHANNEL:PROP "10271021882991968256","Used",OFF
-> :CHANNEL:PROP? "10271021882991968256","Used"
<- :CHANNEL:PROP OFF
```

### 24.3.2 Set a string Item example

```
-> :CHANNEL:ID? "AI 2/1 Sim" //get channel id
<- :CHANNEL:ID "8785115489526349845"
-> :CHANNEL:PROP? "8785115489526349845","Mode" //get current value
<- :CHANNEL:PROP (STRING,"Voltage")
-> :CHANNEL:CONSTR? "8785115489526349845","Mode" //get item_
↳constraints
<- :CHANNEL:CONSTR (STRING,"Calibration"), (STRING,"Voltage"),
  (STRING,"Resistance"), (STRING,"IEPE"), (STRING,"Bridge"),
  (STRING,"ExcCurrentMonitor"), (STRING,"ExcVoltMonitor"),
  (STRING,"Current")
-> :CHANNEL:PROP "8785115489526349845","Mode","Resistance" //set_
↳new value
-> :CHANNEL:PROP? "8785115489526349845","Mode" //get current value
<- :CHANNEL:PROP (STRING,"Resistance")
```

### 24.3.3 Set a floating point Item example

```

-> :CHANNEL:PROP? "3789779077842337813", "Neon/PhysicalScaleFactor"
<- :CHANNEL:PROP (FLOAT,1.2)
-> :CHANNEL:PROP? "3789779077842337813", "Neon/PhysicalScaleOffset"
<- :CHANNEL:PROP (FLOAT,0.0)
-> :CHANNEL:PROP "3789779077842337813", "Neon/PhysicalScaleFactor", 1.
↪1
-> :CHANNEL:PROP "3789779077842337813", "Neon/PhysicalScaleOffset", 0.
↪1
-> :CHANNEL:PROP? "3789779077842337813", "Neon/PhysicalScaleFactor"
<- :CHANNEL:PROP (FLOAT,1.1)
-> :CHANNEL:PROP? "3789779077842337813", "Neon/PhysicalScaleOffset"
<- :CHANNEL:PROP (FLOAT,1.0E-1)

```

### 24.3.4 Set an enum item example

```

-> :CHANNEL:ID? "Sync Sim 0" //get channel id
<- :CHANNEL:ID "10439625394041651200"
-> :CHANNEL:PROP? "10439625394041651200", "Neon/Stored" //get_
↪current value
<- :CHANNEL:PROP (ENUM, "ChannelStored", "Auto")
-> :CHANNEL:CONSTR? "10439625394041651200", "Neon/Stored" //get_
↪constraints
<- :CHANNEL:CONSTR (ENUM, "ChannelStored", "Auto"), (ENUM,
↪"ChannelStored", "No")
-> :CHANNEL:PROP "10439625394041651200", "Neon/Stored", "No" //set_
↪item
-> :CHANNEL:PROP? "10439625394041651200", "Neon/Stored" //get_
↪current value
-> :CHANNEL:PROP (ENUM, "ChannelStored", "No")
-> :CHANNEL:PROP "10439625394041651200", "Neon/Stored",
↪"ChannelStored", "Auto"
-> :CHANNEL:PROP? "10439625394041651200", "Neon/Stored"
-> :CHANNEL:PROP (ENUM, "ChannelStored", "Auto")

```

### 24.3.5 Set scalar item example

```

-> :CHANNEL:ID? "AI 2/1 Sim"
<- :CHANNEL:ID "14649928104269578261"
-> :CHANNEL:PROP? "14649928104269578261", "SensorDelay" //get_
↪current value
<- :CHANNEL:PROP (SCALAR,0.0, "ms")
-> :CHANNEL:CONSTR? "14649928104269578261", "SensorDelay" //get item_
↪constraints
-> :CHANNEL:CONSTR (FLOAT,0.0), (FLOAT,500.0)
-> :CHANNEL:PROP "14649928104269578261", "SensorDelay", SCALAR, 100,
↪"ms" //set item
-> :CHANNEL:PROP? "14649928104269578261", "SensorDelay" //get_
↪current value

```

```
<- :CHANNEL:PROP (SCALAR,100.0,"ms")
-> :CHANNEL:PROP "14649928104269578261","SensorDelay",500 //set_
    ↪item
-> :CHANNEL:PROP? "14649928104269578261","SensorDelay" //get_
    ↪current value
-> :CHANNEL:PROP (SCALAR,500.0,"ms")
-> :CHANNEL:PROP "14649928104269578261","SensorDelay",0.4s //set_
    ↪item
-> :CHANNEL:PROP? "14649928104269578261","SensorDelay" //get_
    ↪current value
<- :CHANNEL:PROP (SCALAR,4.0E-1,"s")
-> :CHANNEL:PROP "14649928104269578261","SensorDelay",300,"ms"
-> :CHANNEL:PROP? "14649928104269578261","SensorDelay"
<- :CHANNEL:PROP (SCALAR,300.0,"ms")
```

### 24.3.6 Set range item example

```
-> :CHANNEL:PROP? "8785115489526349845","Range"
<- :CHANNEL:PROP (RANGE,-10.0,"V",10.0,"V")
-> :CHANNEL:CONSTR? "8785115489526349845","Range"
<- :CHANNEL:CONSTR (FLOAT,2.0E-4),(FLOAT,10.0),(RANGE,-10.0,"V",10.
    ↪0,"V"),
    (RANGE,-3.0,"V",3.0,"V"),(RANGE,-1.0,"V",1.0,"V"),
    (RANGE,-3.0E-1,"V",3.0E-1,"V"),(RANGE,-1.0E-1,"V",1.0E-1,"V"),
    (RANGE,-3.0E-2,"V",3.0E-2,"V"),(RANGE,-1.0E-2,"V",1.0E-2,"V")
-> :CHANNEL:PROP "8785115489526349845","Range",RANGE,-1.0E-2,"V",1.
    ↪0E-2,"V"
-> :CHANNEL:PROP? "8785115489526349845","Range" //get current_
    ↪value
<- :CHANNEL:PROP (RANGE,-1.0E-2,"V",1.0E-2,"V")
-> :CHANNEL:PROP "8785115489526349845","Range",-3.0V,3.0V //set_
    ↪value
-> :CHANNEL:PROP? "8785115489526349845","Range" //get current_
    ↪value
<- :CHANNEL:PROP (RANGE,-3.0,"V",3.0,"V")
```

### 24.3.7 Change sample rate with the sample rate divider example

```
-> :CHANNEL:PROP? "6995216112623288362","Neon/ReducedSampleRateActive"
<- :CHANNEL:PROP (BOOL,OFF)
-> :CHANNEL:PROP "6995216112623288362","Neon/ReducedSampleRateActive",
    ↪ON
-> :CHANNEL:PROP? "6995216112623288362","Neon/ReducedSampleRateActive"
<- :CHANNEL:PROP (BOOL,ON)
-> :CHANNEL:PROP? "6995216112623288362","Neon/WantedReducedSampleRate"
<- :CHANNEL:PROP (SCALAR,5000.0,"Hz")
-> :CHANNEL:PROP "6995216112623288362","Neon/WantedReducedSampleRate",
    ↪10000,"Hz"
-> :CHANNEL:PROP? "6995216112623288362","Neon/WantedReducedSampleRate"
<- :CHANNEL:PROP (SCALAR,10000.0,"Hz")
```