

# **Agilent N432A Thermistor Power Meter**

## **Programming Guide**



**Agilent Technologies**

## Notices

© Agilent Technologies, Inc. 2010

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Agilent Technologies, Inc. as governed by United States and international copyright laws.

### Manual Part Number

N432A-90004

### Edition

First Edition, May 13, 2010

Printed in Malaysia

Agilent Technologies, Inc.  
5301 Stevens Creek Blvd.  
Santa Clara, CA 95051 USA

### Warranty

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

### Technology Licenses

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

### Restricted Rights Legend

U.S. Government Restricted Rights. Software and technical data rights granted to the federal government include only those rights customarily provided to end user customers. Agilent provides this customary commercial license in Software and technical data pursuant to FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for the Department of Defense, DFARS 252.227-7015 (Technical Data - Commercial Items) and DFARS 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation).

### Safety Notices

#### CAUTION

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

---

#### WARNING

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

---

## General Warranty

The material contained in this document is provided “as is,” and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control. Duration and conditions of warranty for this product may be superseded when the product is integrated into (becomes a part of) other Agilent products. During the warranty period, Agilent will, at its option, either repair or replace products which prove to be defective. The warranty period begins on the date of delivery or on the date of installation if installed by Agilent.

## Restricted Rights Legend

The Software and Documentation have been developed entirely at private expense. They are delivered and licensed as “commercial computer software” as defined in DFARS 252.227-7013 (Oct 1988), DFARS 252.211-7015 (May 1991), or DFARS 252.227-7014 (Jun 1995), as a “commercial item” as defined in FAR 2.101(a), or as “restricted computer software” as defined in FAR 52.227-19 (Jun 1987) (or any equivalent agency regulation or contract clause), whichever is applicable. You have only those rights provided for such Software and Documentation by the applicable FAR or DFARS clause or the Agilent standard software agreement for the product involved.

# Equipment Operation

## Warnings and cautions

This guide uses warnings and cautions to denote hazards.

### WARNING

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

---

### CAUTION

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

---

## Personal safety considerations

This is a Safety Class I instrument (provided with a protective earthing ground incorporated in the power cord). The mains plug shall only be inserted in a socket outlet provided with a protective earth contact. Any interruption of the protective conductor inside or outside the instrument, is likely to make the instrument dangerous. Intentional interruption is prohibited. If this instrument is not used as specified, the protection provided by the instrument could be impaired. This instrument must be used in a normal condition (in which all means of protection are intact) only.

No operator serviceable parts inside. Only refer servicing to qualified personnel. To prevent electric shock, do not remove covers. For continued protection against fire, replace the line fuse(s) only with fuses of the same type and rating (for example, normal blow, time delay, and so on). The use of other fuses or material is prohibited.

## General Safety Considerations

The following general safety precautions must be observed during all phases of operation of this instrument. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the instrument. Agilent Technologies, Inc. assumes no liability for the customer's failure to comply with these requirements.

### **WARNING**

**Before this instrument is switched on, make sure that it has been properly grounded through the protective conductor of the AC power cable to a socket outlet provided with protective earth contact. Any interruption of the protective (grounding) conductor, inside or outside the instrument, or disconnection of the protective earth terminal can result in personal injury.**

---

### **CAUTION**

Any adjustments or service procedures that require operation of the instrument with protective covers removed should be performed only by trained service personnel.

---

## User Environment

This instrument is designed for indoor use only.

## In This Guide ...

### **1 N432A Remote Operation**

This chapter provides an overview on configuring the N432A remotely.

### **2 IEEE-488 Command Reference**

This chapter describes the IEEE-488 common commands supported by the N432A.

### **3 MEASurement Commands**

This chapter explains how to acquire data using high-level measurement commands.

### **4 CALCulate Subsystem**

This chapter explains how to use the CALCulate commands to perform post-acquisition data processing.

### **5 CALibration Subsystem**

This chapter describes how to perform zeroing and configure the reference calibration factor using the CALibration commands.

### **6 DISPlay Subsystem**

This chapter explains how the DISPlay commands are applied in the configuration of the windows used on the N432A display.

### **7 FORMat Subsystem**

This chapter describes how to set the data format for transferring numeric information using the FORMat commands.

### **8 HCOPy Subsystem**

This chapter discusses on how to use the HCOPy commands to output the N432A front panel display as an image file and to select the file format.

## **9 LXI Subsystem**

This chapter contains the information on how to enable or disable LXI identification using the LXI command.

## **10 MEMory Subsystem**

This chapter guides you on how to use the MEMory commands to configure the stored tables and save/recall registers.

## **11 OUTPut Subsystem**

This chapter explains how the OUTPut commands are applied in the N432A recorder and POWER REF output configuration.

## **12 SENSE Subsystem**

This chapter describes how the SENSE commands directly affect device-specific settings used to make measurements.

## **13 SERVICE Subsystem**

This chapter discusses how to obtain and set service-related information for servicing the N432A using the SERVICE commands.

## **14 STATUS Subsystem**

This chapter explains how the STATUS commands enable you to examine the status of the N432A by monitoring the status registers.

## **15 SYSTEM Subsystem**

This chapter provides the information on how the SYSTEM commands are used to perform system-related configurations for the N432A.

## **16 TRIGGER Subsystem**

This chapter describes how to synchronize device actions with events using the TRIGGER commands.

## **17 UNIT Subsystem**

This chapter explains how the UNIT commands are used to set the N432A measurement unit.

## **18 Programming Examples**

This chapter provides programming examples that help you to remotely program the N432A.

### **A Measurement Polling Example Using VEE**

This appendix gives an example on measurement polling using VEE.



# Contents

- 1 N432A Remote Operation 1**
  - Introduction **3**
  - Introduction to the SCPI Language **4**
  - Configuring the Remote Interface **14**
  - Zeroing **17**
  - Making Measurements **18**
  - Setting the Bridge Resistance **31**
  - Setting the Measurement Frequency **31**
  - Setting Offsets **32**
  - Setting Measurement Limits **40**
  - Using the Calibration/Correction Factors **43**
  - Optimizing the N432A Performance **49**
  - How Measurements are Calculated **52**
  - Status Reporting **53**
  - Saving and Recalling N432A Configurations **73**
  - Using Device Clear to Halt Measurements **74**
  - Summary of Commands **74**
  
- 2 IEEE-488 Command Reference 75**
  - SCPI Compliance Information **76**
  - \*CLS **77**
  - \*DDT <arbitrary block program data> | <string program data> **78**
  - \*ESE <NRf> **80**
  - \*ESR? **82**
  - \*IDN? **83**
  - \*LRN? **84**
  - \*OPC **85**
  - \*OPT? **86**
  - \*RCL <NRf> **87**

\*RST **88**  
\*SAV <NRf> **89**  
\*SRE <NRf> **90**  
\*STB? **92**  
\*TRG **93**  
\*TST? **94**  
\*WAI **95**  
GPIB Universal Commands **96**

### **3 MEASurement Commands 101**

MEASurement Commands **103**

CONFigure[1]|2|3|4? **108**

CONFigure [1]|2|3|4 Commands **110**

CONFigure[1]|2|3|4[:SCALar][:POWer:AC]  
[<expected\_value>,<resolution>,<source list>]] **111**

CONFigure[1]|2|3|4[:SCALar][:POWer:AC]:RELative  
[<expected\_value>,<resolution>,<source list>]] **113**

CONFigure[1]|2|3|4[:SCALar][:POWer:AC]:DIFFerence  
[<expected\_value>,<resolution>,<source list>]] **115**

CONFigure[1]|2|3|4[:SCALar][:POWer:AC]:DIFFerence:  
RELative [<expected\_value>,<resolution>,<source list>]] **117**

CONFigure[1]|2|3|4[:SCALar][:POWer:AC]:RATio  
[<expected\_value>,<resolution>,<source list>]] **119**

CONFigure[1]|2|3|4[:SCALar][:POWer:AC]:RATio:  
RELative [<expected\_value>,<resolution>,<source list>]] **121**

FETCh[1]|2|3|4 Queries **123**

FETCh[1]|2|3|4[:SCALar][:POWer:AC]? [<expected\_value>,<resolu-  
tion>,<source list>]] **124**

FETCh[1]|2|3|4[:SCALar][:POWer:AC]:RELative?  
[<expected\_value>,<resolution>,<source list>]] **126**

FETCh[1]|2|3|4[:SCALar][:POWer:AC]:DIFFerence?  
[<expected\_value>,<resolution>,<source list>]] **128**

FETCh[1]|2|3|4[:SCALar][:POWer:AC]:DIFFerence:

RELative? [<expected\_value>[,<resolution>[,<source list>]]] **130**  
 FETCh[1]|2|3|4[:SCALar][:POWer:AC]:RATio?  
 [<expected\_value>[,<resolution>[,<source list>]]] **132**  
 FETCh[1]|2|3|4[:SCALar][:POWer:AC]:RATio:RELative?  
 [<expected\_value>[,<resolution>[,<source list>]]] **134**  
 FETCh[1]|2|3|4[:SCALar]:V0? **136**  
 FETCh[1]|2|3|4[:SCALar]:V1? **137**  
 FETCh[1]|2|3|4[:SCALar]:VCMP0? **138**  
 FETCh[1]|2|3|4[:SCALar]:VCMP1? **139**  
 FETCh[1]|2|3|4[:SCALar]:VRF0? **140**  
 FETCh[1]|2|3|4[:SCALar]:VRF1? **141**  
 FETCh[1]|2|3|4[:SCALar]:PV1? **142**  
 READ[1]|2|3|4 Queries **143**  
 READ[1]|2|3|4[:SCALar][:POWer:AC]? [<expected\_value>[,<resolu-  
 tion>[,<source list>]]] **144**  
 READ[1]|2|3|4[:SCALar][:POWer:AC]:RELative?  
 [<expected\_value>[,<resolution>[,<source list>]]] **146**  
 READ[1]|2|3|4[:SCALar][:POWer:AC]:DIFFerence?  
 [<expected\_value>[,<resolution>[,<source list>]]] **148**  
 READ[1]|2|3|4[:SCALar][:POWer:AC]:DIFFerence:  
 RELative? [<expected\_value>[,<resolution>[,<source list>]]] **150**  
 READ[1]|2|3|4[:SCALar][:POWer:AC]:RATio? [<expected\_value>[,<resolu-  
 tion>[,<source list>]]] **152**  
 READ[1]|2|3|4[:SCALar][:POWer:AC]:RATio:RELative?  
 [<expected\_value>[,<resolution>[,<source list>]]] **154**  
 MEASure[1]|2|3|4 Queries **156**  
 MEASure[1]|2|3|4[:SCALar][:POWer:AC]? [<expected\_value>[,<resolu-  
 tion>[,<source list>]]] **157**  
 MEASure[1]|2|3|4[:SCALar][:POWer:AC]:RELative?  
 [<expected\_value>[,<resolution>[,<source list>]]] **159**  
 MEASure[1]|2|3|4[:SCALar][:POWer:AC]:DIFFerence?  
 [<expected\_value>[,<resolution>[,<source list>]]] **161**  
 MEASure[1]|2|3|4[:SCALar][:POWer:AC]:DIFFerence:

RELative? [<expected\_value>[,<resolution>[,<source list>]]] **163**

MEASure[1]|2|3|4[:SCALar][:POWer:AC]:RATio?  
[<expected\_value>[,<resolution>[,<source list>]]] **165**

MEASure[1]|2|3|4[:SCALar][:POWer:AC]:RATio:RELative?  
[<expected\_value>[,<resolution>[,<source list>]]] **167**

#### **4 CALCulate Subsystem 169**

CALCulate Subsystem **170**

CALCulate[1]|2|3|4:HOLD:STATe <character\_data> **173**

CALCulate[1]|2|3|4:FEED[1]|2 <string> **175**

CALCulate[1]|2|3|4:GAIN Commands **177**

CALCulate[1]|2|3|4:GAIN[:MAGNitude] <numeric\_value> **178**

CALCulate[1]|2|3|4:GAIN:STATe <boolean> **180**

CALCulate[1]|2|3|4:LIMit Commands **181**

CALCulate[1]|2|3|4:LIMit:CLEar:AUTO <boolean> |ONCE **182**

CALCulate[1]|2|3|4:LIMit:CLEar[:IMMEDIATE] **184**

CALCulate[1]|2|3|4:LIMit:FAIL? **185**

CALCulate[1]|2|3|4:LIMit:FCOunt? **186**

CALCulate[1]|2|3|4:LIMit:LOWer[:DATA] <numeric\_value> **188**

CALCulate[1]|2|3|4:LIMit:UPPer[:DATA] <numeric\_value> **190**

CALCulate[1]|2|3|4:LIMit:STATe <boolean> **192**

CALCulate[1]|2|3|4:MATH Commands **193**

CALCulate[1]|2|3|4:MATH[:EXPRession] <string> **194**

CALCulate[1]|2|3|4:MATH[:EXPRession]:CATalog? **196**

CALCulate[1]|2|3|4:RELative Commands **197**

CALCulate[1]|2|3|4:RELative[:MAGNitude]:AUTO <boolean> |ONCE **198**

CALCulate[1]|2|3|4:RELative:STATe <boolean> **200**

#### **5 CALibration Subsystem 203**

CALibration Subsystem **204**

CALibration[1]:RCFactor <numeric\_value> **205**

CALibration[1]:ZERO:AUTO <boolean> |ONCE **207**

- 6 DISPLAY Subsystem 209**
  - DISPlay Subsystem **210**
  - DISPlay:ENABle <boolean> **211**
  - DISPlay:SCReen:FORMat <character\_data> **212**
  - DISPlay[:WINDow[1]|2] Commands **214**
  - DISPlay[:WINDow[1]|2]:ANALog Commands **215**
  - DISPlay[:WINDow[1]|2]:ANALog:LOWer <numeric\_value> **216**
  - DISPlay[:WINDow[1]|2]:ANALog:UPPer <numeric\_value> **218**
  - DISPlay[:WINDow[1]|2]:FORMat <character\_data> **220**
  - DISPlay[:WINDow[1]|2]:METer Commands **222**
  - DISPlay[:WINDow[1]|2]:METer:LOWer <numeric\_value> **223**
  - DISPlay[:WINDow[1]|2]:METer:UPPer <numeric\_value> **225**
  - DISPlay[:WINDow[1]|2]:NUMeric[1]|2:RESolution <numeric\_value> **227**
  - DISPlay[:WINDow[1]|2]:SElect[1]|2 **229**
  - DISPlay[:WINDow[1]|2][:STATe] <boolean> **231**
- 7 FORMat Subsystem 233**
  - FORMat Subsystem **234**
  - FORMat[:READings]:BORDer <character\_data> **235**
  - FORMat[:READings][:DATA] <character\_data> **237**
- 8 HCOPy Subsystem 239**
  - HCOPy Subsystem **240**
  - HCOPy:SDUMp:DATA? **241**
  - HCOPy:SDUMp:DATA:FORMat BMP|PNG **242**
- 9 LXI Subsystem 245**
  - LXI Subsystem **246**
  - LXI:IDENtify[:STATe] <boolean> **247**
- 10 MEMory Subsystem 249**
  - MEMory Subsystem **250**

MEMory[:METer][1]:CATalog Queries **252**  
 MEMory[:METer][1]:CATalog[:ALL]? **253**  
 MEMory[:METer][1]:CATalog:STATe? **255**  
 MEMory[:METer][1]:CATalog:TABLE? **256**  
 MEMory[:METer][1]:CLEar Commands **258**  
 MEMory[:METer][1]:CLEar[:NAME] <character\_data> **259**  
 MEMory[:METer][1]:CLEar:TABLE **260**  
 MEMory[:METer][1]:FREE Queries **261**  
 MEMory[:METer][1]:FREE[:ALL]? **262**  
 MEMory[:METer][1]:FREE:STATe? **263**  
 MEMory[:METer][1]:FREE:TABLE? **264**  
 MEMory[:METer][1]:NSTates? **265**  
 MEMory[:METer][1]:STATe Commands **266**  
 MEMory[:METer][1]:STATe:CATalog? **267**  
 MEMory[:METer][1]:STATe:DEFine <character\_data>,<numeric\_value>  
**268**  
 MEMory[:METer][1]:TABLE Commands **270**  
 MEMory[:METer][1]:TABLE:FREQuency  
 <numeric\_value>{,<numeric\_value>} **271**  
 MEMory[:METer][1]:TABLE:FREQuency:POINts? **274**  
 MEMory[:METer][1]:TABLE:GAIN[:MAGNitude]  
 <numeric\_value>{,<numeric\_value>} **275**  
 MEMory[:METer][1]:TABLE:GAIN[:MAGNitude]:POINts? **277**  
 MEMory[:METer][1]:TABLE:MOVE <character\_data>,<character\_data>  
**278**  
 MEMory[:METer][1]:TABLE:SElect <character\_data> **279**

## **11 OUTPut Subsystem 281**

OUTPut Subsystem **282**  
 OUTPut:REcorder[1]:FEED <data\_handle> **283**  
 OUTPut:REcorder[1]:LIMit:AUTO <boolean> **285**  
 OUTPut:REcorder[1]:LIMit:LOWer <numeric\_value> **286**

OUTPut:REcorder[1]:LIMit:UPPer <numeric\_value> **288**

OUTPut:REcorder[1]:STATe <boolean> **290**

OUTPut:ROScillator[:STATe] <boolean> **291**

## **12 SENSE Subsystem 293**

SENSe Subsystem **294**

[SENSe[1]:]AVERAge Commands **296**

[SENSe[1]:]AVERAge:COUnT:VOLTage <character\_data> **297**

[SENSe[1]:]AVERAge:SDETect <boolean> **299**

[SENSe[1]:]BRESistance <character\_data> **301**

[SENSe[1]:]RSElection <character\_data> **303**

[SENSe[1]:]RVALue <numeric\_value> **305**

[SENSe[1]:]CORRection Commands **307**

[SENSe[1]:]CORRection:CFACtor|GAIN[1][:INPut][:MAGNitude]  
<numeric\_value> **308**

[SENSe[1]:]CORRection:CSET[1] Commands **310**

[SENSe[1]:]CORRection:CSET2 Commands **311**

[SENSe[1]:]CORRection:CSET[1][:SElect] <string> **312**

[SENSe[1]:]CORRection:CSET2[:SElect] <string> **314**

[SENSe[1]:]CORRection:CSET[1]:STATe <boolean> **316**

[SENSe[1]:]CORRection:CSET2:STATe <boolean> **318**

[SENSe[1]]|2|3|4:CORRection:DCYClE|GAIN3 Commands **320**

[SENSe[1]:]CORRection:DCYClE|GAIN3[:INPut][:MAGNitude]  
<numeric\_value> **321**

[SENSe[1]:]CORRection:DCYClE|GAIN3:STATe <boolean> **323**

[SENSe[1]:]CORRection:FDOffset Commands **324**

[SENSe[1]:]CORRection:FDOffset|GAIN4[:INPut][:MAGNitude]? **325**

[SENSe[1]:]CORRection:FDOffset:UNIT <character\_data> **326**

[SENSe[1]:]CORRection:GAIN2 Commands **327**

[SENSe[1]:]CORRection:GAIN2:STATe <boolean> **328**

[SENSe[1]:]CORRection:GAIN2[:INPut][:MAGNitude] <numeric\_value>  
**329**

[SENSe[1]:]FREQuency[:CW|:FIXed] <numeric\_value> **331**

### **13 SERVICE Subsystem 333**

SERVice Subsystem **334**

SERVice:BACKlight:BRIGhtness <numeric\_value> **336**

SERVice:BIST:CALibrator <boolean> **338**

SERVice:BIST:VRF <boolean> **339**

SERVice:BIST:VCOM <boolean> **340**

SERVice:BIST:V0 <boolean> **341**

SERVice:CALibrator:ADJust:COUR <numeric\_value> **342**

SERVice:CALibrator:ADJust:FINE <numeric\_value> **344**

SERVice:DISPlay:BScreen <boolean> **346**

SERVice:DISPlay:BScreen:SECure:ACTivation <numeric\_value> **347**

SERVice:DISPlay:BScreen:SECure:DEACTivation <numeric\_value> **348**

SERVice:FAN:FULL <boolean> **350**

SERVice:LAN:PHOSname **351**

SERVice:SECure:ERASe **352**

SERVice:SENSor[1]:TYPE? **353**

SERVice:SNUMber? **354**

SERVice:STATe <boolean> **355**

SERVice:VERSion:PROCCessor? **356**

SERVice:VERSion:SYSTem? **357**

### **14 STATus Subsystem 359**

STATus Subsystem **360**

Status Register Set Commands **362**

Device Status Register Set **366**

Operation Status Registers Sets **368**

STATus:OPERation **369**

STATus:OPERation:CALibrating[:SUMMary] **370**

STATus:OPERation:LLFail[:SUMMary] **371**



STATus:OPERation:MEASuring[:SUMMary] **372**  
STATus:OPERation:TRIGger[:SUMMary] **373**  
STATus:OPERation:ULFail[:SUMMary] **374**  
STATus:PRESet **375**  
Questionable Status Register Sets **376**  
STATus:QUEStionable **377**  
STATus:QUEStionable:CALibration[:SUMMary] **378**  
STATus:QUEStionable:POWer[:SUMMary] **379**  
Status Block Diagram **381**

## **15 SYSTEM Subsystem 383**

SYSTEM Subsystem **384**  
SYSTEM:COMMunicate:GPIB[:SELf]:ADDRes <numeric\_value> **386**  
SYSTEM:COMMunicate:LAN:AIP[:STATe] <boolean> **388**  
SYSTEM:COMMunicate:LAN:CURRent:ADDRes? **389**  
SYSTEM:COMMunicate:LAN:CURRent:DGATeway? **390**  
SYSTEM:COMMunicate:LAN:CURRent:DNAMe? **391**  
SYSTEM:COMMunicate:LAN:CURRent:SMASk? **392**  
SYSTEM:COMMunicate:LAN:ADDRes <character\_data> **393**  
SYSTEM:COMMunicate:LAN:DGATeway <character\_data> **394**  
SYSTEM:COMMunicate:LAN:DHCP[:STATe] <boolean> **395**  
SYSTEM:COMMunicate:LAN:DNAMe <character\_data> **396**  
SYSTEM:COMMunicate:LAN:HNAMe <character\_data> **397**  
SYSTEM:COMMunicate:LAN:MAC? **398**  
SYSTEM:COMMunicate:LAN:REStart **399**  
SYSTEM:COMMunicate:LAN:SMASk <character\_data> **400**  
SYSTEM:COMMunicate:TCPIp:CONTRol? **401**  
SYSTEM:DISPlay:BMP? **402**  
SYSTEM:ERRor? **403**  
SYSTEM:LOCal **411**  
SYSTEM:PRESet **412**

SYSTem:REMOte **416**  
SYSTem:RWLOCK **417**  
SYSTem:VERSion? **418**

## **16 TRIGger Subsystem 419**

TRIGger Subsystem **420**  
ABORt[1] **421**  
INITiate Commands **422**  
INITiate[1]:CONTinuous <boolean> **423**  
INITiate[1][:IMMediate] **425**  
INITiate:CONTinuous:ALL <boolean> **426**  
INITiate:CONTinuous:SEQuence[1] <boolean> **428**  
INITiate[:IMMediate]:ALL **430**  
INITiate[:IMMediate]:SEQuence[1] **431**  
TRIGger Commands **432**  
TRIGger[1]:DELay:AUTO <boolean> **433**  
TRIGger[1][:IMMediate] **435**  
TRIGger[1]:SOURce BUS|HOLD|IMMediate **436**  
TRIGger[:SEQuence[1]]:DELay:AUTO <boolean> **438**  
TRIGger[:SEQuence[1]]:IMMediate **440**  
TRIGger[:SEQuence[1]]:SOURce BUS|HOLD|IMMediate **441**

## **17 UNIT Subsystem 443**

UNIT Subsystem **444**  
UNIT[1]|2|3|4:POWer <amplitude\_unit> **445**  
UNIT[1]|2|3|4:POWer:RATio <ratio\_unit> **447**

## **18 Programming Examples 449**

Example 1: Identifying the N432A In Use **450**  
Example 2: FETCh, MEASure, and READ Queries **451**  
Example 3: Basic Power Measurement **453**  
Example 4: Status of Various Settings **454**

Example 5: Display Offset, Max/Min Hold, and Limit Settings **455**

Example 6: Frequency-Dependent Offsets **456**

**A Measurement Polling Example Using VEE 457**



## List of Figures

- Figure 1-1** Hierarchical structure of SCPI **4**
- Figure 1-2** <character\_data> format **7**
- Figure 1-3** <non-decimal numeric> format **9**
- Figure 1-4** <NR1> format **10**
- Figure 1-5** <NR2> format **10**
- Figure 1-6** <NR3> format **11**
- Figure 1-7** <string> format **12**
- Figure 1-8** Frequency-dependent offset tables **34**
- Figure 1-9** How measurements are calculated **52**
- Figure 1-10** General status register model **53**
- Figure 1-11** Typical status register bit changes **55**
- Figure 1-12** Status system **62**
- Figure 3-1** Measurement display CONFigure block window **103**
- Figure 4-1** Measurement display CALCulate block window **170**
- Figure 4-2** CALCulate block **171**
- Figure 17-1** Measurement display UNIT block window **444**



## List of Tables

<b>Table 1-1</b>	MEASure? and CONFigure preset states	<b>18</b>
<b>Table 1-2</b>	Range of values for window limits	<b>40</b>
<b>Table 1-3</b>	Bit definitions – Status Byte register	<b>63</b>
<b>Table 1-4</b>	Bit definitions – Standard Event register	<b>65</b>
<b>Table 1-5</b>	Bit definitions – Questionable Status register	<b>66</b>
<b>Table 1-6</b>	Bit change conditions for the Questionable Status register set	<b>67</b>
<b>Table 1-7</b>	Bit definitions – Operation Status	<b>68</b>
<b>Table 1-8</b>	Bit change conditions for the Operation Status register set	<b>69</b>
<b>Table 1-9</b>	Bit definitions – Device Status register	<b>70</b>
<b>Table 1-10</b>	Bit change conditions for the Device Status register	<b>71</b>
<b>Table 2-1</b>	Bit definitions: Standard Event register	<b>80</b>
<b>Table 2-2</b>	Bit definitions: Standard Event register	<b>82</b>
<b>Table 2-3</b>	Bit definitions: Status Byte register	<b>90</b>
<b>Table 2-4</b>	Bit definitions: Status Byte register	<b>92</b>
<b>Table 2-5</b>	Bit definitions: PPD	<b>97</b>
<b>Table 2-6</b>	Bit definitions: PPE	<b>98</b>
<b>Table 4-1</b>	Measurement units	<b>188</b>
<b>Table 4-2</b>	Measurement units	<b>190</b>
<b>Table 6-1</b>	Measurement units	<b>216</b>
<b>Table 6-2</b>	Measurement units	<b>218</b>
<b>Table 6-3</b>	Measurement units	<b>223</b>
<b>Table 6-4</b>	Measurement units	<b>225</b>
<b>Table 10-1</b>	Predefined sensor calibration tables	<b>254</b>
<b>Table 10-2</b>	Predefined sensor calibration tables	<b>257</b>
<b>Table 14-1</b>	Commands and events affecting the status registers	<b>360</b>
<b>Table 15-1</b>	Preset settings	<b>412</b>







# 1 N432A Remote Operation

Introduction	3
Introduction to the SCPI Language	4
Mnemonic forms	4
Using a colon (:)	4
Using a semicolon (;)	5
Using a comma (,)	5
Using whitespace	5
Using "?" commands	5
Using "*" commands	6
Syntax conventions	6
Syntax diagram conventions	6
SCPI data types	7
Input message terminators	13
Configuring the Remote Interface	14
Interface selection	14
GPIB address	14
LAN configuration	15
USB configuration	16
Zeroing	17
Zeroing	17
Making Measurements	18
Using MEASure?	19
Using the CONFigure command	22
Using the lower-level commands	29
Setting the Bridge Resistance	31
Setting the Measurement Frequency	31
Setting Offsets	32
Channel offsets	32



## 1 Power Meter Remote Operation

Display offsets	32
Using frequency-dependent offset tables	34
Editing frequency-dependent offset tables	35
Selecting frequency-dependent offset tables	37
Enabling a frequency-dependent offset table	38
Making the measurement	38
Setting Measurement Limits	40
Checking for limit failures	41
Using the Calibration/Correction Factors	43
Setting frequency-specific calibration factors	43
Using sensor calibration tables	43
Editing sensor calibration tables	44
Selecting sensor calibration tables	46
Enabling sensor calibration tables	47
Making the measurement	47
Optimizing the N432A Performance	49
Trigger mode	49
Output format	50
Units	51
Measurement commands	51
How Measurements are Calculated	52
Status Reporting	53
General status register model	53
How to use registers	57
Condition polling method	57
SRQ method	58
Using the Operation Complete commands	71
Saving and Recalling N432A Configurations	73
How to save and recall a configuration	73
Using Device Clear to Halt Measurements	74
Summary of Commands	74

This chapter describes the parameters that configure the N432A thermistor power meter and helps you determine the settings for performance optimization.

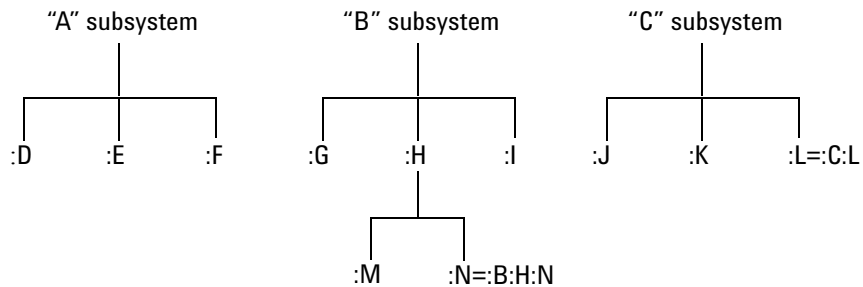
## Introduction

This chapter contains the following sections which provide the information on configuring the N432A remotely:

- [“Introduction to the SCPI Language”](#) on page 4
- [“Configuring the Remote Interface”](#) on page 14
- [“Zeroing”](#) on page 17
- [“Making Measurements”](#) on page 18
- [“Setting the Bridge Resistance”](#) on page 31
- [“Setting the Measurement Frequency”](#) on page 31
- [“Setting Offsets”](#) on page 32.
- [“Setting Measurement Limits”](#) on page 40.
- [“Using the Calibration/Correction Factors”](#) on page 43
- [“Optimizing the N432A Performance”](#) on page 49.
- [“How Measurements are Calculated”](#) on page 52.
- [“Status Reporting”](#) on page 53.
- [“Saving and Recalling N432A Configurations”](#) on page 73.
- [“Using Device Clear to Halt Measurements”](#) on page 74.

## Introduction to the SCPI Language

SCPI, also known as Standard Commands for Programmable Instruments, defines how you communicate with an instrument from a bus controller. The SCPI language uses a hierarchical structure similar to the file systems used by many bus controllers. The command tree is organized with root-level commands (also called subsystems) positioned at the top, with multiple levels below each root-level command. You must specify the complete path to execute the individual lower-level commands.



**Figure 1-1** Hierarchical structure of SCPI

## Mnemonic forms

Each keyword has both a long and a short form. A standard notation is used to differentiate the short-form keyword from the long-form keyword. The long form of the keyword is shown, with the short-form portion shown in upper-case characters, and the rest of the keyword shown in lower-case characters. For example, the short form of TRIGger is TRIG.

## Using a colon (:)

When a colon is the first character of a command keyword, it indicates that the next command mnemonic is a root-level command. When a colon is inserted between two command mnemonics, the colon moves the path down one level in the present path (for the specified root-level command) of the command tree. You *must* separate command mnemonics from each other using a colon. *You can omit the leading colon if the command is the first of a new program line.*

## Using a semicolon (;)

Use a semicolon to separate two commands within the same command string. The semicolon does not change the present path specified. For example, the following two statements are equivalent. Note that in the first statement, the first colon is optional but the third is compulsory.

```
:DISP:FORM DIG;:DISP:SEL2  
:DISP:FORM DIG;SEL2
```

## Using a comma (,)

If a command requires more than one parameter, you must separate adjacent parameters using a comma.

## Using whitespace

You *must* use whitespace characters, [tab], or [space] to separate a parameter from a command keyword. Whitespace characters are generally ignored *only* in parameter lists.

## Using “?” commands

The bus controller may send commands at any time, but a SCPI instrument may only send responses when *specifically* instructed to do so. Only queries (commands that end with a “?”) instruct the instrument to send a response message. Queries return either measured values or internal instrument settings.

### NOTE

If you send two queries without reading the response from the first, then attempt to read the second response, you may receive some data from the first response followed by the complete second response. To avoid this, do not send a query without reading the response. When you cannot avoid this situation, send a Device Clear before sending the second query.

## Using "\*" commands

Commands starting with a "\*" are called common commands. They are required to perform the identical function for *all* instruments that are compliant with the IEEE-488.2 interface standard. The "\*" commands are used to control reset, save/recall, self-test, and status operations in the N432A.

## Syntax conventions

Throughout this guide, the following conventions are used for SCPI command syntax:

- Square brackets ([]) indicate optional keywords or parameters.
- Braces ({} ) enclose one or more parameters that may be included zero or more times.
- Triangle brackets (<>) indicate that you must substitute a value for the enclosed parameter.
- Bars (|) can be read as "or" and are used to separate alternative parameter options.

## Syntax diagram conventions

Throughout this guide, the following conventions are used for SCPI command syntax diagrams:

- Solid lines represent the recommended path.
- Ovals enclose command mnemonics. The command mnemonic must be entered exactly as shown.
- Dotted lines indicate an optional path for bypassing secondary keywords.
- Arrows and curved intersections indicate command path direction.

## SCPI data types

The SCPI language defines different data formats for use in program messages and response messages. Instruments are flexible listeners and can accept commands and parameters in various formats. However, SCPI instruments are precise talkers. This means that SCPI instruments *always* respond to a particular query in a predefined, rigid format.

### <boolean> definition

Throughout this document, <boolean> is used to represent ON|OFF|<Nrf>. Boolean parameters have a value of 0 or 1 and are unitless. ON corresponds to 1 and OFF corresponds to 0.

On input, an <Nrf> is rounded to an integer. A nonzero result is interpreted as 1.

Queries always return a 1 or 0, never ON or OFF.

### <character\_data> definition

Throughout this document, <character\_data> is used to represent character data, that is, A to Z, a to z, 0 to 9, and \_ (underscore). START and R6\_5F are examples of character data. The first character must be an alphanumeric, followed by either alphanumeric or underscore characters up to a maximum of 12 characters. The format is defined as shown in the following figure.

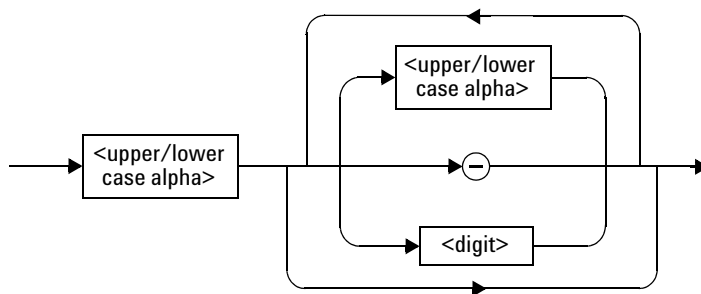


Figure 1-2 <character\_data> format

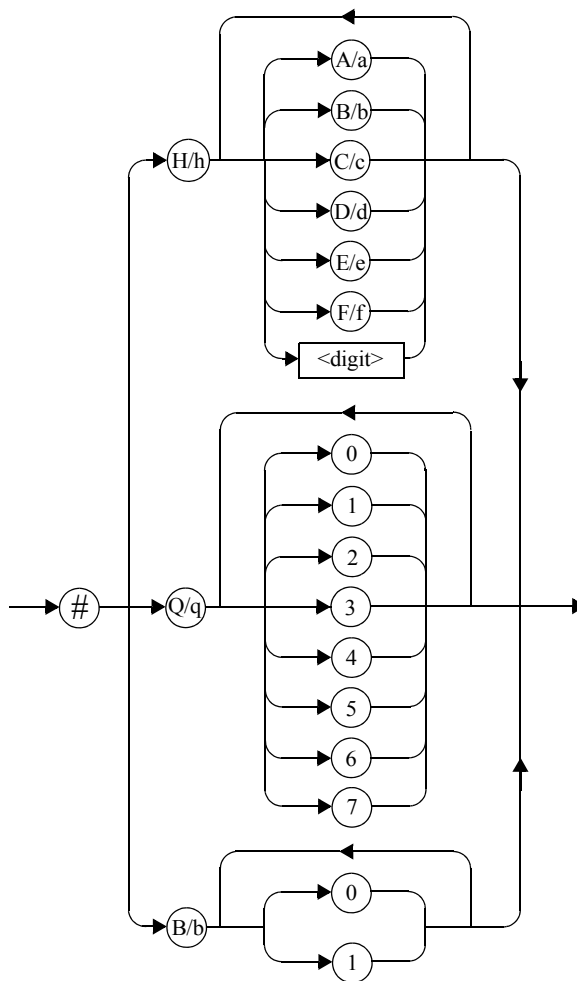
### **<NAN> definition**

Not a number (NAN) is represented as 9.91 E37. Not a number is defined in IEEE 754.

### **<non-decimal numeric> definition**

Throughout this document, <non-decimal numeric> is used to represent numeric information in bases other than ten (that is, hexadecimal, octal, and binary). Examples of non-decimal numeric include #HA2F, #ha4e, #Q62, #q15, and #B01011. [Figure 1-3](#) shows the non-decimal numeric standard data structures.





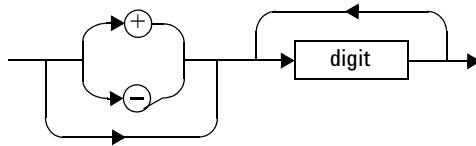
**Figure 1-3** <non-decimal numeric> format

**<NRf> definition**

Throughout this document, <NRf> is used to denote a flexible numeric representation, for example, +200; -56; +9.9E36.

### <NR1> definition

Throughout this document, the <NR1> numeric response data is defined as:



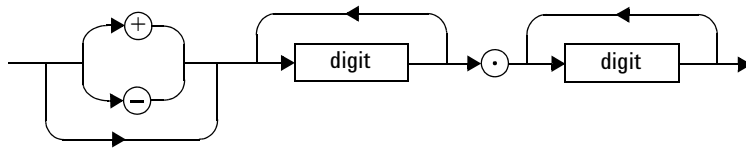
**Figure 1-4** <NR1> format

The following shows the examples of <NR1>:

- 146
- +146
- -12345

### <NR2> definition

Throughout this document, the <NR2> numeric response data is defined as:



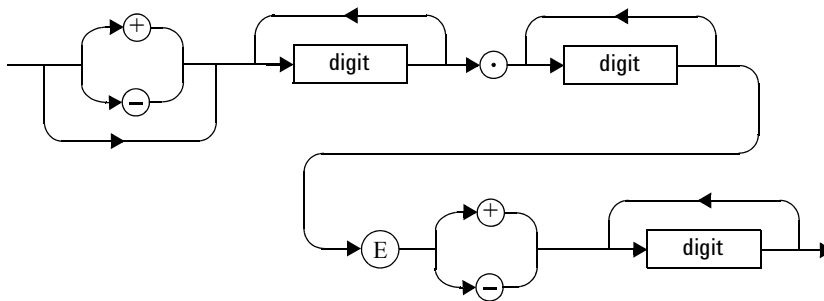
**Figure 1-5** <NR2> format

The following shows the examples of <NR2>:

- 12.3
- +1.2345
- -0.123

**<NR3> definition**

Throughout this document, the <NR3> numeric response data is defined as:



**Figure 1-6** <NR3> format

The following shows the examples of <NR3>:

- 1.23E+6
- 123.4E-54
- -1234.567E+90

**<numeric\_value> definition**

Throughout this document, the decimal numeric element is abbreviated to <numeric\_value>, for example, <NRf>, MINimum, MAXimum, DEFault, or Not A Number (NAN).

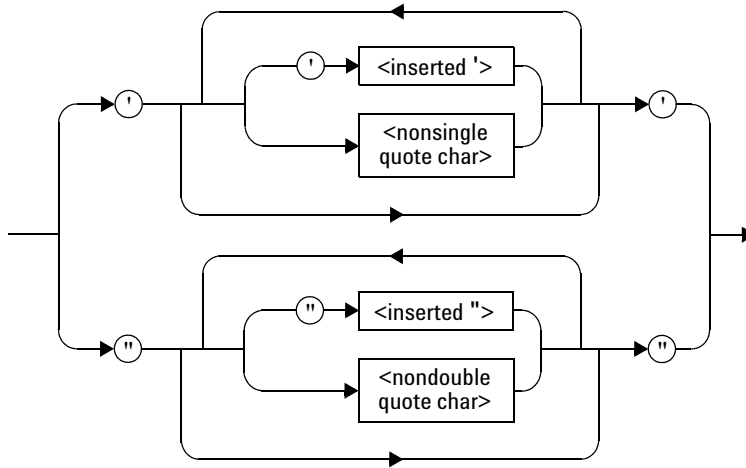
## 1 Power Meter Remote Operation

### <string> definition

Throughout this document, <string> is used to represent 7-bit ASCII characters.

The format is defined as:

#### Program data



#### Response data

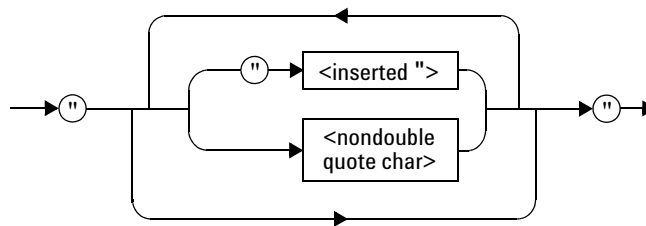


Figure 1-7 <string> format

## Input message terminators

Program messages sent to a SCPI instrument *must* terminate with a <newline> character. The IEEE.488 EOI (end or identify) signal is interpreted as a <newline> character and may also be used to terminate a message in place of the <newline> character. A <carriage return> followed by a <newline> is also accepted. Many programming languages allow you to specify a message terminator character or EOI state to be automatically sent with each bus transaction. Message termination *always* sets the current path back to the root-level.

## Configuring the Remote Interface

This section briefly describes how to configure the GPIB, LAN, and USB remote interfaces.

### NOTE

For more information on configuring the remote interface connectivity, refer to the *Agilent Technologies USB/LAN/GPIB Interfaces Connectivity Guide*. If you have installed the *IO Libraries Suite*, you can access the *Connectivity Guide* via the Agilent IO Libraries Control icon. Alternatively, you can access the *Connectivity Guide* via the Web at [www.agilent.com/find/connectivity](http://www.agilent.com/find/connectivity).

---

### Interface selection

You can choose to control the N432A remotely using the GPIB, LAN, or USB interface.

For information on selecting and configuring the remote interface manually from the front panel, refer to the *N432A Thermistor Power Meter Installation Guide*.

### NOTE

- It is expected that most users will use the front panel keys to set up the remote interfaces. The remote interface commands are provided for completeness (for the front panel operation).
  - To obtain optimum accuracy of the N432A in a remote interface environment, it is recommended to use the LAN interface to minimize ground noise contribution.
- 

### GPIB address

Each device on the GPIB (IEEE-488) interface must have a unique address. You can set the N432A address to any value between 0 and 30. The N432A is shipped with a default address of 10. The GPIB address is stored in non-volatile memory, and does not change when the N432A is switched off, or after a remote interface reset.

Your GPIB bus controller has its own address. Avoid using the bus controller address for any instrument on the interface bus. Agilent Technologies's controllers generally use address 21.

For information on setting the GPIB address manually from the front panel, refer to the *N432A Thermistor Power Meter Installation Guide*.

- To set the GPIB address from the remote interface, use the `SYSTEM:COMMunicate:GPIB[:SELF]:ADDRESS <numeric_value>` command.
- To query the GPIB address from the remote interface, use the `SYSTEM:COMMunicate:GPIB[:SELF]:ADDRESS? query`.

Refer to “`SYSTEM:COMMunicate:GPIB[:SELF]:ADDRESS <numeric_value>`” on page 386 for details.

## LAN configuration

The N432A has three LAN operating modes:

- Dynamic IP (Dynamic Host Configuration Protocol or DHCP)
- Auto IP (Local PC control or isolated (non-site) LAN)
- Static IP (Manual mode)

These three modes can be set up from the front panel. For front panel operation, refer to the *N432A Thermistor Power Meter Installation Guide*.

### Configuring the LAN remotely

To automatically configure the LAN settings, enable DHCP operation using the `SYSTEM:COMMunicate:LAN:DHCP[:STATe] <boolean>` command.

In this dynamic IP mode, the IP address, subnet mask, and default gateway values are obtained from a DHCP server. Using this dynamic IP mode does not require a detailed knowledge of your network configuration.

The IP address, subnet mask, default gateway, domain name, and host settings can be changed manually or remotely. To individually specify the LAN settings, use the following commands:

- IP address – `SYSTEM:COMMunicate:LAN:ADDRESS <character_data>`
- Subnet mask – `SYSTEM:COMMunicate:LAN:SMASk <character_data>`

## 1 Power Meter Remote Operation

- Default gateway – `SYSTEM:COMMunicate:LAN:DGATeway <character_data>`
- Domain name – `SYSTEM:COMMunicate:LAN:DNAME <character_data>`
- Host name – `SYSTEM:COMMunicate:LAN:HNAME <character_data>`
- Restart network – `SYSTEM:COMMunicate:LAN:REStart`

The `character_data` values for the IP address, subnet mask, and default gateway can range between 0.0.0.0 and 255.255.255.255.

For details on the individual command, refer to “[SYSTEM Subsystem](#)” on page 384.

### NOTE

If you configure an invalid IP address or an IP address that is used by another device or host, an error message is generated. This error can be read using the `SYSTEM:ERRor?` query.

The LAN setting values are stored in non-volatile memory and are not part of the save-recall function.

## USB configuration

The USB interface requires no front panel or remote configuration.

The USB address cannot be changed; it is set at the factory and is unique for each instrument.

### NOTE

- For further information on the USB configuration, refer to the *N432A Thermistor Power Meter Installation Guide*.
- Before connecting the USB cable, make sure that I/O software is installed on your PC.
- For more information on the *Agilent IO Libraries* software, refer to the *Connectivity Guide*.
- If you have installed other I/O software, refer to the documentation that accompanies the software.



## Zeroing

Zeroing adjusts the N432A for a zero power reading with no power applied to the thermistor sensor.

The `CALibration[1]:ZERO:AUTO <boolean>|ONCE` command causes the N432A to perform its zeroing routine when enabled.

Refer to “[CALibration Subsystem](#)” on page 204 for details on the available `CALibration[1]` commands.

## Making Measurements

The MEASure? query and CONFigure commands provide a straightforward method to program the N432A for measurements. You can select the measurement expected power level, resolution, and measurement type (single, difference, or ratio measurements), all in one command. The N432A automatically presets other measurement parameters to default values as shown in the table below.

**Table 1-1** MEASure? and CONFigure preset states

Command	MEASure? and CONFigure settings
Trigger source (TRIGger:SOURce)	Immediate
Trigger cycle (INITiate:CONTinuous)	Off
Trigger delay (TRIGger:DELay:AUTO)	On

An alternative method to program the N432A is to use the lower-level commands. The advantage of using the lower-level commands over the MEASure? query and CONFigure command is that they give you more precise control of the N432A. As shown in [Table 1-1](#), the CONFigure command presets various states in the N432A. It may be likely that you do not want to preset these states. Refer to [“Using the lower-level commands”](#) on page 29 for further information.

## Using MEASure?

The simplest way to program the N432A for measurements is by using the MEASure? query. However, this method does not offer much flexibility. When you send the query, the N432A selects the best settings for the requested configuration and immediately performs the measurement. You cannot change any settings (other than the expected power value, resolution, and measurement type) before the measurement is taken. To make more flexible and accurate measurements, use the CONFIGure command. MEASure? is a compound command which is equivalent to an ABORT, followed by a CONFIGure and a READ?.

### MEASure? examples

The following examples demonstrate how to use the MEASure? query to make a measurement. It is advisable to read through these examples in order, as they become increasingly more detailed. These examples configure the N432A for a measurement (as described in each individual example), automatically place the N432A in the “wait-for-trigger” state, internally trigger the N432A to take one reading, and then send the reading to the output buffer.

These examples give an overview of the MEASure? query. For further information, refer to “[MEASurement Commands](#)” on page 101.

### Example 1 – The simplest method

The following example shows the simplest method of making single measurements using MEAS?. Using MEAS1? results in an upper-window measurement, and MEAS2? in a lower-window measurement. The channel can be set using the source list parameter (refer to “[Example 2 – Specifying the source list parameter](#)”), or defaults as in this example.

specifies window



MEAS1?

MEAS2?

### Example 2 – Specifying the source list parameter

The MEASure? query has three optional parameters, an expected power value, a resolution, and a source list. These parameters must be entered in the specified order. Parameters are defaulted from the right by omitting them, or anywhere by substituting the keyword DEFault. DEFault is used as a placeholder.

The source list parameter is used to specify a measurement channel. The N432A supports only one channel (channel A), therefore, the value is (@1). In the following example, the expected power level and resolution parameters are set to their default values, leaving them at their current settings. The measurement is carried out on the upper window.

specifies window                      specifies channel

↓    ↓

MEAS1? DEF, DEF, (@1)

### Example 3 – Specifying the expected power parameter

#### NOTE

The expected power parameter is only required for non-thermistor sensors. This parameter is provided for backward compatibility purpose and has no effect on thermistor sensors.

The previous example details the three optional parameters which can be used with the MEASure? query. The first optional parameter is used to enter an expected power value.

The following example uses the expected power level parameter to specify a value of -50 dBm. The resolution parameter is defaulted, leaving it at its current setting. The source list parameter specifies a Channel A measurement. The measurement is displayed on the lower window.

specifies expected power value

specifies window                      ↓    ↓    ↓    specifies channel

MEAS2? -50, DEF, (@1)

### Example 4 – Specifying the resolution parameter

The previous examples detailed the use of the expected power value and source list parameters. The resolution parameter is used to set the resolution of the specified window.

The following example uses the resolution parameter to specify a resolution setting of 3. This setting represents three significant digits if the measurement unit is W or %, and 0.01 dB if the unit is dB or dBm. Refer to [Chapter 3, “MEASurement Commands”](#) for further details on the resolution parameter. The expected power level and source list parameters are set to their default values in the example. The expected power level remains unchanged at its current setting. As the source list parameter is the last specified parameter, you do not have to specify DEF. The measurement is carried out on the upper window.

specifies window      specifies resolution setting

MEAS1? DEF, 3

### Example 5 – Making a difference measurement

The following example queries the lower window to make a difference measurement of Channel A – Channel A. The expected power level and resolution parameters are defaulted, leaving them at their current settings.

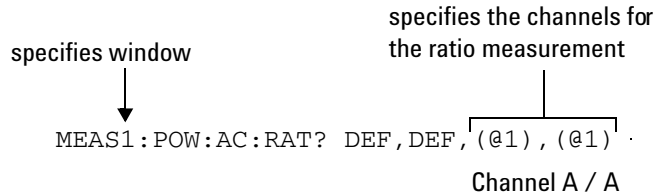
specifies window      specifies the channels for the difference measurement

MEAS2:POW:AC:DIFF? DEF, DEF, (@1), (@1)

Channel A – A

### Example 6 – Making a ratio measurement

The following example queries the upper window to make a ratio measurement of Channel A/A. The expected power level and resolution parameters are defaulted, leaving them at their current settings.



## Using the CONFigure command

When you execute this command, the N432A presets the optimum settings for the requested configuration (similar to the MEASure? query). However, the measurement is not automatically started and you can change the measurement parameters before making measurements. This allows you to change the N432A configuration from the preset conditions. The N432A offers a variety of low-level commands in the SENSE, CALCulate, and TRIGger subsystems. For example, if you want to change the frequency setting, use the [SENSE[1]:]FREQuency[:CW|:FIXed] command.

**Use the *INITiate* command or *READ?* query to initiate the measurement.**

### Using READ?

CONFigure does not take the measurement. One method of obtaining a measurement result is to use the READ? query. The READ? query takes the measurement using the parameters set by the CONFigure command then sends the reading to the output buffer. New data is obtained using the READ? query.

## Using INITiate and FETCh?

Another method of obtaining the measurement results is to use the INITiate command and FETCh? query. The INITiate command causes the measurement to be taken. The FETCh? query retrieves a reading when the measurement has completed, and sends the reading to the output buffer. FETCh? can be used to retrieve the measurement results in a number of different formats without taking fresh data for each measurement.

## CONFigure examples

The following examples demonstrate how to use the CONFigure commands, together with READ?, INITiate?, and FETCh? to make measurements.

It is advisable to read through these examples in order as they become increasingly more detailed.

For further information on the CONFigure commands, refer to [Chapter 3, “MEASurement Commands”](#).

### Example 1 – The simplest method

The following program segments show the simplest method of querying the upper-window and lower-window measurement results.

#### Using READ?

*RST	<i>Resets the N432A.</i>
CONF1	<i>Configures the upper- window measurement – sets to a single measurement by default.</i>
READ1?	<i>Initiates and retrieves the upper- window measurement.</i>
*RST	<i>Resets the N432A.</i>
CONF2	<i>Configures the lower- window measurement – sets to a single measurement by default.</i>
READ2?	<i>Initiates and retrieves the lower- window measurement.</i>

### Using INITiate and FETCh?

*RST	<i>Resets the N432A.</i>
CONF1	<i>Configures the upper- window measurement – sets to a single measurement by default.</i>
INIT1?	<i>Sets the channel in a wait-for- trigger state.</i>
FETC1?	<i>Triggers the upper- window measurement, then retrieves the measurement reading.</i>

*RST	<i>Resets the N432A.</i>
CONF2	<i>Configures the lower- window measurement – sets to a single measurement by default.</i>
INIT1?	<i>Sets the channel in a wait-for- trigger state.</i>
FETC2?	<i>Triggers the lower- window measurement, then retrieves the measurement reading.</i>

### Example 2 – Specifying the source list parameter

The CONFigure command and READ? query have three optional parameters, an expected power value, a resolution, and a source list. These parameters must be entered in the specified order. Parameters may be defaulted from the right by omitting them, or anywhere by substituting the keyword DEFault. DEFault is used as a placeholder.

The following examples use the source list parameter to specify the measurement channel as channel A. The expected power and resolution parameters are set to their default values, leaving them at their current settings. The measurement is carried out on the upper window.

Although the READ? and FETCh? queries have three optional parameters, it is not necessary to define them as shown in these examples. If they are defined, they must be identical to those defined in the CONFigure command, otherwise an error will occur.



**Using READ?**

ABOR1	<i>Aborts the measurement.</i>
CONF1 DEF,DEF, (@1)	<i>Configures the upper window to make a measurement using the current expected power and resolution settings.</i>
READ1?	<i>Initiates and retrieves the upper- window measurement.</i>

**Using INITiate and FETCh?**

ABOR1	<i>Aborts the measurement.</i>
CONF1 DEF,DEF, (@1)	<i>Configures the upper window to make a measurement using the current expected power and resolution settings.</i>
INIT1	<i>Sets the channel in a wait- for- trigger state.</i>
FETC1?	<i>Triggers the upper- window measurement, then retrieves the measurement reading.</i>

**Example 3 – Specifying the expected power parameter****NOTE**

The expected power parameter is only required for non-thermistor sensors. This parameter is provided for backward compatibility purpose and has no effect on thermistor sensors.

The previous example details the three optional parameters which can be used with the CONFIGure command and READ? query. The first optional parameter is used to enter an expected power value.

The following example uses the expected power level parameter to specify a value of -50 dBm. The resolution parameter is defaulted, leaving it at its current setting. The source list parameter specifies a Channel A measurement. The measurement is carried out on the upper window.

**Using READ?**

ABOR1	<i>Aborts the measurement.</i>
CONF1 -50,DEF, (@1)	<i>Configures the upper window to make a measurement using an expected power level of -50 dBm and the current resolution setting.</i>
READ1?	<i>Takes the upper- window measurement.</i>

## 1 Power Meter Remote Operation

Some finetuning of measurements can be performed using the `CONFigure` and `READ?` commands. For example, in the previous program segment, some finetuning can be performed by turning the trigger delay off.

```
1 ABOR1
2 CONF1 -50,DEF, (@1)
3 TRIG1:DEL:AUTO OFF
4 READ1?
```

### Using `INITiate` and `FETCh?`

<code>ABOR1</code>	<i>Aborts the measurement.</i>
<code>CONF1 -50,DEF, (@1)</code>	<i>Configures the upper window to make a measurement using an expected power level of -50 dBm and the current resolution setting.</i>
<code>INIT1</code>	<i>Sets the channel in a wait-for-trigger state.</i>
<code>FETC1? -50,DEF, (@1)</code>	<i>Triggers the upper-window measurement, then retrieves the measurement reading.</i>

Some finetuning of measurements can be carried out using the `CONFigure` command and `INITiate` and `FETCh?` commands. For example, in the above program segment some finetuning can be carried out by turning the trigger delay off.

```
1 ABOR1
2 CONF1 -50,DEF, (@1)
3 TRIG1:DEL:AUTO OFF
4 INIT1
5 FETC1? -50,DEF, (@1)
```

### Example 4 – Specifying the resolution parameter

The previous examples detailed the use of the expected value and source list parameters. The resolution parameter is used to set the resolution of the specified window.

The following example uses the resolution parameter to specify a resolution setting of 3. This setting represents three significant digits if the measurement suffix is W or %, and 0.01 dB if the suffix is dB or dBm (for further details on the resolution parameter, refer to the commands in [Chapter 3, “MEASurement Commands”](#)). Also, in this example, the expected power level and source list parameters are set to their default values. The expected power level is left unchanged at its current setting. As the source list parameter is the last specified parameter, you do not have to specify DEF.

### Using READ?

ABOR1	<i>Aborts the measurement.</i>
CONF1 DEF, 3	<i>Configures the upper window to make a measurement using the current settings of the expected power level and source list, and a resolution setting of 3.</i>
READ1?	<i>Initiates and retrieves the upper- window measurement.</i>

Some finetuning of the above program segment can be carried out, for example, by turning the trigger delay off, as shown below.

```

1 ABOR1
2 CONF1 DEF, 3
3 TRIG1:DEL:AUTO OFF
4 READ1?

```

### Using INITiate and FETCh?

ABOR1	<i>Aborts the measurement.</i>
CONF1 DEF, 3	<i>Configures the upper window to make a measurement using the current settings of the expected power and source list, and a resolution setting of 3.</i>
INIT1	<i>Sets the channel in a wait-for-trigger state.</i>
FETC1? DEF, 3	<i>Triggers the upper- window measurement, then retrieves the measurement reading.</i>

## 1 Power Meter Remote Operation

Some finetuning of the previous program segment can be carried out, for example, by turning the trigger delay off, as shown below.

```
1 ABOR1
2 CONF1 DEF, 3
3 TRIG1:DEL:AUTO OFF
4 INIT1
5 FETC1? DEF, 3
```

### Example 5 – Making a difference measurement

The following program segment queries the lower window to make a difference measurement of Channel A – Channel A. The expected power level and resolution parameters are defaulted, leaving them at their current settings. Some finetuning of the measurement is carried out by turning the trigger delay off.

#### Using READ?

```
ABOR1
CONF2:POW:AC:DIFF DEF,DEF, (@1), (@1)
TRIG1:DEL:AUTO OFF
READ2:POW:AC:DIFF?
READ2:POW:AC:DIFF? DEF,DEF, (@1), (@1) (A second READ? query is sent to
make a Channel A – Channel A measurement using fresh measurement
data.)
```

#### Using INITiate and FETCh?

```
ABOR1
CONF2:POW:AC:DIFF DEF,DEF, (@1), (@1)
TRIG1:DEL:AUTO OFF
INIT1
FETC2:POW:AC:DIFF?
FETC2:POW:AC:DIFF? DEF,DEF, (@1), (@1) (A second FETCh? query is sent
to make a Channel A – Channel A measurement using the current
measurement data.)
```

### Example 6 – Making a ratio measurement

The following program segment queries the lower window to make a ratio measurement of Channel A/A. The expected power level and resolution parameters are defaulted, leaving them at their current settings.

#### Using READ?

```
ABOR1
CONF2:POW:AC:RAT DEF,DEF, (@1) , (@1)
READ2:POW:AC:RAT?
READ2:POW:AC:RAT? DEF,DEF, (@1) , (@1) (A second READ? query is sent to
make a Channel A/Channel A ratio measurement using fresh measurement
data.)
```

#### Using INITiate and FETCh?

```
ABOR1
CONF2:POW:AC:RAT DEF,DEF, (@1) , (@1)
INIT1
FETC2:POW:AC:RAT?
FETC2:POW:AC:RAT? DEF,DEF, (@1) , (@1) (A second FETCh? query is sent to
make a Channel A/Channel A ratio measurement using the current
measurement data.)
```

## Using the lower-level commands

An alternative method in making measurements is to use the lower-level commands to set up the resolution and measurement type as follows:

```
DISPlay[:WINDow[1]|2]:NUMeric[1]|2:RESolution <numeric_value>
```

```
CALCulate[1]|2|3|4:MATH[:EXPRession] <string>
```

```
CALCulate[1]|2|3|4:RELative[:MAGNitude]:AUTO <boolean>|ONCE
```

The advantage of using the lower-level commands over the CONFIGure command is that they give you more precise control of the N432A. As shown in [Table 1-1](#), the CONFIGure command presets various states in the N432A. It may be likely that you do not want to preset these states.

### Example

The following example sets the measurement type to a single measurement and the resolution setting to 3 using the lower-level commands. The measurement is carried out on the lower window.

ABOR1	<i>Aborts the measurement.</i>
CALC2:MATH:EXPR "(SENS1)"	<i>Sets the lower window to a single measurement.</i>
DISP:WIND2:RES 3	<i>Sets the lower- window resolution setting to 3.</i>
INIT1	<i>Sets the channel in a wait-for- trigger state.</i>
FETC2?	<i>Triggers the lower- window measurement, then retrieves the measurement reading.</i>

## Setting the Bridge Resistance

A bridge resistance can be configured for the thermistor sensor to balance bridges.

The `[SENSe[1]:]BRESistance <character_data>` command allows you to specify a bridge resistance value of either 100  $\Omega$ , 200  $\Omega$ , 300  $\Omega$ , or 400  $\Omega$ .

To check the current setting of the bridge resistance, use the `[SENSe[1]:]BRESistance?` query.

For details, refer to “[`SENSe[1]:]BRESistance <character_data>`” on page 301.

## Setting the Measurement Frequency

You can set the frequency of the RF signal you are measuring by using the `[SENSe[1]:]FREQuency[:CW|:FIXed] <numeric_value>` command. Setting this value optimizes the accuracy and minimizes measurement uncertainty, especially when making comparative measurements between signals.

To check the current frequency setting, use the `[SENSe[1]:]FREQuency[:CW|:FIXed]? [MIN|MAX]` query.

For details, refer to “[`SENSe[1]:]FREQuency[:CW|:FIXed] <numeric_value>`” on page 331.

## Setting Offsets

### Channel offsets

Specifying a channel offset value allows the N432A to correct every measurement by this value to compensate for external gain or loss in your test setup. This gain or loss is applied to the measured power before any mathematical functions, display offsets, or relative functions are included.

You can use the

```
[SENSE[1]:]CORREction:GAIN2[:INPut][:MAGNitude]
<numeric_value> command to enter the channel offset value. Entering
this value automatically enables the channel offset. However, it can be
enabled or disabled using the [SENSE[1]:]CORREction:GAIN2:STATe
<boolean> command.
```

For details on these commands, refer to “[SENSE[1]:]CORREction:GAIN2 Commands” on page 327.

#### NOTE

You can only use GAIN2 for external losses/gains. GAIN1 is used specifically for calibration factors.

### Display offsets

A display offset is applied to the measured power after any channel offset or mathematical functions have been included (refer to [Figure 1-9](#) on page 52).

The display offset value can be entered using the

```
CALCulate[1]|2|3|4:GAIN[:MAGNitude] <numeric_value>
command. Entering this value automatically enables the display offset.
```

The CALCulate[1]|2|3|4:GAIN:STATe <boolean> command can be used to enable or disable the display offset.

For details on these commands, refer to “CALCulate[1]|2|3|4:GAIN Commands” on page 177.



**Example**

The following example program, in HP Basic, details how to use the channel and display offsets on the N432A making a Channel A/A ratio measurement.

The final result is:

$$\left( \left( \frac{A_{dBm} - 10}{A_{dBm} - 10} \right) - 20 \right)_{dB}$$

```

10 !Create I/O path name
20 ASSIGN @POWER TO 713
30 !Clear the N432A interface
40 CLEAR @POWER
50 !Set the N432A to a known state
60 OUTPUT @POWER;"*RST"
70 !Configure the N432A to make the measurement
80 OUTPUT @Power;"CONF:POW:AC:RAT 20DBM,2,(@1),(@1)"
90 !Set the measurement unit to dBm
100 OUTPUT @POWER;"UNIT:POW DBM"
110 !Set the N432A for a channel offset of -10 dB
120 OUTPUT @POWER;"SENS1:CORR:GAIN2 -10"
130 !Enable the gain correction
140 OUTPUT @POWER;"SENS1:CORR:GAIN2:STATE ON"
150 !Set the N432A for a display offset of -20 dB
160 OUTPUT @POWER;"CALC1:GAIN -20 DB"
170 PRINT "MAKING THE MEASUREMENT"
180 !Initiate the measurement
190 OUTPUT @Power;"INIT1:IMM"
200 ! ... and get the result
210 OUTPUT @Power;"FETC:POW:AC:RAT? 20DBM,2,(@1),(@1)"
220 ENTER @Power;Reading
230 !
240 PRINT "The measurement result is ";Reading;"dB."
250 END

```

## Using frequency-dependent offset tables

This section describes how to use frequency-dependent offset tables to compensate for frequency-related changes in the response of your test system. Note that when selected, frequency-dependent offset corrections are applied IN ADDITION to any correction for the sensor frequency response. The N432A is capable of storing 10 frequency-dependent offset tables with a maximum of 80 frequency points each.

Below illustrates how frequency-dependent offset tables operate:

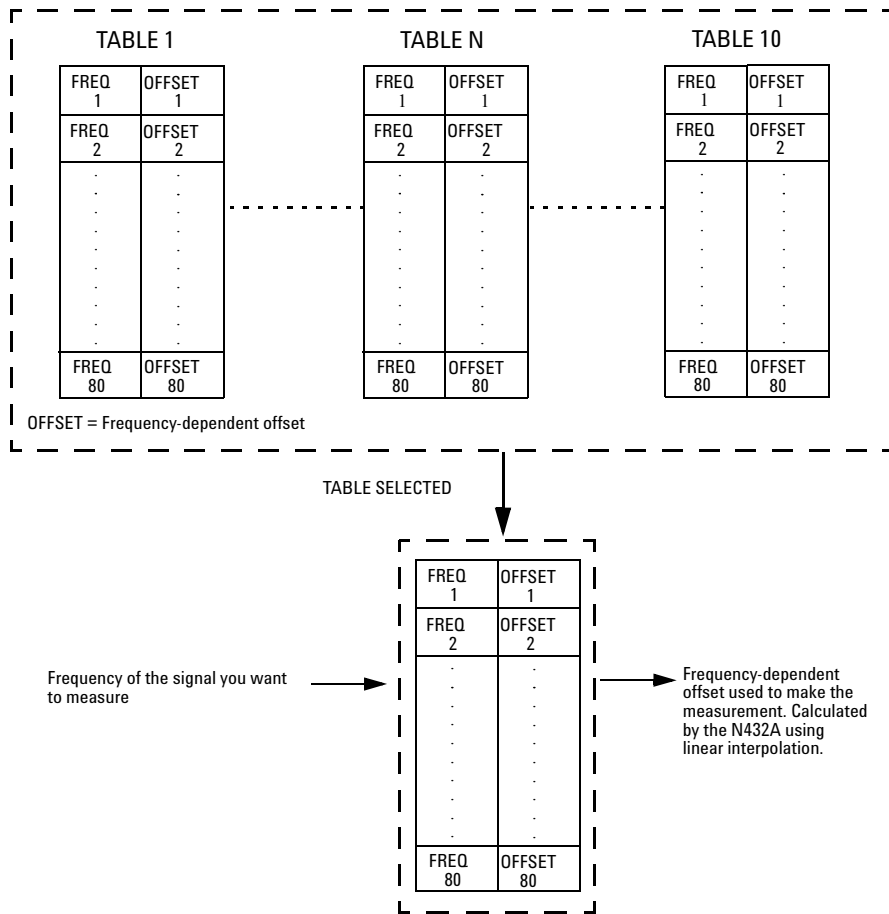


Figure 1-8 Frequency-dependent offset tables

## Editing frequency-dependent offset tables

It is not possible to delete any of the existing frequency-dependent offset tables or create any additional tables. However, the existing tables can be edited using the MEMORY subsystem. To do this, perform the following procedure:

- 1 Select one of the existing tables using the `MEMory[:METer][1]:TABLE:SElect <character_data>` command. For information on renaming frequency-dependent offset tables, refer to “Renaming frequency-dependent offset tables” on page 36. For information on the currently stored tables in the N432A, refer to “Listing the stored frequency-dependent offset tables” on page 36.
- 2 Enter the frequency data using the `MEMory[:METer][1]:TABLE:FREQuency <numeric_value>{,<numeric_value>}` command.
- 3 Enter the offset values using the `MEMory[:METer][1]:TABLE:GAIN[:MAGNitude] <numeric_value>{,<numeric_value>}` command.

The following table shows how entries in the frequency lists correspond with entries in the offset lists:

Frequency	Offset
Frequency 1	Offset 1
"	"
Frequency 80	Offset 80

### NOTE

- Ensure that the frequency points you use cover the frequency range of the signals you want to measure. If you measure a signal with a frequency outside the frequency range defined in the frequency-dependent offset table, then the N432A uses the highest or lowest frequency point in the table to calculate the offset.
- To make subsequent editing of a frequency-dependent offset table simpler, it is recommended that you retain a copy of your data in a program.

### Listing the stored frequency-dependent offset tables

To list the frequency-dependent offset tables currently stored in the N432A, use the `MEMory[:METer][1]:CATalog:TABLE?` query.

Note that all tables are listed, including sensor calibration tables.

The N432A returns the data in the form of the following example:

```
560,8020,"Offset_1,TABL,220","Offset_2,TABL,340" . . .
```

For details, refer to “[MEMory\[:METer\]\[1\]:CATalog:TABLE?](#)” on page 256.

### Renaming frequency-dependent offset tables

To rename a frequency-dependent offset table, use the `MEMory[:METer][1]:TABLE:MOVE <character_data>,<character_data>` command.

The first `<character_data>` parameter contains the existing table name, while the second details the new table name.

The following rules apply to naming frequency-dependent offset tables:

- The name must consist of not more than 12 characters.
- All characters must be upper-case or lower-case alphabetic characters, or numeric (0 to 9), or an underscore (\_).
- No spaces are allowed in the name.

For details, refer to “[MEMory\[:METer\]\[1\]:TABLE:MOVE <character\\_data>,<character\\_data>](#)” on page 278.

### Reviewing table data

To review the data stored in a frequency-dependent offset table, use the following commands:

```
MEMory[:METer][1]:TABLE:SElect <character_data>  
Select a frequency- dependent offset table of the specified name. The  
MEMory[:METer][1]:TABLE:SElect? query returns the name of the  
currently selected table.
```

```
MEMory[:METer][1]:TABLE:FREQuency:POINts?  
Returns the number of frequency points for the currently selected table.
```

MEMory[:METer][1]:TABLE:FREQuency?

*Returns a list of frequency points (in Hz) for the currently selected table.*

MEMory[:METer][1]:TABLE:GAIN[:MAGNitude]:POINTs?

*Returns the number of offset points for the currently selected table.*

MEMory[:METer][1]:TABLE:GAIN[:MAGNitude]?

*Returns a list of offsets for the currently selected table.*

Refer to the individual command under “MEMory[:METer][1]:TABLE Commands” on page 270 for details.

### Modifying table data

If you need to modify the frequency and offset data stored in a frequency-dependent offset table, you need to resend the complete data lists.

If you have retained the original data in a program, edit the program and resend the data.

## Selecting frequency-dependent offset tables

After you have edited a frequency-dependent offset table, you can select it using the following command:

```
[SENSe[1]:]CORRection:CSET2[:SElect] <string>
```

To determine which frequency-dependent offset table is currently selected, use the following query:

```
[SENSe[1]:]CORRection:CSET2[:SElect]?
```

Refer to “[SENSe[1]:]CORRection:CSET2[:SElect] <string>” on page 314 for details.

## Enabling a frequency-dependent offset table

To enable a frequency-dependent offset table, use the following command:

```
[SENSe[1]:]CORRection:CSET2:STATe ON
```

If you enable this command and no frequency-dependent offset table has been selected, error -221, "Settings conflict" occurs and this command remains disabled.

Refer to "[SENSe[1]:]CORRection:CSET2:STATe <boolean>" on page 318 for details.

## Making the measurement

The following program segments provide the examples on how to use the frequency-dependent offset tables in measuring power.

### INITiate example

```
ABOR  
CONF:POW:AC DEF,1,(@1)  
SENS:CORR:CSET2:SEL "Offset1"  
SENS:CORR:CSET2:STAT ON  
SENS:FREQ 500KHZ  
INIT  
FETC?
```

### READ? example

```
ABOR  
CONF:POW:AC DEF,2,(@1)  
SENS:CORR:CSET2:SEL "Offset1"  
SENS:CORR:CSET2:STAT ON  
SENS:FREQ 500KHZ  
READ?
```

**NOTE**

- If the measurement frequency does not correspond directly to a frequency in the frequency-dependent offset table, the N432A calculates the offset using linear interpolation.
  - If you enter a frequency outside the frequency range defined in the frequency-dependent offset table, then the N432A uses the highest or lowest frequency point in the table to set the offset.
  - To determine the value of the offset being used by the N432A to make a measurement, use the following query:  
`[SENSE[1]:]CORREction:FDOFfset|GAIN4[:INPut]  
[:MAGNitude]?`  
The response may be an interpolated value.
-

## Setting Measurement Limits

You can configure the N432A to detect when a measurement is outside of a predefined upper and/or lower limit value.

Limits are boundaries set for a certain power range and can be applied to the power, ratio, or difference measurement.

The N432A can be configured to verify the power being measured against an upper and/or lower limit value. The range of values that can be set for lower and upper limits is  $-150.00$  dBm to  $+230.00$  dBm. The default upper limit is  $+90.00$  dBm and the default lower limit is  $-90.00$  dBm.

The range of values that can be set for the upper and lower limits and the default values depend on the measurement units in the currently selected measurement line, as listed in the following table:

**Table 1-2** Range of values for window limits

Window unit			Default	
	Maximum	Minimum	Maximum	Minimum
dB	+200 dB	-180 dB	60 dB	-120 dB
dBm	+230 dBm	-150 dBm	90 dBm	-90 dBm
%	999.9 X%	100.0 a%	100.0 M%	100.0 p%
W	100.000 XW	1.000 aW	1.000 MW	1.000 pW

To enter a value for the upper test limit or query the current upper limit setting for the specified window/measurement, use the `CALCulate[1]|2|3|4:LIMit:UPPER[:DATA] <numeric_value>` command and `CALCulate[1]|2|3|4:LIMit:UPPER[:DATA]? [MIN|MAX]` query respectively.

To enter a value for the lower test limit or query the current lower limit setting for the specified window/measurement, use the `CALCulate[1]|2|3|4:LIMit:LOWer[:DATA] <numeric_value>` command and `CALCulate[1]|2|3|4:LIMit:LOWer[:DATA]? [MIN|MAX]` query respectively.



You can enable or disable the test limits for the specified window/measurement using the `CALCulate[1]|2|3|4:LIMit:STATe <boolean>` command.

For details on the individual command, refer to “[CALCulate\[1\]|2|3|4:LIMit Commands](#)” on page 181.

## Checking for limit failures

There are two methods to check for limit failures:

- Using the `CALCulate[1]|2|3|4:LIMit:FAIL?` and `CALCulate[1]|2|3|4:LIMit:FCOunt?` queries, or
- Using the `STATus` command subsystem

### Using the CALCulate queries

Using the `CALCulate` queries to check the limit failures for the specified window/measurement would return the following result formats:

`CALCulate[1]|2|3|4:LIMit:FAIL?` *Returns 1 if there has been one or more limit failures or 0 if there have been no limit failures.*

`CALCulate[1]|2|3|4:LIMit:FCOunt?` *Returns the total number of limit failures.*

For more information on these queries, refer to “[CALCulate\[1\]|2|3|4:LIMit:FAIL?](#)” on page 185 and “[CALCulate\[1\]|2|3|4:LIMit:FCOunt?](#)” on page 186.

#### NOTE

If `TRIGger[1]:DELay:AUTO <boolean>` is set to ON, then the number of failures returned by `CALCulate[1]|2|3|4:LIMit:FCOunt?` is affected by the current filter settings.

### Using the STATus subsystem

If using GPIB, you can use the STATus subsystem to generate an SRQ to interrupt your program when a limit failure occurs. This is a more efficient method than using CALCulate, as you do not need to check the limit failures after every power measurement.

Refer to “[Status Reporting](#)” on page 53 and “[STATus Subsystem](#)” on page 359 for further information.

## Using the Calibration/Correction Factors

This section provides the information on how to set frequency-specific calibration factors and to use sensor calibration tables.

### Setting frequency-specific calibration factors

If you are making most of your measurements at a single frequency, or in a narrow range of frequencies, entering a specific calibration factor is a more effective method than using a sensor calibration table. Only a minimal amount of data entry is required.

Use the `[SENSE[1]:]CORRection:CFACTOR|GAIN[1][:INPut][:MAGNitude]? query` to check the current calibration factor setting. Verify that the value returned matches the calibration factor of the sensor listed in the calibration factor table on the sensor body.

A gain correction value for the calibration factor can be entered using the `[SENSE[1]:]CORRection:CFACTOR|GAIN[1][:INPut][:MAGNitude]<numeric_value>` command. The N432A corrects every measurement by this factor to compensate for the gain. A gain correction of 1% to 150% can be specified.

For details, refer to [“\[SENSE\[1\]:\]CORRection:CFACTOR|GAIN\[1\]\[:INPut\]\[:MAGNitude\]<numeric\\_value>”](#) on page 308.

### Using sensor calibration tables

If you are making measurements on a wide range of signal frequencies, using a sensor calibration table is more effective than entering an individual calibration factor, as you only need to enter the frequency of the signal you are measuring. The N432A automatically selects and applies the calibration factor from the selected table.

The N432A is capable of storing 20 sensor calibration tables, each containing up to 80 frequency points. The N432A is supplied with a set of two predefined sensor calibration tables plus a “100%” default table.

## Editing sensor calibration tables

It is not possible to delete any of the existing calibration tables or create any additional tables. However, the existing tables can be edited using the MEMORY subsystem. To do this, perform the following procedure:

- 1 Select one of the existing tables using the `MEMory[:METer][1]:TABLE:SElect <character_data>` command.  
For information on renaming the sensor calibration tables, refer to “Renaming sensor calibration tables” on page 45. For information on the currently stored tables in the N432A, refer to “Listing stored sensor calibration tables” on page 45.
- 2 Enter the frequency data using the `MEMory[:METer][1]:TABLE:FREQuency <numeric_value>{,<numeric_value>}` command.
- 3 Enter the calibration factors using the `MEMory[:METer][1]:TABLE:GAIN[:MAGNitude] <numeric_value>{,<numeric_value>}` command.

The following table shows how entries in the frequency lists correspond with entries in the calibration factor lists:

Frequency	Calibration factor
–	Reference calibration factor
Frequency 1	Calibration factor 1
”	”
Frequency 80	Calibration factor 80

### NOTE

The number of frequency points must be one less than the number of calibration factor data points. The first calibration factor data point is the reference calibration factor, and each subsequent data point is a calibration factor in the sensor calibration table.

**NOTE**

- Ensure that the frequency points you use cover the frequency range of the signals you want to measure. If you measure a signal with a frequency outside the frequency range defined in the sensor calibration table, then the N432A uses the highest or lowest frequency point in the table to set the calibration factor.
- To make subsequent editing of a sensor calibration table simpler, it is recommended that you retain a copy of your data in a program.

**Listing stored sensor calibration tables**

To list the sensor calibration tables currently stored in the N432A, use the `MEMory[:METer][1]:CATalog:TABLE?` query.

Note that all tables are listed, including frequency-dependent offset tables.

The N432A returns the data in the form of the following example:

```
198,4664,"DEFAULT,TABL,14","478A,TABL,86",
"8478B,TABL,98".....
```

For details, refer to [“MEMory\[:METer\]\[1\]:CATalog:TABLE?”](#) on page 256.

**Renaming sensor calibration tables**

To rename a sensor calibration table, use the

```
MEMory[:METer][1]:TABLE:MOVE
<character_data>,<character_data>
```

command.

The first `<character_data>` parameter contains the existing table name, while the second details the new table name.

The following rules apply to naming sensor calibration tables:

- The name must consist of not more than 12 characters.
- All characters must be upper-case or lower-case alphabetic characters, or numeric (0 to 9), or an underscore (\_).
- No spaces are allowed in the name.

For details, refer to [“MEMory\[:METer\]\[1\]:TABLE:MOVE <character\\_data>,<character\\_data>”](#) on page 278.

### Reviewing table data

To review the data stored in a sensor calibration table, use the following commands:

```
MEMory[:METer][1]:TABLE:SElect <character_data>
```

*Select a sensor calibration table of the specified name. The*

*MEMory[:METer][1]:TABLE:SElect? query returns the name of the currently selected table.*

```
MEMory[:METer][1]:TABLE:FREquency:POINTs?
```

*Returns the number of frequency points for the currently selected table.*

```
MEMory[:METer][1]:TABLE:FREquency?
```

*Returns a list of frequency points (in Hz) for the currently selected table.*

```
MEMory[:METer][1]:TABLE:GAIN[:MAGNitude]:POINTs?
```

*Returns the number of calibration factor points (including the reference calibration factor) for the currently selected table.*

```
MEMory[:METer][1]:TABLE:GAIN[:MAGNitude]?
```

*Returns a list of calibration factors for the currently selected table.*

Refer to the individual command under “[MEMory\[:METer\]\[1\]:TABLE Commands](#)” on page 270 for details.

### Modifying table data

If you need to modify the frequency and calibration factor data stored in a sensor calibration table, you need to resend the complete data lists.

If you have retained the original data in a program, edit the program and resend the data.

## Selecting sensor calibration tables

After you have edited a sensor calibration table, you can select it using the following command:

```
[SENSe[1]:]CORRection:CSET[1][:SElect] <string>
```

To determine which sensor calibration table is currently selected, use the following query:

```
[SENSe[1]:]CORRection:CSET[1][:SElect]?
```

Refer to “[SENSe[1]:]CORRection:CSET[1][:SElect] <string>” on page 312 for details.

## Enabling sensor calibration tables

To enable a sensor calibration table, use the following command:

```
[SENSe[1]:]CORRection:CSET1:STATe ON
```

If you enable this command and no sensor calibration table has been selected, error -221, “Settings conflict” occurs and this command remains disabled.

Refer to “[SENSe[1]:]CORRection:CSET[1]:STATe <boolean>” on page 316 for details.

## Making the measurement

The following program segments provide the examples on how to use the calibration factors in measuring power.

### **INITiate example (using a frequency-specific calibration factor)**

```
ABOR
CONF:POW:AC DEF,1,(@1)
SENS:CORR:GAIN 60
SENS:FREQ 500KHZ
INIT
FETC?
```

### **READ? example (using a sensor calibration table)**

```
ABOR
CONF:POW:AC DEF,2,(@1)
SENS:CORR:CSET1:SEL "Sensor1"
SENS:CORR:CSET1:STAT ON
SENS:FREQ 500KHZ
READ?
```

### NOTE

- If the measurement frequency does not correspond directly to a frequency in the sensor calibration table, the N432A calculates the calibration factor using linear interpolation.
- If you enter a frequency outside the frequency range defined in the sensor calibration table, the N432A uses the highest or lowest frequency point in the sensor calibration table to set the calibration factor.
- To determine the calibration factor being used by the N432A to make a measurement, use the following query:

```
[SENSE[1]:]CORREction:CFACTOR|GAIN[1][:INPut]
[:MAGNitude]? [MIN|MAX]
```

The response may be an interpolated value.

---



## Optimizing the N432A Performance

This section discusses the factors that influence the speed of operation of the N432A.

The following factors are those which have the greatest effect upon measurement speed (in no particular order):

- Trigger mode (for example, free run, triggered free run, or single shot)
- Output format: ASCii or REAL
- Measurement unit
- Command used to take a measurement

The following paragraphs provide a brief description of the above factors and how they are controlled from SCPI.

### Trigger mode

The N432A has a flexible triggering system. For simplicity, it can be described as having three modes:

- Free Run: When the N432A is in the free run mode, it continuously takes measurements. A measurement is in free run when `INITiate[1]:CONTInuous <boolean>` is set to ON and `TRIGger[1]:SOURce BUS|HOLD|IMMEdiate` is set to IMMEdiate.
- Triggered Free Run: When the N432A is in the triggered free run or continuous trigger mode, it takes a new measurement each time a trigger event is detected. A measurement is in triggered free run/continuous trigger when `INITiate[1]:CONTInuous <boolean>` is set to ON and `TRIGger[1]:SOURce BUS|HOLD|IMMEdiate` is not set to IMMEdiate.
- Single Shot: When the N432A is in the single shot mode, it takes a new measurement when a trigger event is detected and then returns to the idle state. A measurement is in single shot when `INITiate[1]:CONTInuous <boolean>` is set to OFF. Note that a measurement can take several internal triggers depending on the filter settings. Refer to “`TRIGger[1]:DELay:AUTO <boolean>`” on page 433 for further information.

### NOTE

A trigger event can be any of the following:

- The input signal meeting the trigger level criteria.
- Auto-level triggering being used.
- A `TRIGger[1][:IMMediate]` or `GET` command being sent.

### Trigger with delay

This can be achieved using the same sequences above (apart from the second) with `TRIGger[1]:DELAy:AUTO <boolean>` set to `ON`. Also, the `MEASure?` query operates in the trigger with delay mode.

In the trigger with delay mode, a measurement is not completed until the N432A filter is full. In this way, the reading returned is guaranteed to be settled. In all other modes, the result returned is simply the current result from the filter and may or may not be settled. This depends on the current length of the filter and the number of readings that have been taken since a change in power level.

## Output format

The N432A has two output formats for measurement results: `ASCIi` and `REAL`. These formats are selected using the `FORMat[:READings][:DATA] <character_data>` command. When the format is set to `REAL`, the returned result is in the IEEE-754 floating-point format (note that the byte order can be changed using the `FORMat[:READings]:BORDER <character_data>` command).

For further information on the `FORMat` commands, refer to [Chapter 7, “FORMat Subsystem”](#) on page 234.

## Units

The N432A can output results in either linear or log units. The internal units are linear, therefore optimal performance is achieved when the result outputs are also in linear units (since the overhead of performing a log function is removed).

## Measurement commands

In the free run mode, `FETCh?` must be used to return a result.

In other trigger modes, there are a number of queries which can be used, for example, `MEASure?`, `READ?`, `FETCh?`. Note that `MEASure?` and `READ?` are compound commands as they perform a combination of other lower-level commands. Typically, the best speed performance is achieved using the low-level commands directly.

# How Measurements are Calculated

The following figure illustrates how measurements are calculated. It shows the order in which the various N432A functions are implemented in the measurement calculation.

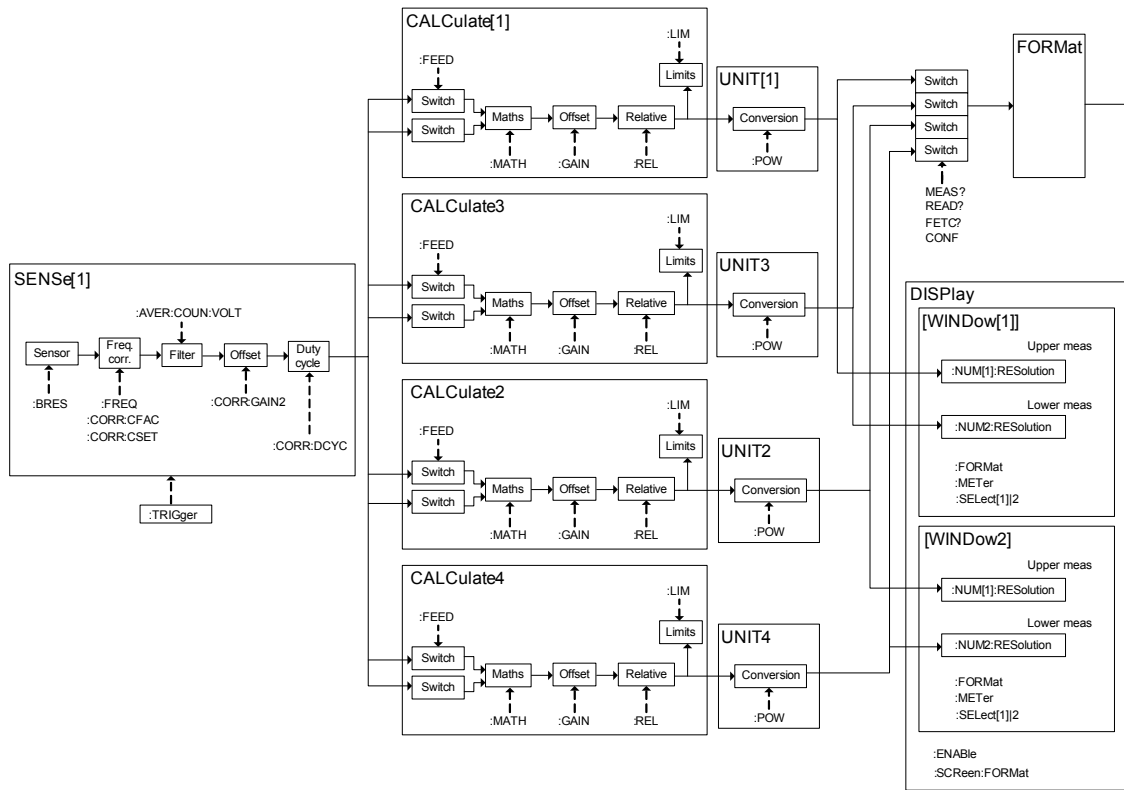


Figure 1-9 How measurements are calculated

## Status Reporting

Status reporting is used to monitor the N432A to determine when events have occurred. Status reporting is accomplished by configuring and reading status registers.

The N432A has the following main registers:

- Status Byte Register
- Standard Event Register
- Operation Status Register
- Questionable Status Register
- Device Status Register

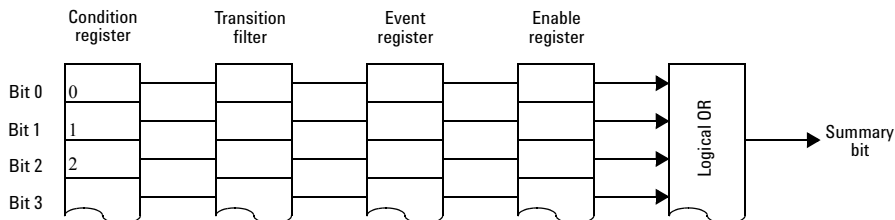
There are other registers that exist “behind” the main registers, and are described later in this chapter.

Status Byte and Standard Event registers are read using the IEEE-488.2 common commands.

Operation, Questionable, and Device Status registers are read using the SCPI `STATus` command subsystem.

### General status register model

The generalized status register model shown below is the building block of the SCPI status system. This model consists of a condition register, a transition filter, an event register, and an enable register. A set of these registers is called a status group.



**Figure 1-10** General status register model

When a status group is implemented in an instrument, it always contains all of the component registers. However, there is not always a corresponding command to read or write to every register.

### **Condition register**

The condition register continuously monitors the hardware and firmware status of the N432A. There is no latching or buffering for this register; it is updated in real time. Condition registers are read-only.

### **Transition filter**

The transition filter specifies which type of changes to the bit state in the condition register will set corresponding bits in the event register. Transition filter bits may be set for positive transitions (PTR), negative transitions (NTR), or both. Positive transition will cause the corresponding bit in the event register to be set when the condition bit changes from 0 to 1. Negative transition will cause the corresponding bit in the event register to be set when the condition bit changes from 1 to 0. Setting both positive and negative transitions will cause the corresponding bit in the event register to be set whenever the condition bit changes. Clearing both the positive and negative transition filters disables the corresponding bit in the event register to be set. Transition filters are read-write. They are unaffected by clear status (\*CLS) or queries. After STATus:PRESet, the NTR register is set to 0 and all bits of the PTR are set to 1.

### **Event register**

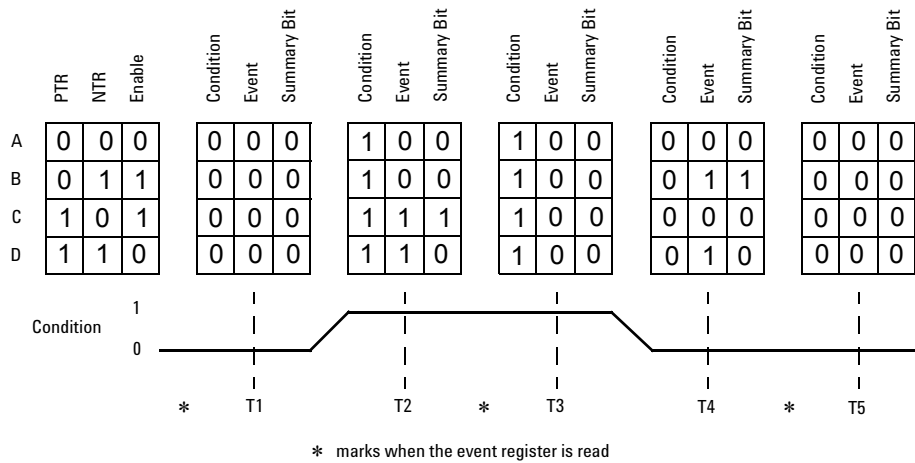
The event register latches transition events from the condition register as specified by the transition filter. Bits in the event register are latched and once the bits are set, they will remain set until they are cleared by a query or clear status (\*CLS). There is no buffering, therefore, while an event bit is set, subsequent events corresponding to that bit are ignored. Event registers are read-only.

### **Enable register**

The enable register specifies which bits in the event register can generate a summary bit. The instrument logically ANDs corresponding bits in the event and enable registers, and ORs all the resulting bits to obtain a summary bit. Enable registers are read-write. They are not affected by clear status (\*CLS) or querying the enable registers. There is always a command to read and write to the enable register of a particular status group.

**Example sequence**

The figure below illustrates the response of a single bit position in a typical status group for various settings. The changing state of the condition in question is shown at the bottom of the figure. A small binary table shows the state of the chosen bit in each status register at selected times T1 to T5.



**Figure 1-11** Typical status register bit changes

Consider **Case C**, where the positive transition filter is set to 1 and the negative transition filter is set to 0. This configures the N432A to set the corresponding bit in the event register whenever the condition bit changes from 0 to 1. The enable register is set to 1 to enable the summary bit to be generated each time there is a change in the event register.

At time **T1**, the condition bit is 0. Since there is no changes to the condition bit at this time, no corresponding bit in the event register will be set and the summary bit is 0.

At time **T2**, the condition bit changes from 0 to 1. Since the positive transition filter is set to detect condition bit changes from 0 to 1, the corresponding bit in the event register will be set to 1. The enable register is set to 1, which means that the summary bit will also be set to 1 whenever any bit in the event register is set to 1.

At time **T3**, the condition bit remains 1. The event register is cleared by a query. Hence, the event register bit and summary bit are set to 0.

## 1 Power Meter Remote Operation

At time **T4**, the condition bit changes from 1 to 0. Since the positive transition filter is set to detect condition bit changes from 0 to 1, the corresponding bit in the event register will be set to 0, signifying no event has been logged. The summary bit is set to 0 as no bit is set in the event register.

At time **T5**, the condition bit remains 0. Since there is no changes to the condition bit at this time, no corresponding bit in the event register will be set and the summary bit is 0.

Consider **Case D**, where the positive and negative transition filters are set to 1. This configures the N432A to set the corresponding bit in the event register whenever there are changes to the condition bit. The enable register is set to 0 to disable the summary bit to be generated.

At time **T1**, the condition bit is 0. Since there is no changes to the condition bit at this time, no corresponding bit in the event register will be set and the summary bit is 0.

At time **T2**, the condition bit changes from 0 to 1. Since the positive and negative transition filters are set to detect any changes to the condition bit, the corresponding bit in the event register will be set to 1. The enable register is set to 0, which means that the summary bit will not be set.

At time **T3**, the condition bit remains 1. The event register is cleared by a query. Hence, the event register bit and summary bit are set to 0.

At time **T4**, the condition bit changes from 1 to 0. Since the positive and negative transition filters are set to detect any changes to the condition bit, the corresponding bit in the event register will be set to 1, signifying an event has been logged. The summary bit is 0 as the enable register is set to 0.

At time **T5**, the condition bit remains 0. The event register is cleared by a query. Hence, the event register bit and summary bit are set to 0.



## How to use registers

There are two methods to access the information in status groups:

- the polling method, or
- the service request (SRQ) method. (GPIB mode only)

Use the polling method when:

- your language/development environment does not support SRQ interrupts.
- you want to write a simple, single-purpose program, and do not want to add the complexity of setting an SRQ handler.

Use the SRQ method when you:

- need time-critical notification of changes.
- are monitoring more than one device which supports SRQ interrupts.
- need to have the controller do something else while it is waiting.
- cannot afford the performance penalty inherent to polling.

## Condition polling method

In this polling method, the N432A has a passive role. It only informs the controller that conditions have changed when the controller queries. When you monitor a condition with the polling method, you must:

- 1 Determine which register contains the bit that monitors the condition.
- 2 Send the unique query that reads the register.
- 3 Examine the bit to see if the condition has changed.

The polling method works well if you do not need to know about the changes the moment they occur. The SRQ method is more effective if you must know immediately when a condition changes. Detecting an immediate change in a condition using the polling method requires your program to continuously read the registers at very short intervals. This is not particularly efficient and there is a possibility that an event may be missed.

For an example on measurement polling, refer to [“Measurement Polling Example Using VEE”](#) on page 457.

### SRQ method

When a bit of the Status register is set and has been enabled to assert SRQ (\*SRE command), the N432A sets the GPIB SRQ line true. This interrupt can be used to interrupt your program, suspend its current operation, and find out what service the N432A requires. Refer to your PC and language manuals for information on how to program the PC to respond to the interrupt.

To allow any of the Status register bits to set the SRQ line true, you must enable the appropriate bit(s) with the \*SRE command, for example, if your application requires an interrupt whenever a message is available in the output queue (Status register bit 4, decimal value 16). To enable bit 4 to assert SRQ, use command \*SRE 16.

#### NOTE

You can determine which bits are enabled in the Status register using \*SRE?. This command returns the decimal-weighted sum of all the bits.

#### Procedure

- Send a bus device clear message.
- Clear the event registers with the \*CLS (clear status) command.
- Set the \*ESE (standard event register) and \*SRE (status byte register) enable masks.
- Enable the IEEE-488 SRQ interrupt of your bus controller.

#### Examples

The following two examples are written in HP BASIC and illustrate possible uses for SRQ. In both cases, it is assumed that the N432A has been zeroed.

##### Example 1

```
10 ! Program to generate an SRQ when a channel A sensor
20 ! connect or disconnect occurs
30 !
40 ASSIGN @Pm TO 713 ! Power meter GPIB address
50 ON ON INTR 7 GOTO Srq_i! Define service request handler
60 CLEAR @Pm ! Selective device clear
```

```

70   OUTPUT @Pm;"*CLS;*RST" ! Clear registers and reset meter
80   !
90   ! Configure the device status register so that a sensor
100  ! connect or disconnect on channel A will cause an SRQ.
110  !
120  OUTPUT @Pm;"STAT:DEV:ENAB 2"
130  OUTPUT @Pm;"STAT:DEV:NTR 2"
140  OUTPUT @Pm;"STAT:DEV:PTR 2"
150  OUTPUT @Pm;"*SRE 2"
160  !
170  ENABLE INTR 7;2 ! Enable an SRQ to cause an interrupt
180  LOOP           ! Idle loop
190  ! Forever
200  END LOOP
210  !
220  ! When an SRQ is detected, the following routine will
    service it.
230  !
240  Srq_i:        !
250  St=SPOLL(@Pm) ! Serial Poll (reads status byte)

260  IF BIT(St,1)=1 THEN ! Device status reg bit set ?
270  OUTPUT @Pm;"STAT:DEV:EVEN?" ! Yes , read register

280  ENTER @Pm;Event           ! (this also clears it)

290  OUTPUT @Pm;"STAT:DEV:COND?"
300  ENTER @Pm;Cond
310  IF Cond=0 THEN
320  PRINT "Sensor disconnected"
330  ELSE
340  PRINT "Sensor connected"
350  END IF
360  END IF
370  GOTO 170                   ! Return to idle loop
380  END

```

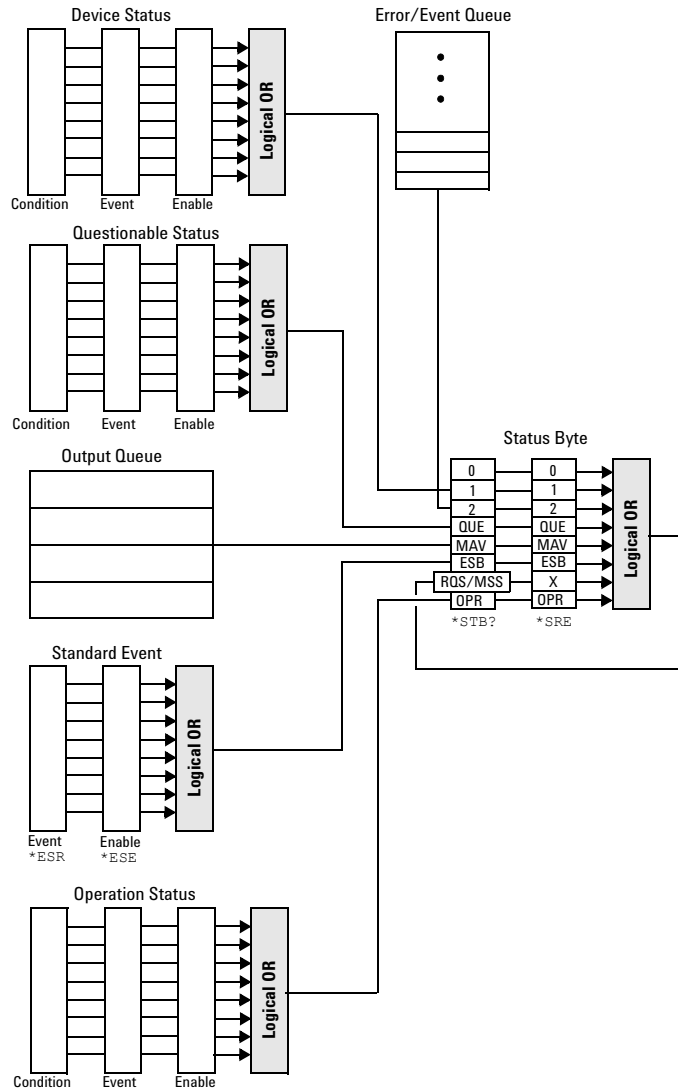
### Example 2

```
10      ! Program to generate an SRQ when an overlimit
20      ! condition occurs.
30      !
40      ASSIGN @Pm TO 713      ! Power meter GPIB address
50      ON INTR 7 GOTO Srq_i   ! Define service request handler
60      CLEAR @Pm             ! Selective device clear
70      OUTPUT @Pm;"*CLS"     ! Clear registers
80      OUTPUT @Pm;"SYST:PRES" ! Preset meter
90      !
100     ! Set upper limit to 2 dBm and configure the operation
status
110     ! so that an overlimit condition will cause an SRQ.
120     !
130     OUTPUT @Pm;"CALC:LIM:UPP 2DBM"
140     OUTPUT @Pm;"CALC:LIM:STAT ON"
150     OUTPUT @Pm;"STAT:OPER:PTR 4096"
160     OUTPUT @Pm;"STAT:OPER:ENAB 4096"
170     OUTPUT @Pm;"*SRE 128"
180     !
190     ENABLE INTR 7;2      ! Enable an SRQ to cause an interrupt
200     LOOP                 ! Idle loop
210     ! Forever
220     END LOOP
230     !
240     ! When an SRQ is detected, the following routine will
service it.
250     !
260     Srq_i:               !
270     St=SPOLL(@Pm) ! Serial Poll (reads status byte)
280     IF BIT(St,7)=1 THEN      ! Operation status bit set?
290         OUTPUT @Pm;"STAT:OPER?"! Yes , read register
300         ENTER @Pm;Oper      ! (this also clears it)
310         OUTPUT @Pm;"STAT:OPER:ULF?"
320         ENTER @Pm;Ulf
330         IF Ulf=2 THEN PRINT "Over limit detected"
340     END IF
350     GOTO 190                ! Return to idle loop
360     END
```

### Status registers

The Status system in the N432A is shown in [Figure 1-12](#). The Operation Status, Questionable Status, and Device Status groups are 16-bit wide, while the Status Byte and Standard Event groups are 8-bit wide. In all 16-bit groups, the most significant bit (bit 15) is not used and is always set to 0.

# 1 Power Meter Remote Operation



**Figure 1-12** Status system

### Status Byte summary register

The Status Byte summary register reports conditions from other status registers. Query data waiting in the N432A output buffer is immediately reported through the “Message Available” bit (bit 4). Clearing an event register clears the corresponding bits in the Status Byte summary register. Reading all messages in the output buffer, including any pending queries, clears the Message Available bit.

**Table 1-3** Bit definitions – Status Byte register

Bit	Weight/Decimal value	Definition
0	1	Not used (always set to 0)
1	2	Device Status register summary bit One or more bits are set in the Device Status register (bits must be “enabled” in the enable register)
2	4	Error/Event queue
3	8	Questionable Status register summary bit One or more bits are set in the Questionable Status register (bits must be “enabled” in the enable register)
4	16	Message available Data is available in the N432A output buffer
5	32	Standard Event register summary bit One or more bits are set in the Standard Event register (bits must be “enabled” in the enable register)
6	64	Request for service The N432A is requesting for service (serial poll)
7	128	Operation Status register summary bit One or more bits are set in the Operation Status register (bits must be “enabled” in the enable register)

Particular bits in the Status Byte register are cleared when:

- The Standard Event, Questionable Status, Operation Status, and Device Status are queried.
- The error/event queue becomes empty.
- The output queue becomes empty.

The Status Byte enable register (`*SRE <NRf>`, service request enable) is cleared when you:

- cycle the instrument power.
- execute an `*SRE 0` command.

### Using `*STB?` to read the Status Byte

`*STB?` (status byte query) is similar to a serial poll except that it is processed like any other N432A command. The `*STB?` query returns the same result as an IEEE-488 serial poll except that the request service bit (bit 6) is *not* cleared if a serial poll has occurred. The `*STB?` query is not handled automatically by the IEEE-488 bus interface hardware and the query is executed only after previous commands have completed. Using the `*STB?` query does not clear the Status Byte summary register.

### Standard Event register

The Standard Event register reports the following types of instrument events: power-on detected, command and syntax errors, command execution errors, self-test errors, query errors, or when an overlapped command completes following an `*OPC` command. Any or all of these conditions can be reported in the Standard Event summary bit through the enable register. You must write a decimal value using the `*ESE <NRf>` (event status enable) command to set the enable register mask.



**Table 1-4** Bit definitions – Standard Event register

Bit	Weight/Decimal value	Definition
0	1	Operation complete All overlapped commands following an *OPC command have completed
1	2	Not used (always set to 0)
2	4	Query error A query error occurred
3	8	Device-dependent error A device error occurred
4	16	Execution error An execution error occurred
5	32	Command error A command syntax error occurred
6	64	User request
7	128	Power on Power has been turned off and on since the last time the event register was read or cleared

The Standard Event register is cleared when you:

- send a \*CLS (clear status) command.
- query the event register using the \*ESR? (event status register) command.

The Standard Event enable register is cleared when you:

- cycle the N432A power.
- execute an \*ESE 0 command.

### Questionable Status register

The Questionable Status register set provides information on the quality of the N432A measurement results. Any or all of these conditions can be reported in the questionable data summary bit through the enable register. You must write a value using the `STATUS:QUESTIONABLE:ENABLE` command to set the enable register mask.

The following bits in this register set are used by the N432A.

**Table 1-5** Bit definitions – Questionable Status register

Bit	Weight/Decimal value	Definition
0 to 2	–	Not used
3	8	POWER summary
4 to 7	–	Not used
8	256	CALibration summary
9	512	Power-on self-test
10 to 15	–	Not used (bit 15 is always 0)

Refer to “[Questionable Status Register Sets](#)” on page 376 for the `STATUS:QUESTIONABLE` commands.

The condition bits are set and cleared under the following conditions:

**Table 1-6** Bit change conditions for the Questionable Status register set

Bit	Definition	EVENTs causing bit changes
3	POWer summary	<p>This is the summary bit for the Questionable POWer register</p> <p><b>SET:</b>            Error -230, "Data corrupt or stale"            Error -231, "Data questionable;Input Overload"            Error -231, "Data questionable;Input Overload ChA"            Error -231, "Data questionable;PLEASE ZERO"            Error -231, "Data questionable;PLEASE ZERO ChA"            Error -231, "Data questionable;Lower window log error"            Error -231, "Data questionable;Upper window log error"</p> <p><b>CLEARED:</b> When no errors are detected by the N432A during a measurement covering the causes given for it to set</p>
8	CALibration summary	<p>This is the summary bit for the Questionable CALibration register</p> <p><b>SET:</b>            This may be caused by            CALibration[1]:ZERO:AUTO ONCE            Error -231, "Data questionable; ZERO ERROR"            Error -231, "Data questionable; ZERO ERROR ChA"</p> <p><b>CLEARED:</b> When any of the events listed above succeed and no errors are placed on the error queue</p>
9	Power-on self-test	<p><b>SET:</b> When the power-on self-test fails</p> <p><b>CLEARED:</b> When the power-on self-test passes</p>

### Operation Status register

The Operation Status group monitors conditions in the N432A measurement process.

The following bits in this register set are used by the N432A:

**Table 1-7** Bit definitions – Operation Status

Bit	Weight/Decimal value	Definition
0	1	CALibration summary
1	2	Self-test started
2 to 3	–	Not used
4	16	MEASure summary
5	32	TRIGger summary
6 to 10	–	Not used
11	2048	Lower limit fail summary
12	4096	Upper limit fail summary
13 to 15	–	Not used (bit 15 is always 0)

Refer to “[Operation Status Registers Sets](#)” on page 368 for the `STATUS:OPERation` commands.

The condition bits are set and cleared under the following conditions:

**Table 1-8** Bit change conditions for the Operation Status register set

Bit	Definition	EVENTs causing bit changes
0	CALibrating	This is the summary bit for the Operation CALibrating register <b>SET:</b> At the beginning of zeroing (CALibration:ZERO:AUTO ONCE) <b>CLEARED:</b> At the end of zeroing
4	MEASuring	This is the summary bit for the Operation MEASuring register <b>SET:</b> When the N432A is taking a measurement <b>CLEARED:</b> When the measurement has completed
5	Waiting for TRIGger	This is the summary bit for the Operation TRIGger register <b>SET:</b> When the N432A enters the “wait-for-trigger” state <b>CLEARED:</b> When the N432A enters the “idle” state
11	Lower Limit Fail	This is the summary bit for the Lower Limit Fail register <b>SET:</b> If a measurement is made and a window lower limit test fails <b>CLEARED:</b> If a measurement is made and the lower limit test is not enabled or the test is enabled and passes
12	Upper Limit Fail	This is the summary bit for the Upper Limit Fail register <b>SET:</b> If a measurement is made and a window upper limit test fails <b>CLEARED:</b> If a measurement is made and the upper limit test is not enabled or the test is enabled and passes

### Device Status register

The Device Status register contains bits which provide device-dependent information.

The following bits in this register are used by the N432A:

**Table 1-9** Bit definitions – Device Status register

Bit	Weight/Decimal value	Definition
0	1	Not used
1	2	Channel A sensor connected
2 to 11	–	Not used
12	4096	Over temperature
13	8192	Fan failed
14	16384	Front panel keypress
15	–	Not used (always 0)

Refer to “[Device Status Register Set](#)” on page 366 for the `STATUS:DEVICE` commands.

The condition bits are set and cleared under the following conditions:

**Table 1-10** Bit change conditions for the Device Status register

Bit	Definition	EVENTs causing bit changes
1	Channel A sensor connected	<b>SET:</b> When a sensor is connected to the Channel A input <b>CLEARED:</b> When no sensor is connected to the Channel A input
12	Over temperature	This is an event, and DOES NOT set the condition register. The bit is set in the event register which is cleared when read. Note that the transition registers are of no use for this bit.
13	Fan failed	This is an event, and DOES NOT set the condition register. The bit is set in the event register which is cleared when read. Note that the transition registers are of no use for this bit.
14	Front panel keypress	This is an event, and DOES NOT set the condition register. The bit is set in the event register which is cleared when read. Note that the transition registers are of no use for this bit.

## Using the Operation Complete commands

The `*OPC?` query and `*OPC` command allow you to maintain synchronization between the PC and the N432A. The `*OPC?` query places an ASCII character 1 into the N432A output queue when all pending N432A commands have completed. If your program reads this response before continuing program execution, you can ensure synchronization between one or more instruments and the PC.

The `*OPC` command sets bit 0 (Operation Complete) in the Standard Event Status register when all pending N432A operations have completed. By enabling this bit to be reflected in the Status register, you can ensure synchronization using the GPIB serial poll.

### NOTE

For LAN and USB, use the `*STB?` command. Refer to “Using `*STB?` to read the Status Byte” on page 64.

### Procedure

- 1 Send a device clear message to clear the N432A output buffer.
- 2 Clear the event registers with the \*CLS (clear status) command.
- 3 Enable Operation Complete using the \*ESE 1 command (Standard Event register).
- 4 Send the \*OPC? query and enter the result to assure synchronization.
- 5 Send your programming command string, and place the \*OPC command as the last command.
- 6 Send the \*STB? (status byte) query to poll the register. This command does not clear the Status Byte summary register.
- 7 In the GPIB mode only, you can use a serial poll to check when bit 5 (Standard Event) is set in the Status Byte summary register. You can also configure the N432A for an SRQ interrupt by sending \*SRE 32 (Status Byte enable register, bit 5).



## Saving and Recalling N432A Configurations

To reduce repeated programming, up to ten N432A configurations can be stored in the N432A non-volatile memory. The error list, remote addresses, sensor calibration table data, and zeroing information are not stored.

### How to save and recall a configuration

The N432A configurations can be saved and recalled using the following commands:

```
*SAV <NRf>
```

```
*RCL <NRf>
```

The range of values for <NRf> in the above commands is 1 to 10.

For details on these commands, refer to “\*SAV <NRf>” on page 89 and “\*RCL <NRf>” on page 87.

#### Example program

```
10 ASSIGN @POWER TO 713
20 !Configure the power meter
30 OUTPUT @POWER;"UNIT:POW W"
40 OUTPUT @POWER;"SENS:CORR:GAIN2 10"
50 OUTPUT @POWER;"SENS:CORR:GAIN2:STAT ON"
60 !Save the configuration
70 OUTPUT @POWER;"*SAV 5"
80 PRINT "Configuration Saved"
90 !Now reset the power meter
100 OUTPUT @POWER;"*RST"
110 !Recall the configuration
120 OUTPUT @POWER;"*RCL 5"
130 PRINT "Configuration Recalled"
140 PRINT "Save and Recall complete"
150 END
```

## Using Device Clear to Halt Measurements

Device Clear is an IEEE-488 low-level bus message which can be used to halt measurements in progress. The status registers, error queue, and all configuration states are left unchanged when a Device Clear message is received. Device Clear performs the following actions:

- All measurements in progress are aborted.
- The N432A returns to the trigger “idle state”.
- The N432A input and output buffers are cleared.
- The N432A is prepared to accept a new command string.

### NOTE

For interfaces that do not support a low-level Device Clear, use the [ABORt\[1\]](#) command.

---

## Summary of Commands

For details of each SCPI command available to program the N432A, refer to later chapters for more details on each command.

For commands that require you to specify a window, the upper window is represented by a 1 and the lower window by a 2. If you omit the window number, the upper window is assumed.



## 2 IEEE-488 Command Reference

SCPI Compliance Information 76

\*CLS 77

\*DDT <arbitrary block program data> | <string program data> 78

\*ESE <NRf> 80

\*ESR? 82

\*IDN? 83

\*LRN? 84

\*OPC 85

\*OPT? 86

\*RCL <NRf> 87

\*RST 88

\*SAV <NRf> 89

\*SRE <NRf> 90

\*STB? 92

\*TST? 94

\*WAI 95

GPIO Universal Commands 96

This chapter contains information on the IEEE-488 common commands supported by the N432A.



## SCPI Compliance Information

This chapter describes the IEEE-488 common commands supported by the N432A. It also describes the GPIB universal command statements which form the nucleus of GPIB programming; they are understood by all instruments in the network. When combined with programming language codes, they provide all management and data communication instructions for the system.

The IEEE-488 common command descriptions are listed below in the alphabetical order:

*CLS	Clear Status	<a href="#">page 77</a>
*DDT and *DDT?	Define Device Trigger	<a href="#">page 78</a>
*ESE and *ESE?	Event Status Enable	<a href="#">page 80</a>
*ESR?	Event Status Register	<a href="#">page 82</a>
*IDN?	Identify	<a href="#">page 83</a>
*LRN?	Learn	<a href="#">page 84</a>
*OPC and *OPC?	Operation Complete	<a href="#">page 85</a>
*OPT?	Options	<a href="#">page 86</a>
*RCL	Recall	<a href="#">page 87</a>
*RST	Reset	<a href="#">page 88</a>
*SAV	Save	<a href="#">page 89</a>
*SRE and *SRE?	Service Request Enable	<a href="#">page 90</a>
*STB?	Status Byte	<a href="#">page 92</a>
*TST?	Self-test	<a href="#">page 94</a>
*WAI	Wait	<a href="#">page 95</a>

## \*CLS

The \*CLS (Clear Status) command clears the status data structures. The SCPI registers (Questionable Status, Operation Status, and all the other SCPI registers), the Standard Event Status register, the Status Byte, and the Error/Event Queue are all cleared.

### Syntax

\*CLS →

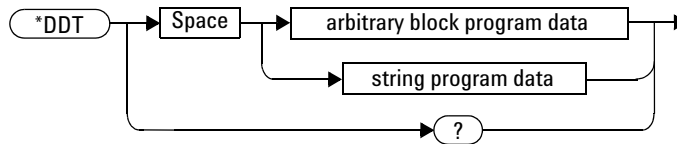
### Example

\*CLS            *This command clears the status data structures of the N432A.*

## \*DDT <arbitrary block program data> | <string program data>

The \*DDT (Define Device Trigger) command determines the N432A response to a GET (Group Execute Trigger) message. This command effectively turns GET into a query, with the measured power being returned.

### Syntax



### Parameters

Type	Description	Range of values
arbitrary block program data	The command which is executed on a GET	#nN<action> <sup>[1], [2]</sup>
string program data		"<action>" <sup>[1]</sup>

[1] The <action> field of the parameter may contain:

FETC?  
TRIG

[2] The first digit after # indicates the number of following digits. The following digits indicate the length of the data.

Examples of <arbitrary block program data> parameters are:

#15FETC? and #206FETCh?

Examples of <string program data> parameters are:

"FETCh?" and "TRIG;FETC"

## Reset condition

On reset, the <action> field of \*DDT is set to \*TRG.

## Query

\*DDT?

This query returns the action which is performed on receipt of a GET. This is returned as a <definite-length arbitrary block response data> value in the form of #nN<action>.

## Error message

If an invalid parameter is received, error -224, "Illegal parameter value" occurs.

## \*ESE <Nrf>

The \*ESE (Event Status Enable) command sets the bits in the Standard Event Enable register. The selected bits are then reported to bit 5 of the Status Byte register. The <Nrf> parameter is used to specify which bits will be enabled. The specified decimal value corresponds to the binary-weighted sum of the bits you wish to enable in the register. For example, to enable bit 2 (decimal value = 4), bit 3 (decimal value = 8), and bit 7 (decimal value = 128), the corresponding decimal value would be 140 (4 + 8 + 128).

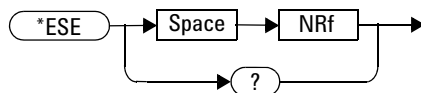
The following table lists the bit definitions for the Standard Event register:

**Table 2-1** Bit definitions: Standard Event register

Bit	Weight/Decimal value	Definition
0	1	Operation complete
1	2	Not used
2	4	Query error
3	8	Device-dependent error
4	16	Execution error
5	32	Command error
6	64	User request
7	128	Power on

Refer to “Standard Event register” on page 64 for more details.

## Syntax





## Parameter

Type	Description/Default	Range of values
NRf	A decimal value which corresponds to the binary-weighted sum of the bits to enable in the Standard Event register	0 to 255

## Example

\*ESE 16      *This command enables bit 4 (decimal value = 16) in the enable register.*

## Query

\*ESE?

This query reads the Standard Event Enable register and returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register. The return format is <NR1> in the range of 0 to 255.

## Query example

\*ESE?      *Returns the bits currently set in the enable register.*

## \*ESR?

The \*ESR? query reads the event register of the Standard Event register group and returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register. The return format is <NR1> in the range of 0 to 255. The bits are cleared by executing this query or \*CLS.

The following table lists the bit definitions for the Standard Event register:

**Table 2-2** Bit definitions: Standard Event register

Bit	Weight/Decimal value	Definition
0	1	Operation complete
1	2	Not used
2	4	Query error
3	8	Device-dependent error
4	16	Execution error
5	32	Command error
6	64	User request
7	128	Power on

Refer to “[Standard Event register](#)” on page 64 for more details.

## Syntax



## Example

\*ESR?      *Returns the bits currently set in the event register.*

## \*IDN?

The \*IDN? query reads the N432A identification string and returns the string in the following format:

```
Agilent Technologies,N432A,<serial number>,A1.XX.YY
```

where,

- <serial number> uniquely identifies each N432A, and
- A1.XX.YY represents the firmware revision with XX and YY representing the major and minor revisions respectively.

## Syntax



## Example

\*IDN?      *Returns the N432A identification string.*

## \*LRN?

The \*LRN? query returns the N432A state.

### Syntax



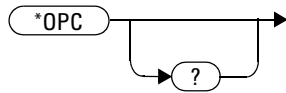
### Example

\*LRN?      *Returns the N432A state.*

## \*OPC

The \*OPC (Operation Complete) command sets the “Operation Complete” bit (bit 0) in the Standard Event register when all pending operations have completed. This command is used to synchronize your application with the N432A.

### Syntax



### Example

\*OPC            *This command sets the "Operation Complete" bit.*

### Query

\*OPC?

This query sends 1 to the output buffer when all pending operations have completed.

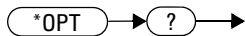
### Query example

\*OPC?            *Waits until the completion of the current command and then sends 1 to the output buffer.*

## \*OPT?

The \*OPT? query reports the options installed in the N432A and returns an empty string " ".

### Syntax



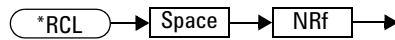
### Query example

\*OPT?      *Queries the options installed in the N432A.*

## \*RCL <NRf>

The \*RCL (ReCaLL) command restores the state of the N432A from the specified save/recall register. This command is only effective when an instrument setup has been stored previously in the specified register.

### Syntax



### Parameter

Type	Description/Default	Range of values
NRf	The register number to recall	1 to 10

### Example

\*RCL 5      *This command restores the N432A state from register 5.*


### Error message

If the register does not contain a saved state, error -224, “Illegal parameter value” occurs.

## \*RST

The \*RST (ReSeT) command places the N432A in a known state.

### Syntax



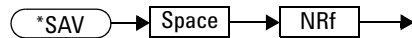
### Example

\*RST      *This command resets the N432A to its default settings.*



**\*SAV <NRf>**

The \*SAV (SAVe) command stores the current state of the N432A in the specified save/recall register.

**Syntax****Parameter**

Item	Description/Default	Range of values
NRf	The register number for storing the N432A state	1 to 10

**Example**

\*SAV 3      *This command saves the current N432A state to register 3.*

**\*SRE <Nrf>**

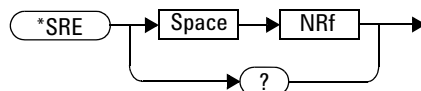
The \*SRE command enables the bits in the Status Byte enable register. The selected enabled bits are summarized in the “Master Status Summary” (MSS) bit (bit 6) of the Status Byte register. If any of the selected bit condition changes from 0 to 1, a Service Request is generated. The <Nrf> parameter is used to specify which bits to enable. The specified decimal value corresponds to the binary-weighted sum of the bits you wish to enable in the register. For example, to enable bit 2 (decimal value = 4) and bit 5 (decimal value = 32), the corresponding decimal value would be 36 (4 + 32).

The following table lists the bit definitions of the Status Byte register:

**Table 2-3** Bit definitions: Status Byte register

Bit	Weight/Decimal value	Definition
0	1	Not used
1	2	Device Status register summary
2	4	Error/Event queue
3	8	QUESTionable Status summary
4	16	Message available
5	32	Event Status Byte summary
6	64	Master Status summary (Request for service)
7	128	OPERation Status summary

Refer to “[Status Byte summary register](#)” on page 63 for more details.

**Syntax**

## Parameter

Type	Description/Default	Range of values
NRf	A decimal value which corresponds to the binary-weighted sum of the bits to enable in the Status Byte register	0 to 255

## Example

\*SRE 16      *This command enables bit 4 (decimal value = 16) in the enable register.*

## Query

\*SRE?

This query reads the Status Byte enable register and returns a decimal value that corresponds to the binary-weighted sum of all bits set in the register. The return format is <NR1> in the range of 0 to 255.

## Query example

\*SRE?      *Returns the bits currently set in the enable register.*

**\*STB?**

The \*STB? query reads the condition register of the Status Byte register and returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register. The return format is <NR1> in the range of 0 to 255. This query is similar to a Serial Poll but it is processed like any other instrument command. This is a read-only register and the bits are not cleared when you read the register.

**NOTE**

This query returns the same results as a Serial Poll but the “Master Status Summary” (MSS) bit (bit 6) is not cleared if a Serial Poll has occurred.

The following table lists the bit definitions of the Status Byte register:

**Table 2-4** Bit definitions: Status Byte register

Bit	Weight/Decimal value	Definition
0	1	Not used
1	2	Device Status register summary
2	4	Error/Event queue
3	8	QUESTionable Status summary
4	16	Message available
5	32	Event Status Byte summary
6	64	Master Status summary (Request for service)
7	128	OPERation Status summary

Refer to “[Status Byte summary register](#)” on page 63 for more details.

**Syntax****Example**

\*STB?      *Returns the bits currently set in the condition register.*

## \*TRG

The \*TRG (TRiGger) command triggers channel A when it is in the wait-for-trigger state, and only if the bus trigger source is currently selected. The \*TRG command has the same effect as a Group Execute Trigger (GET).

Using the \*DDT command may change the function of the \*TRG command.

### Syntax

\*TRG →

### Example

\*TRG            *Triggers channel A that is in the wait-for-trigger state.*

### Error message

If TRIGger:SOURce is not set to BUS, or the N432A is not in the wait-for-trigger state, error -211, “Trigger ignored” occurs.

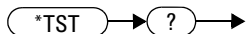
## \*TST?

The \*TST? (TeST) query initiates an internal self-test of the N432A and returns a pass or fail indication where:

- 0 is returned if all tests pass, or
- 1 is returned if one or more tests fail.

The self-test takes approximately 60 seconds to complete.

## Syntax



## Example

\*TST?      *Performs a self-test and returns a pass or fail indication.*

## \*WAI

The \*WAI (WAIt) command causes the N432A to wait until either:

- all pending operations have completed
- the device clear command is received
- power is cycled

before executing any subsequent commands or queries.

## Syntax

 \*WAI →

## Example

\*WAI

*Causes the N432A to wait before executing any other command.*

## GPIB Universal Commands

### DCL

The DCL (Device CLear) command causes all GPIB instruments to assume a cleared condition. The definition of device clear is unique for each instrument. For the N432A:

- All pending operations are halted, that are, \*OPC? and \*WAI.
- The parser (software that interprets the programming codes) is reset and now expects to receive the first character of a programming code.
- The output buffer is cleared.

### GET

The GET (Group Execute Trigger) command triggers all channels that are in the wait-for-trigger state.

Using the \*DDT command may change the function of the GET command.

#### Error message

If TRIGger:SOURce is not set to BUS, or the N432A is not in the wait-for-trigger state, error -211, “Trigger ignored” occurs.

### GTL

The GTL (Go To Local) command is the complement to remote. It causes the N432A to return to local control with a fully-enabled front panel. When reverting to local mode, the N432A triggering is set to free run.



## LLO

The LLO (Local Lock Out) command can be used to disable the front panel local key. With this key disabled, only the controller (or a hard reset by the line power switch) can restore local control.

## PPC

When addressed to listen, the PPC (Parallel Poll Configure) command causes the N432A to be configured according to the parallel poll enable secondary command which should follow this command.

## PPD

Sending the PPC command followed by the PPD (Parallel Poll Disable) command disables the N432A from responding to a parallel poll. This is effectively a selective disable.

**Table 2-5** Bit definitions: PPD

Bit	Weight/Decimal value	Definition
0	1	Always 0
1	2	Always 0
2	4	Always 0
3	8	Always 0
4	16	Always 1
5	32	Always 1
6	64	Always 1
7	128	Always 0

## PPE

Once the N432A has received a PPC command, the PPE (Parallel Poll Enable) secondary command configures the N432A to respond to a parallel poll on a particular data line with a particular level.

**Table 2-6** Bit definitions: PPE

Bit	Weight/Decimal value	Definition
0	1	Bit positions for response: 000 (bit 0), 001 (bit 1), 010 (bit 2), 011 (bit 3), 100 (bit 4), 101 (bit 5), 110 (bit 6), 111 (bit 7)
1	2	
2	4	
3	8	Sense bit 0 – response bit is cleared during a parallel poll if requesting service 1 – response bit is set during a parallel poll if requesting service
4	16	Always 0
5	32	Always 1
6	64	Always 1
7	128	Always 0

## PPU

The PPU (Parallel Poll Unconfigure) command disables the N432A from responding to a parallel poll. This is effectively a universal disable.

## SDC

The SDC (Selected Device Clear) command causes instruments using GPIB in the listen state, to assume a cleared condition. The definition of a selected device clear is unique for each instrument. For the N432A:

- All pending operations are halted, that are, \*OPC? and \*WAI.
- The parser (software that interprets the programming codes) is reset and now expects to receive the first character of a programming code.
- The output buffer is cleared.

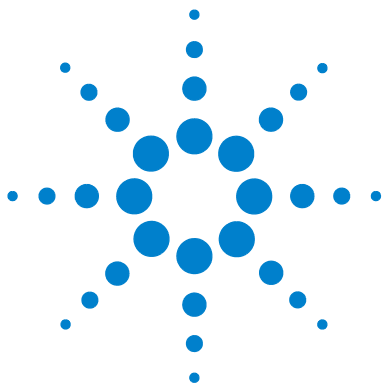
## SPD

The SPD (Serial Poll Disable) command terminates the serial poll mode for the N432A and returns it to its normal talker state where device-dependent data is returned rather than the status byte.

## SPE

The SPE (Serial Poll Enable) command establishes the serial poll mode for the N432A. When the N432A is addressed to talk, a single 8-bit status byte is returned.





## 3 MEASurement Commands

MEASurement Commands	103
CONFigure[1] 2 3 4? 108	
CONFigure [1] 2 3 4 Commands	110
CONFigure[1] 2 3 4[:SCALar][:POWer:AC] [<expected_value>,<resolution>,<source list>]]	111
CONFigure[1] 2 3 4[:SCALar][:POWer:AC]:RELative [<expected_value>,<resolution>,<source list>]]	113
CONFigure[1] 2 3 4[:SCALar][:POWer:AC]:DIFFerence [<expected_value>,<resolution>,<source list>]]	115
CONFigure[1] 2 3 4[:SCALar][:POWer:AC]:DIFFerence: RELative [<expected_value>,<resolution>,<source list>]]	117
CONFigure[1] 2 3 4[:SCALar][:POWer:AC]:RATio [<expected_value>,<resolution>,<source list>]]	119
CONFigure[1] 2 3 4[:SCALar][:POWer:AC]:RATio: RELative [<expected_value>,<resolution>,<source list>]]	121
FETCh[1] 2 3 4 Queries	123
FETCh[1] 2 3 4[:SCALar][:POWer:AC]? [<expected_value>,<resolution>,<source list>]]	124
FETCh[1] 2 3 4[:SCALar][:POWer:AC]:RELative? [<expected_value>,<resolution>,<source list>]]	126
FETCh[1] 2 3 4[:SCALar][:POWer:AC]:DIFFerence? [<expected_value>,<resolution>,<source list>]]	128
FETCh[1] 2 3 4[:SCALar][:POWer:AC]:DIFFerence: RELative? [<expected_value>,<resolution>,<source list>]]	130
FETCh[1] 2 3 4[:SCALar][:POWer:AC]:RATio? [<expected_value>,<resolution>,<source list>]]	132
FETCh[1] 2 3 4[:SCALar][:POWer:AC]:RATio:RELative? [<expected_value>,<resolution>,<source list>]]	134
FETCh[1] 2 3 4[:SCALar]:V0?	136
FETCh[1] 2 3 4[:SCALar]:V1?	137



FETCh[1] 2 3 4[:SCALar]:VCMP0?	138
FETCh[1] 2 3 4[:SCALar]:VCMP1?	139
FETCh[1] 2 3 4[:SCALar]:VRF0?	140
FETCh[1] 2 3 4[:SCALar]:VRF1?	141
FETCh[1] 2 3 4[:SCALar]:PV1?	142
READ[1] 2 3 4 Queries	143
READ[1] 2 3 4[:SCALar][:POWer:AC]? [<expected_value>[,<resolution>[,<source list>]]]	144
READ[1] 2 3 4[:SCALar][:POWer:AC]:RELative? [<expected_value>[,<resolution>[,<source list>]]]	146
READ[1] 2 3 4[:SCALar][:POWer:AC]:DIFFerence? [<expected_value>[,<resolution>[,<source list>]]]	148
READ[1] 2 3 4[:SCALar][:POWer:AC]:DIFFerence: RELative? [<expected_value>[,<resolution>[,<source list>]]]	150
READ[1] 2 3 4[:SCALar][:POWer:AC]:RATio? [<expected_value>[,<resolution>[,<source list>]]]	152
READ[1] 2 3 4[:SCALar][:POWer:AC]:RATio:RELative? [<expected_value>[,<resolution>[,<source list>]]]	154
MEASure[1] 2 3 4 Queries	156
MEASure[1] 2 3 4[:SCALar][:POWer:AC]? [<expected_value>[,<resolution>[,<source list>]]]	157
MEASure[1] 2 3 4[:SCALar][:POWer:AC]:RELative? [<expected_value>[,<resolution>[,<source list>]]]	159
MEASure[1] 2 3 4[:SCALar][:POWer:AC]:DIFFerence? [<expected_value>[,<resolution>[,<source list>]]]	161
MEASure[1] 2 3 4[:SCALar][:POWer:AC]:DIFFerence: RELative? [<expected_value>[,<resolution>[,<source list>]]]	163
MEASure[1] 2 3 4[:SCALar][:POWer:AC]:RATio? [<expected_value>[,<resolution>[,<source list>]]]	165
MEASure[1] 2 3 4[:SCALar][:POWer:AC]:RATio:RELative? [<expected_value>[,<resolution>[,<source list>]]]	167

This chapter explains how to use high-level measurement commands to acquire data.

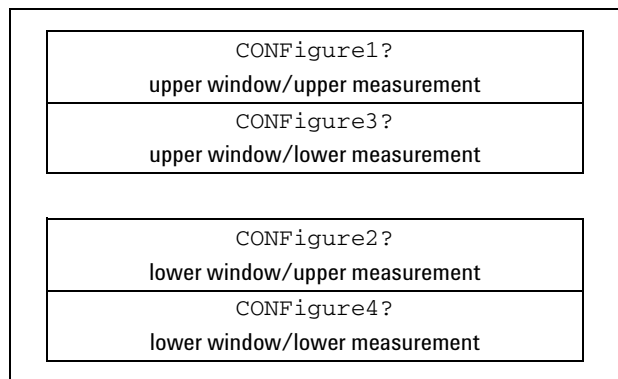
## MEASurement Commands

Measurement commands are high-level commands used to acquire data. They enable you to trade interchangeability against fine control of the measurement process.

Measurement command	Description
MEASure?	Provides the simplest way to program the N432A for measurements. MEASure? is a compound command which is equivalent to a CONFigure followed by a READ?. It does not enable much flexibility or control over measurement settings.
CONFigure	Used to change the N432A configuration values. CONFigure must then be followed by another command which takes the measurement, for example, INITiate <sup>[1]</sup> followed by FETCh?.
READ?	Takes a measurement using parameters previously set up using either CONFigure or lower-level commands. READ? is equivalent to an INITiate (which performs the data acquisition) and a FETCh?.
FETCh?	Retrieves measurements taken by INITiate.

[1] INITiate is described in [Chapter 16](#), “TRIGger Subsystem,” on page 419.

CONFigure, FETCh?, READ?, and MEASure? all have a numeric suffix which refers to a specific window/measurement. The figure below shows an example of the configuration returned result windows.



**Figure 3-1** Measurement display CONFigure block window

## Optional parameters

CONFigure, FETCh? (excluding FETCh? for voltage measurement results), READ?, and MEASure? have the following three optional parameters:

- An expected power value
- A resolution
- A source list

### Expected power value

The <expected\_value> parameter is only required for non-thermistor sensors. This parameter is provided for backward compatibility purpose and has no effect on thermistor sensors.

### Resolution

The <resolution> parameter sets the resolution of the specified window. This parameter does not affect the resolution of the returned data.

### Source list

The <source list> parameter is used to define the measurement channel, which is channel A.



## Summary of CONFigure, FETCh?, READ?, and MEASure?

The following commands are described in this chapter:

Keyword	Parameter form	Notes	Page
CONFigure[1]   2   3   4?		[query only]	<a href="#">page 108</a>
CONFigure[1]   2   3   4 [:SCALar]			
[:POWER:AC]	[<expected_value> [,<resolution>[,<source list>]]]	[no query]	<a href="#">page 111</a>
:RELative	[<expected_value> [,<resolution>[,<source list>]]]	[no query]	<a href="#">page 113</a>
:DIFFerence	[<expected_value> [,<resolution>[,<source list>]]]	[no query]	<a href="#">page 115</a>
:RELative	[<expected_value> [,<resolution>[,<source list>]]]	[no query]	<a href="#">page 117</a>
:RATio	[<expected_value> [,<resolution>[,<source list>]]]	[no query]	<a href="#">page 119</a>
:RELative	[<expected_value> [,<resolution>[,<source list>]]]	[no query]	<a href="#">page 121</a>
<hr/>			
FETCh[1]   2   3   4 [:SCALar]			
[:POWER:AC]?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	<a href="#">page 124</a>
:RELative?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	<a href="#">page 126</a>
:DIFFerence?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	<a href="#">page 128</a>
:RELative?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	<a href="#">page 130</a>
:RATio?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	<a href="#">page 132</a>

### 3 MEASurement Commands

Keyword	Parameter form	Notes	Page
:RELative?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	<a href="#">page 134</a>
:V0?		[query only]	<a href="#">page 136</a>
:V1?		[query only]	<a href="#">page 137</a>
:VCMP0?		[query only]	<a href="#">page 138</a>
:VCMP1?		[query only]	<a href="#">page 139</a>
:VRF0?		[query only]	<a href="#">page 140</a>
:VRF1?		[query only]	<a href="#">page 141</a>
:PV1?		[query only]	<a href="#">page 142</a>
<hr/>			
READ[1]   2   3   4			
[:SCALar]			
[:POWER:AC]?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	<a href="#">page 144</a>
:RELative?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	<a href="#">page 146</a>
:DIFFerence?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	<a href="#">page 148</a>
:RELative?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	<a href="#">page 150</a>
:RATio?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	<a href="#">page 152</a>
:RELative?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	<a href="#">page 154</a>
<hr/>			
MEASure[1]   2   3   4			
[:SCALar]			
[:POWER:AC]?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	<a href="#">page 157</a>
:RELative?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	<a href="#">page 159</a>

Keyword	Parameter form	Notes	Page
:DIFFerence?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	<a href="#">page 161</a>
:RELative?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	<a href="#">page 163</a>
:RATio?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	<a href="#">page 165</a>
:RELative?	[<expected_value> [,<resolution>[,<source list>]]]	[query only]	<a href="#">page 167</a>

## CONFigure[1]|2|3|4?

This query returns the present configuration of the specified window/measurement as a quoted string in the following format:

```
"<function> <expected_value>,<resolution>,<source list>"
```

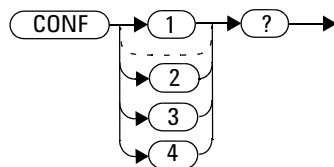
The string returned depends on the settings of the CALCulate:MATH and CALCulate:RELative:STATE commands.

CALCulate:MATH	CALCulate:RELative:STATE	Function	<source list>
(SENSe1)	OFF	[ :POW:AC ]	(@1)
(SENSe1)	ON	[ :POW:AC ]:REL	(@1)
(SENSe1 - SENSe1)	OFF	[ :POW:AC ]:DIFF	(@1) , (@1)
(SENSe1 - SENSe1)	ON	[ :POW:AC ]:DIFF:REL	(@1) , (@1)
(SENSe1/SENSe1)	OFF	[ :POW:AC ]:RAT	(@1) , (@1)
(SENSe1/SENSe1)	ON	[ :POW:AC ]:RAT:REL	(@1) , (@1)

The <expected\_value> returned is the expected power value sent by the last CONFigure command or +20 dBm by default.

The <resolution> returned is the same as the value returned by DISPlay:WINDow:RESolution?. The return format is <NR1> in the range of 1 through 4.

### Syntax



## Example

CONF2?      *Queries the current configuration of the lower window/upper measurement.*

## Reset condition

On reset:

The command function is set to :POWER:AC.

The expected power level is set to +20 dBm.

The resolution is set to 3.

The source list is set to Channel A.

## CONFigure [1] | 2 | 3 | 4 Commands

The CONFigure command is used on the specified window/measurement to set:

- the expected power level being measured.
- the resolution of the window/measurement.
- the channel on which the measurement is to be made.

### NOTE

The expected power level parameter is only required for non-thermistor sensors. This parameter is provided for backward compatibility purpose and has no effect on thermistor sensors.

The CONFigure command does not make the power measurement after setting the configuration. Use READ?, or alternatively use INITiate followed by a FETCh? to make the measurement.

The CONFigure command also applies the following default settings in the specified window.

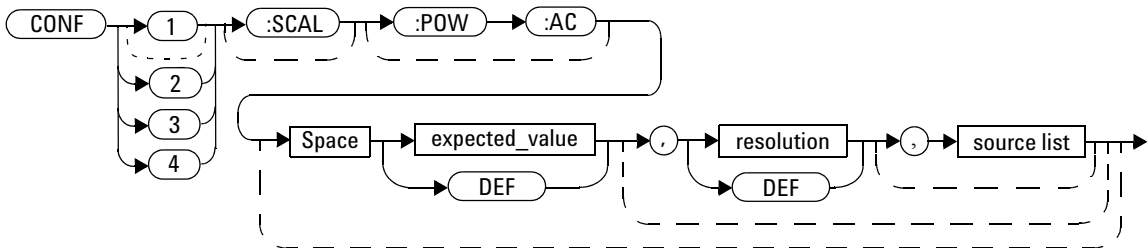
Default settings	Description
INITiate:CONTInuous OFF	Sets the N432A to make one trigger cycle when INITiate is sent
TRIGger:SOURce IMMEDIATE	Sets the N432A to make the measurement immediately once a trigger is received
TRIGger:DELAy:AUTO ON	Enables automatic delay before making the measurement

# CONFigure[1] | 2 | 3 | 4[:SCALar][:POWer:AC] [<expected\_value>[,<resolution>[,<source list>]]]

This command is used on the specified window/measurement to set:

- the expected power level of the measurement.
- the resolution of the window/measurement.
- the channel on which the measurement will be made.

## Syntax



## Parameters

Refer to “Optional parameters” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value	A numeric value for the expected power level. The units of measurement are dBm and W. The default unit is defined by UNIT:POWer.	Sensor dependent DEF <sup>[1]</sup>
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>

### 3 MEASurement Commands

Item	Description/Default	Range of values
source list	The channel which the command is implemented on. If unspecified, the current window setup is used.	(@1)

[1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.

[2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

## Example

```
CONF DEF, 2, (@1)
```

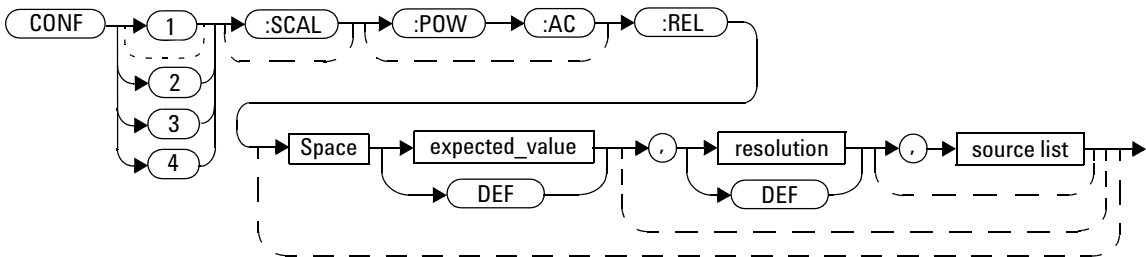
*This command configures the upper window/upper measurement to measure power using the current expected power level and a resolution setting of 2.*



## CONFigure[1] | 2 | 3 | 4[:SCALar][:POWer:AC]:RELative [<expected\_value>[,<resolution>[,<source list>]]]

This command sets the measurement type, range, and resolution of the specified window/measurement. It sets the measurement type to single function measurement with relative mode on. The relative value used is that set by the CALCulate:RELative:MAGNitude:AUTO command.

### Syntax



### Parameters

Refer to “[Optional parameters](#)” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value	A numeric value for the expected power level. The units of measurement are dBm and W. The default unit is defined by UNIT:POWer.	Sensor dependent DEF <sup>[1]</sup>
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>
source list	The channel which the command is implemented on. If unspecified, the current window setup is used.	(@1)

- [1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.
- [2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

## Example

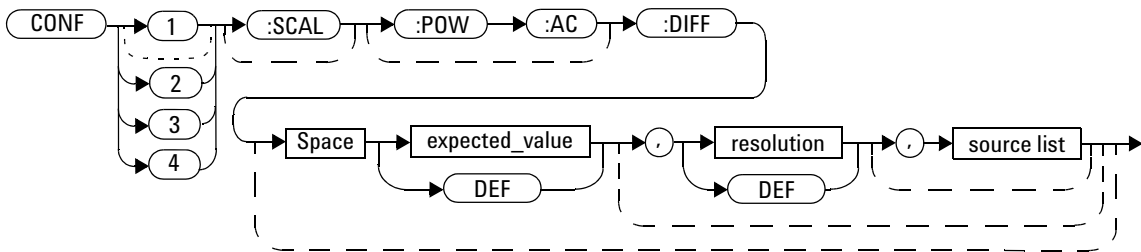
```
CONF2:REL -50DBM,3,(@1)
```

*This command configures the lower window/upper measurement to measure relative power using an expected power level of -50 dBm and a resolution setting of 3.*

## CONFigure[1] | 2 | 3 | 4[:SCALar][:POWer:AC]:DIFFerence [<expected\_value>[,<resolution>[,<source list>]]]

This command sets the measurement type and resolution of the specified window/measurement. It sets the measurement type to a difference measurement with relative mode off.

### Syntax



### Parameters

Refer to “[Optional parameters](#)” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	The N432A ignores the numeric value entered for this parameter. Any value entered is treated like DEF.	Sensor dependent DEF <sup>[1]</sup>
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>
source list	Specifies the channels for calculating the difference value. If unspecified and the current window setup is a difference measurement, then this setup is used, otherwise it defaults to A–A.	(@1),(@1)

- [1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.
- [2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

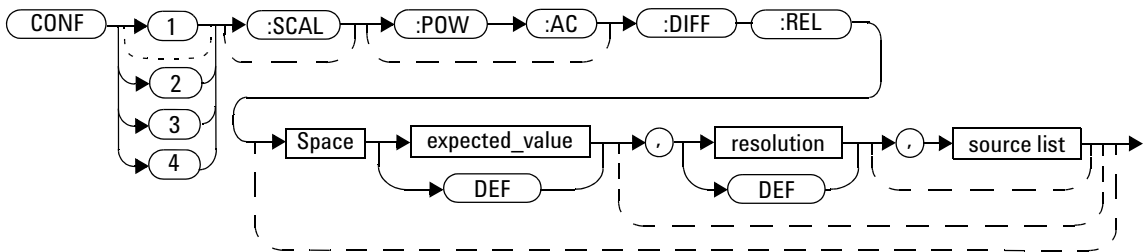
## Example

CONF3:DIFF DEF,1,(@1),(@1) *This command configures the upper window/lower measurement to perform a difference measurement of Channel A – Channel A with relative mode off, using the current expected power level and a resolution setting of 1.*

## CONFigure[1] | 2 | 3 | 4[:SCALar][:POWer:AC]:DIFFerence:RELative [<expected\_value>[,<resolution>[,<source list>]]]

This command sets the measurement type and resolution of the specified window/measurement. It sets the measurement type to a difference measurement with relative mode on. The relative value used is that set by the CALCulate:RELative:MAGNitude:AUTO command.

### Syntax



### Parameters

Refer to “[Optional parameters](#)” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	The N432A ignores the numeric value entered for this parameter. Any value entered is treated like DEF.	Sensor dependent DEF <sup>[1]</sup>
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>

Item	Description/Default	Range of values
source list	Specifies the channels for calculating the difference value. If unspecified and the current window setup is a difference measurement, then this setup is used, otherwise it defaults to A–A.	(@1),(@1)

- [1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.
- [2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

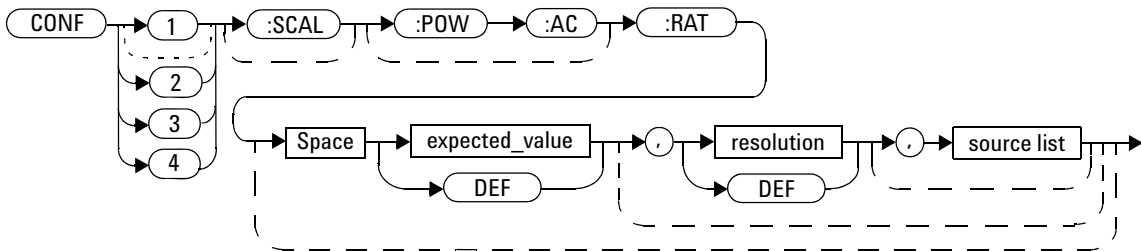
## Example

CONF:DIFF:REL DEF,3,(@1),(@1) *This command configures the upper window/upper measurement to perform a difference measurement of Channel A – Channel A with relative mode on, using the current expected power level and a resolution setting of 3.*

## CONFigure[1] | 2 | 3 | 4[:SCALar][:POWER:AC]:RATio [<expected\_value>[,<resolution>[,<source list>]]]

This command sets the measurement type and resolution of the specified window/measurement. It sets the measurement type to a ratio measurement with relative mode off.

### Syntax



### Parameters

Refer to “[Optional parameters](#)” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	The N432A ignores the numeric value entered for this parameter. Any value entered is treated like DEF.	Sensor dependent DEF <sup>[1]</sup>
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>
source list	Specifies the channels for calculating the ratio value. If unspecified and the current window setup is a ratio measurement, then this setup is used, otherwise it defaults to A/A.	(@1),(@1)

- [1] The mnemonic `DEF` means `DEFault`. This is not equivalent to the `DEFault` parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter `DEFault` is used as a placeholder. Specifying `DEF` leaves the parameter value unchanged.
- [2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

## Example

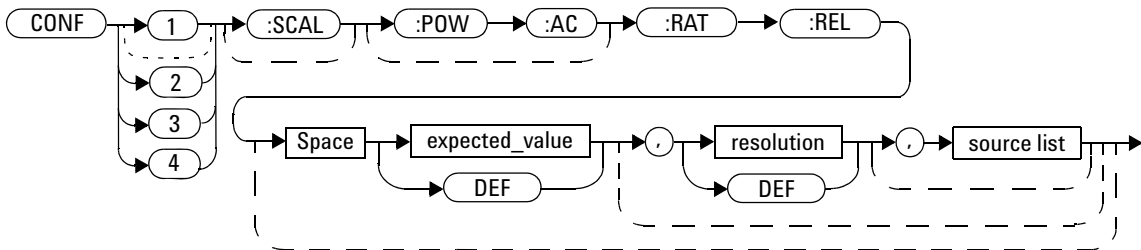
`CONF4:RAT DEF,4,(@1),(@1)` *This command configures the lower window/lower measurement to perform a ratio measurement of Channel A/Channel A with relative mode off, using the current expected power level and a resolution setting of 4.*



## CONFigure[1] | 2 | 3 | 4[:SCALar][:POWER:AC]:RATio: RELative [<expected\_value>[,<resolution>[,<source list>]]]

This command sets the measurement type and resolution of the specified window/measurement. It sets the measurement type to a ratio measurement with relative mode on. The relative value used is that set by the CALCulate:RELative:MAGNitude:AUTO command.

### Syntax



### Parameters

Refer to “[Optional parameters](#)” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	The N432A ignores the numeric value entered for this parameter. Any value entered is treated like DEF.	Sensor dependent DEF <sup>[1]</sup>
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>

Item	Description/Default	Range of values
source list	Specifies the channels for calculating the ratio value. If unspecified and the current window setup is a ratio measurement, then this setup is used, otherwise it defaults to A/A.	(@1),(@1)

[1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.

[2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

## Example

```
CONF:RAT:REL DEF,1,(@1),(@2)
```

*This command configures the upper window/upper measurement to perform a ratio measurement of Channel A/Channel A with relative mode on, using the current expected power level and a resolution setting of 1.*

## FETCh[1]|2|3|4 Queries

The FETCh? query sets the measurement type of the specified window/measurement to either a single, difference, or ratio measurement, with relative mode either off or on. It then recalculates the measurement and places the result on the bus. The result format is set by FORM[:READ][:DATA]. Refer to [Chapter 7, “FORMat Subsystem”](#) on page 233 for further information. The FETCh? query can also be used to retrieve voltage measurement results of  $V_0$ ,  $V_1$ ,  $V_{RF0}$ ,  $V_{RF1}$ ,  $V_{COMP0}$ , and  $V_{COMP1}$ .

The query returns a measurement result when it is valid. The measurement result is invalid under the following conditions:

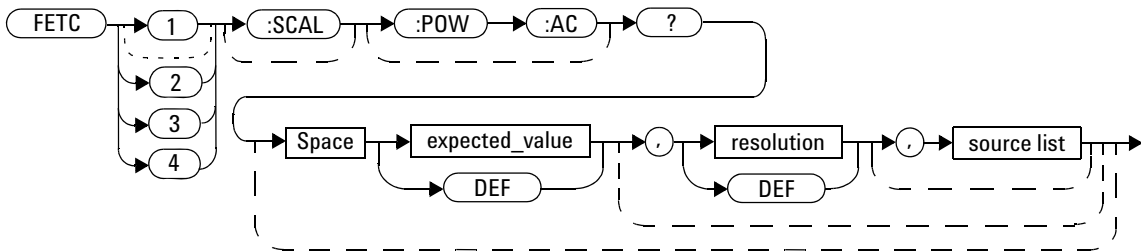
- When \*RST is executed.
- Whenever a measurement is initiated.
- When any SENSE parameter, such as frequency, is changed.

If data is invalid, the FETCh? query is not completed until all data becomes valid. The exceptions to this are, if the N432A is in the idle state and the data is invalid, or the N432A has been reconfigured as defined above and no new measurement has been initiated. In such cases, the FETCh? routine generates the error -230, “Data corrupt or stale” and no result is returned. A common cause for this error is receiving a FETCh? after an \*RST. If the expected value and resolution parameters are not the same as those that were used to collect the data, error -221, “Settings conflict” occurs.

## FETCh[1] | 2 | 3 | 4[:SCALar][:POWer:AC]? [<expected\_value>[,<resolution>[,<source list>]]]

This query sets the measurement type to single measurement with relative mode off for the specified window/measurement, recalculates the measurement, and places the result on the bus. The result is a power-based measurement and is expressed in the unit defined by UNIT[1] | 2 | 3 | 4:POWer.

### Syntax



### Parameters

Refer to “[Optional parameters](#)” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	This parameter can be set to DEF or a numeric value. If a value is entered, it should correspond to that set by CONFIGure otherwise an error occurs. The units of measurement are dBm and W. The default unit is defined by UNIT:POWer.	Sensor dependent DEF <sup>[1]</sup>
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used. If a value is entered, it should correspond to the current resolution setting otherwise an error occurs.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>

Item	Description/Default	Range of values
source list	The channel which the command is implemented on. If unspecified, the current window setup is used.	(@1)

- [1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.
- [2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

## Example

FETC2?      *Queries the lower window/upper measurement result.*

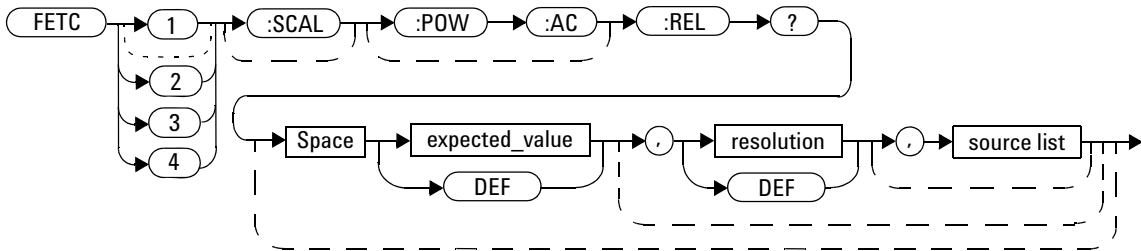
## Error messages

- If the last measurement is not valid, error -230, “Data corrupt or stale” occurs. A measurement is valid after it has been initiated. It becomes invalid when either a reset occurs or any measurement parameter, for example frequency, is changed.
- If the expected power level and resolution parameters are not the same as the current expected value and resolution setting on the specified window/measurement, error -221, “Settings conflict” occurs.

## FETCh[1] | 2 | 3 | 4[:SCALar][:POWer:AC]:RELative? [<expected\_value>[,<resolution>[,<source list>]]]

This query sets the measurement type to single measurement with relative mode on for the specified window/measurement, recalculates the measurement, and places the result on the bus. The result is a ratio-based measurement and is expressed in the unit defined by UNIT[1] | 2 | 3 | 4:POWer:RATio. The relative value used is that set by the CALCulate:RELative:MAGNitude:AUTO command.

### Syntax



### Parameters

Refer to “[Optional parameters](#)” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	This parameter can be set to DEF or a numeric value. If a value is entered, it should correspond to that set by CONFigure otherwise an error occurs. The units of measurement are dBm and W. The default unit is defined by UNIT:POWer.	Sensor dependent DEF <sup>[1]</sup>

Item	Description/Default	Range of values
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used. If a value is entered, it should correspond to the current resolution setting otherwise an error occurs.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>
source list	The channel which the command is implemented on. If unspecified, the current window setup is used.	(@1)

[1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.

[2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

## Example

FETC:REL? DEF, 2, (@1) *Queries the relative measurement of the upper window/upper measurement, using the current expected power level and a resolution setting of 2.*

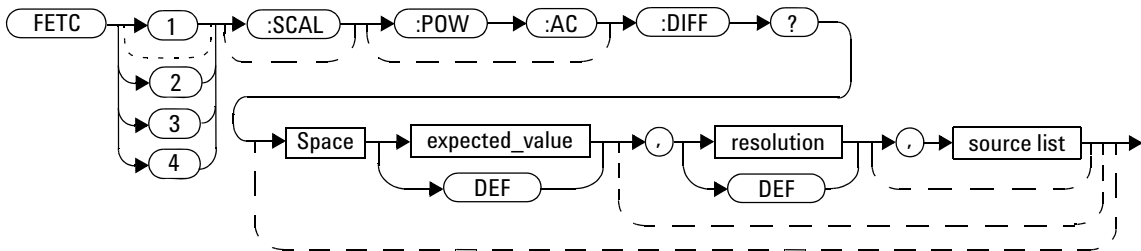
## Error messages

- If the last measurement is not valid, error -230, "Data corrupt or stale" occurs. A measurement is valid after it has been initiated. It becomes invalid when either a reset occurs or any measurement parameter, for example frequency, is changed.
- If the expected power level and resolution parameters are not the same as the current expected value and resolution settings on the specified window/measurement, error -221, "Settings conflict" occurs.

## FETCh[1] | 2 | 3 | 4[:SCALar][:POWer:AC]:DIFFerence? [<expected\_value>[,<resolution>[,<source list>]]]

This query sets the measurement type to a difference measurement with relative mode off for the specified window/measurement, recalculates the measurement, and places the result on the bus. The result is a power-based measurement and is expressed in the unit defined by UNIT[1] | 2 | 3 | 4:POWer.

### Syntax



### Parameters

Refer to “[Optional parameters](#)” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	The N432A ignores the numeric value entered for this parameter. Any value entered is treated like DEF.	Sensor dependent DEF <sup>[1]</sup>
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used. If a value is entered, it should correspond to the current resolution setting otherwise an error occurs.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>



Item	Description/Default	Range of values
source list	Specifies the channels for calculating the difference value. If unspecified and the current window setup is a difference measurement, then this setup is used, otherwise it defaults to A–A.	(@1),(@1)

- [1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.
- [2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

## Example

FETC2 :DIFF?      *Queries the difference measurement with relative mode off for the lower window/upper measurement.*

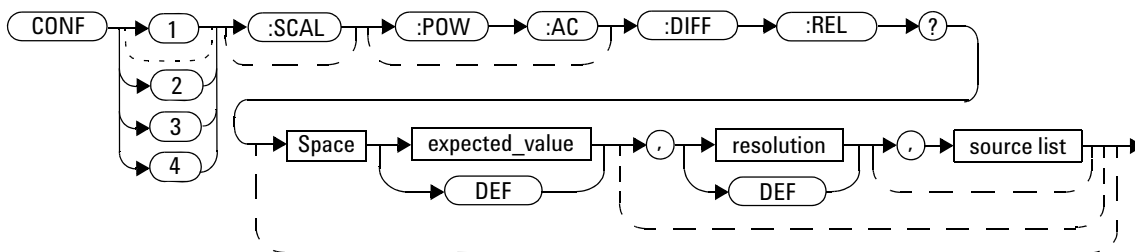
## Error messages

- If the last measurement is not valid, error –230, “Data corrupt or stale” occurs. A measurement is valid after it has been initiated. It becomes invalid when either a reset occurs or any measurement parameter, for example frequency, is changed.
- If the expected power level and resolution parameters are not the same as the current expected value and resolution settings on the specified window/measurement, error –221, “Settings conflict” occurs.

## FETCh[1|2|3|4[:SCALar]][:POWer:AC]:DIFFerence: RELative? [<expected\_value>[,<resolution>[,<source list>]]]

This query sets the measurement type to a difference measurement with relative mode on for the specified window/measurement, recalculates the measurement, and places the result on the bus. The result is a ratio-based measurement and is expressed in the unit defined by UNIT[1|2|3|4:POWer:RATio. The relative value used is that set by the CALCulate:RELative:MAGNitude:AUTO command.

### Syntax



### Parameters

Refer to “[Optional parameters](#)” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	The N432A ignores the numeric value entered for this parameter. Any value entered is treated like DEF.	Sensor dependent DEF <sup>[1]</sup>
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used. If a value is entered, it should correspond to the current resolution setting otherwise an error occurs.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>

Item	Description/Default	Range of values
source list	Specifies the channels for calculating the difference value. If unspecified and the current window setup is a difference measurement, then this setup is used, otherwise it defaults to A–A.	(@1),(@1)

- [1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.
- [2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

## Example

```
FETC:DIFF:REL? DEF, 3, (@1), (@1)
```

*This command queries the relative difference measurement of Channel A – Channel A for the upper window/upper measurement, using the current expected power level and a resolution setting of 3.*

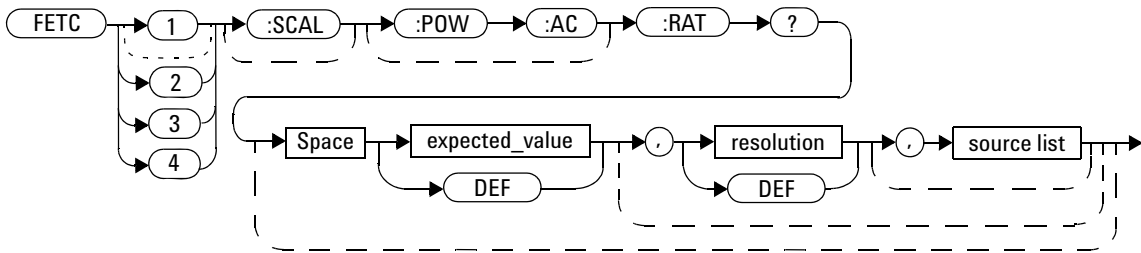
## Error messages

- If the last measurement is not valid, error –230, “Data corrupt or stale” occurs. A measurement is valid after it has been initiated. It becomes invalid when either a reset occurs or any measurement parameter, for example frequency, is changed.
- If the expected power level and resolution parameters are not the same as the current expected value and resolution settings on the specified window/measurement, error –221, “Settings conflict” occurs.

# FETCh[1] | 2 | 3 | 4[:SCALar][:POWer:AC]:RATio? [<expected\_value>[,<resolution>[,<source list>]]]

This query sets the measurement type to a ratio measurement with relative mode off for the specified window/measurement, recalculates the measurement, and places the result on the bus. The result is a ratio-based measurement and is expressed in the unit defined by UNIT[1] | 2 | 3 | 4:POWer:RATio.

## Syntax



## Parameters

Refer to “Optional parameters” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	The N432A ignores the numeric value entered for this parameter. Any value entered is treated like DEF.	Sensor dependent DEF <sup>[1]</sup>
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used. If a value is entered, it should correspond to the current resolution setting otherwise an error occurs.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>

Item	Description/Default	Range of values
source list	Specifies the channels for calculating the ratio value. If unspecified and the current window setup is a ratio measurement, then this setup is used, otherwise it defaults to A/A.	(@1),(@1)

- [1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.
- [2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

## Example

```
FETC2:RAT? DEF,1,(@1),(@1)
```

*Queries the ratio measurement of Channel A/Channel A with relative mode off for the lower window/upper measurement, using the current expected power level and a resolution setting of 1.*

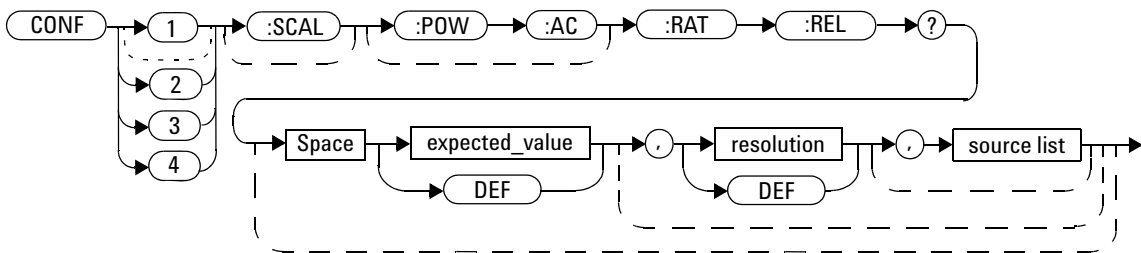
## Error messages

- If the last measurement is not valid, error -230, “Data corrupt or stale” occurs. A measurement is valid after it has been initiated. It becomes invalid when either a reset occurs or any measurement parameter, for example frequency, is changed.
- If the expected power level and resolution parameters are not the same as the current expected value and resolution settings on the specified window/measurement, error -221, “Settings conflict” occurs.

## FETCH[1] | 2 | 3 | 4[:SCALar][:POWer:AC]:RATio:RELative? [<expected\_value>[,<resolution>[,<source list>]]]

This query sets the measurement type to a ratio measurement with relative mode on, recalculates the measurement, and places the result on the bus. The result is a ratio-based measurement and is expressed in the unit defined by UNIT[1] | 2 | 3 | 4:POWer:RATio. The relative value used is that set by the CALCulate:RELative:MAGNitude:AUTO command.

### Syntax



### Parameters

Refer to “Optional parameters” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	The N432A ignores the numeric value entered for this parameter. Any value entered is treated like DEF.	Sensor dependent DEF <sup>[1]</sup>
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used. If a value is entered, it should correspond to the current resolution setting otherwise an error occurs.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>

Item	Description/Default	Range of values
source list	Specifies the channels for calculating the ratio value. If unspecified and the current window setup is a ratio measurement, then this setup is used, otherwise it defaults to A/A.	(@1),(@1)

- [1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.
- [2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

## Example

FETC:RAT:REL?      *Queries the relative ratio measurement on the upper window/upper measurement.*

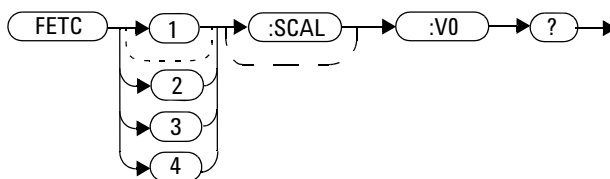
## Error messages

- If the last measurement is not valid, error -230, "Data corrupt or stale" occurs. A measurement is valid after it has been initiated. It becomes invalid when either a reset occurs or any measurement parameter, for example frequency, is changed.
- If the expected power level and resolution parameters are not the same as the current expected value and resolution settings on the specified window/measurement, error -221, "Settings conflict" occurs.

## FETCh[1] | 2 | 3 | 4[:SCALar]:V0?

This query retrieves the  $V_0$  result for the specified window/measurement.  $V_0$  is determined when zeroing is performed with the RF source disconnected. The result is expressed in the VDC unit.

### Syntax



### Example

FETC3:V0?

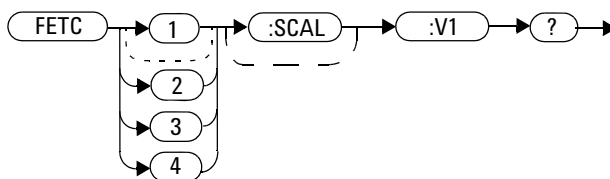
*Queries the  $V_0$  reading of the upper window/lower measurement.*



## FETCh[1]|2|3|4[:SCALar]:V1?

This query retrieves the  $V_1$  result for the specified window/measurement.  $V_1$  is measured after zeroing with the RF source turned on. The result is expressed in the VDC unit.

### Syntax



### Example

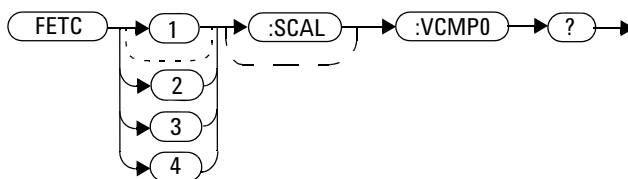
FETC:V1?

*Queries the  $V_1$  reading of the upper window/upper measurement.*

## FETCh[1] | 2 | 3 | 4[:SCALar]:VCMP0?

This query retrieves the  $V_{COMP0}$  result for the specified window/measurement.  $V_{COMP0}$  is determined when zeroing is performed with the RF source disconnected. The result is expressed in the VDC unit.

### Syntax



### Example

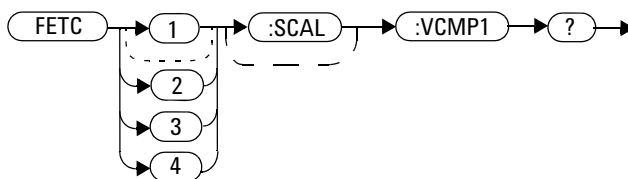
FETC4:VCMP0?

*Queries the  $V_{COMP0}$  reading of the lower window/lower measurement.*

## FETCh[1] | 2 | 3 | 4[:SCALar]:VCMP1?

This query retrieves the  $V_{COMP1}$  result for the specified window/measurement.  $V_{COMP1}$  is measured after zeroing with the RF source turned on. The result is expressed in the VDC unit.

### Syntax



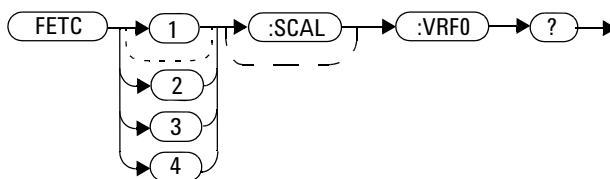
### Example

`FETC:VCMP1?` *Queries the  $V_{COMP1}$  reading of the upper window/upper measurement.*

## FETCh[1] | 2 | 3 | 4[:SCALar]:VRF0?

This query retrieves the  $V_{RF0}$  result for the specified window/measurement.  $V_{RF0}$  is determined when zeroing is performed with the RF source disconnected. The result is expressed in the VDC unit.

### Syntax



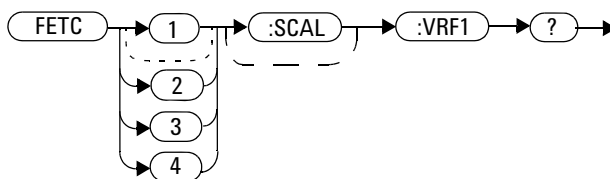
### Example

`FETC3:VRF0?` *Queries the  $V_{RF0}$  reading of the upper window/lower measurement.*

## FETCh[1]|2|3|4[:SCALar]:VRF1?

This query retrieves the  $V_{RF1}$  result for the specified window/measurement.  $V_{RF1}$  is measured after zeroing with the RF source turned on. The result is expressed in the VDC unit.

### Syntax



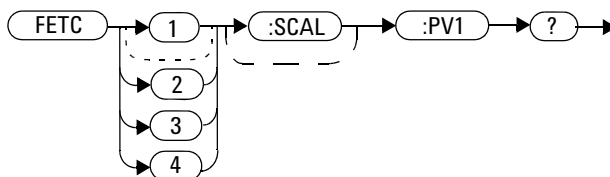
### Example

`FETC:VRF1?` *Queries the  $V_{RF1}$  reading of the upper window/upper measurement.*

## FETCh[1]|2|3|4[:SCALar]:PV1?

This query retrieves the average power measurement value,  $PV_1$  (which is generated in line with voltage measurements) for the specified window/measurement.  $PV_1$  is obtained after any channel offset has been applied but prior to any display offset, and is expressed in the dBm unit.

### Syntax



### Example

`FETC:PV1?`      *Queries the  $PV_1$  reading of the upper window/upper measurement.*

## READ[1]|2|3|4 Queries

The READ? query is most commonly used with the CONFIGure command to cause a new power measurement to be taken and the result returned to the output buffer. The format of the result is set by FORMat[:READING][:DATA]. Refer to [Chapter 7, “FORMat Subsystem”](#) on page 233 for further information.

For the N432A carrying out a single measurement, the READ? query is equivalent to:

```
INITiate
```

```
FETCh?
```

For the N432A carrying out a difference measurement, the READ:DIFFerence? query is equivalent to:

```
INITiate
```

```
FETCh:DIFFerence?
```

For the N432A carrying out a ratio measurement, the READ:RATio? query is equivalent to:

```
INITiate
```

```
FETCh:RATio?
```

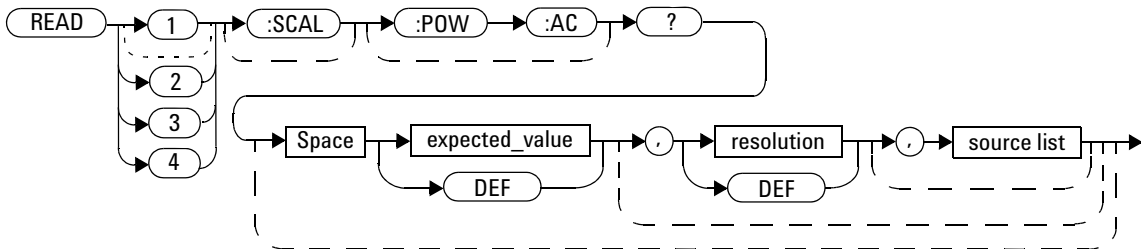
## READ[1|2|3|4[:SCALar][:POWer:AC]? [<expected\_value>[,<resolution>[,<source list>]]]

This query sets the measurement type to single measurement with relative mode off for the specified window/measurement, aborts then initiates the measurement, calculates the measurement result, and places the result on the bus. The result is a power-based measurement and is expressed in the unit defined by `UNIT[1|2|3|4:POWer]`.

### NOTE

`INITiate:CONTinuous` must be set to `OFF`, otherwise error -213, "INIT ignored" occurs. If `TRIGger:SOURce` is set to `BUS`, error -214, "Trigger deadlock" occurs.

### Syntax



### Parameters

Refer to “[Optional parameters](#)” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	This parameter can be set to DEF or a numeric value. If a value is entered, it should correspond to that set by <code>CONFigure</code> otherwise an error occurs. The units of measurement are dBm and W. The default unit is defined by <code>UNIT:POWer</code> .	Sensor dependent DEF <sup>[1]</sup>



Item	Description/Default	Range of values
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used. If a value is entered, it should correspond to the current resolution setting otherwise an error occurs.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>
source list	The channel which the command is implemented on. If unspecified, the current window setup is used.	(@1)

[1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.

[2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

## Example

READ2?      *Queries the lower window/upper measurement.*

## Error messages

- INITiate:CONTinuous must be set to OFF, otherwise error -213, “INIT ignored” occurs.
- If TRIGger:SOURce is set to BUS or HOLD, error -214, “Trigger deadlock” occurs.
- If the expected power level and resolution parameters are not the same as the current expected value and resolution settings on the specified window/measurement, error -221, “Settings conflict” occurs.

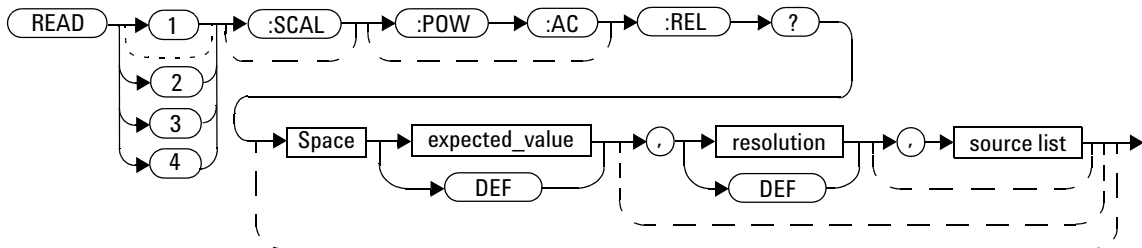
## READ[1|2|3|4[:SCALAR][:POWER:AC]:RELATIVE? [<expected\_value>[,<resolution>[,<source list>]]]

This query sets the measurement type to single measurement with relative mode on for the specified window/measurement, aborts then initiates the measurement, calculates the measurement result, and places the result on the bus. The result is a ratio-based measurement and is expressed in the unit defined by UNIT[1|2|3|4:POWER:RATio. The relative value used is that set by the CALCulate:RELATIVE:MAGNitude:AUTO command.

### NOTE

INITiate:CONTinuous must be set to OFF, otherwise error -213, "INIT ignored" occurs. If TRIGger:SOURce is set to BUS, error -214, "Trigger deadlock" occurs.

### Syntax



### Parameters

Refer to “[Optional parameters](#)” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	This parameter can be set to DEF or a numeric value. If a value is entered, it should correspond to that set by CONFIGure otherwise an error occurs. The units of measurement are dBm and W. The default unit is defined by UNIT:POWER.	Sensor dependent DEF <sup>[1]</sup>

Item	Description/Default	Range of values
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used. If a value is entered, it should correspond to the current resolution setting otherwise an error occurs.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>
source list	The channel which the command is implemented on. If unspecified, the current window setup is used.	(@1)

[1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.

[2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

## Example

```
READ:REL? DEF,1,(@1)
```

*Queries the relative measurement of the upper window/upper measurement, using the current expected power level and a resolution setting of 1.*

## Error messages

- INITiate:CONTinuous must be set to OFF, otherwise error -213, "INIT ignored" occurs.
- If TRIGger:SOURce is set to BUS or HOLD, error -214, "Trigger deadlock" occurs.
- If the expected power level and resolution parameters are not the same as the current expected value and resolution settings on the specified window/measurement, error -221, "Settings conflict" occurs.

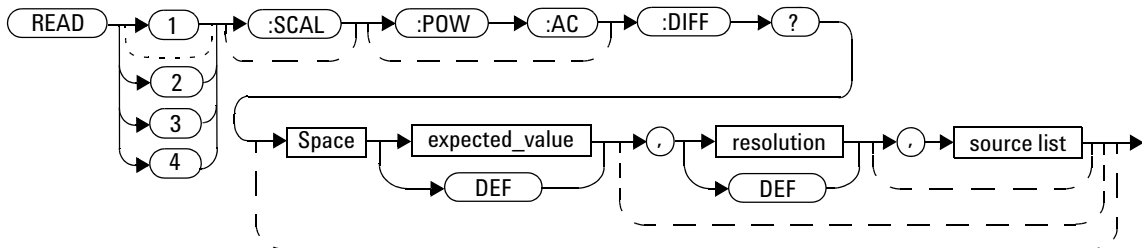
## READ[1|2|3|4[:SCALar]][:POWer:AC]:DIFFerence? [<expected\_value>[,<resolution>[,<source list>]]]

This query sets the measurement type to a difference measurement with relative mode off for the specified window/measurement, aborts then initiates the measurement, calculates the difference measurement result, and places the result on the bus. The result is a power-based measurement and is expressed in the unit defined by UNIT[1] | 2 | 3 | 4 : POWer.

### NOTE

INITiate:CONTinuous must be set to OFF, otherwise error -213, "INIT ignored" occurs. If TRIGger:SOURce is set to BUS, error -214, "Trigger deadlock" occurs.

### Syntax



### Parameters

Refer to “[Optional parameters](#)” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	The N432A ignores the numeric value entered for this parameter. Any value entered is treated like DEF.	Sensor dependent DEF <sup>[1]</sup>

Item	Description/Default	Range of values
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used. If a value is entered, it should correspond to the current resolution setting otherwise an error occurs.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>
source list	Specifies the channels for calculating the difference value. If unspecified and the current window setup is a difference measurement, then this setup is used, otherwise it defaults to A–A.	(@1),(@1)

[1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.

[2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

## Example

READ2:DIFF?      *Queries the difference measurement with relative mode off for the lower window/upper measurement.*

## Error messages

- INITiate:CONTInuous must be set to OFF, otherwise error -213, “INIT ignored” occurs.
- If TRIGger:SOURce is set to BUS or HOLD, error -214, “Trigger deadlock” occurs.
- If the resolution parameter is not the same as the current resolution setting on the specified window/measurement, error -221, “Settings conflict” occurs.

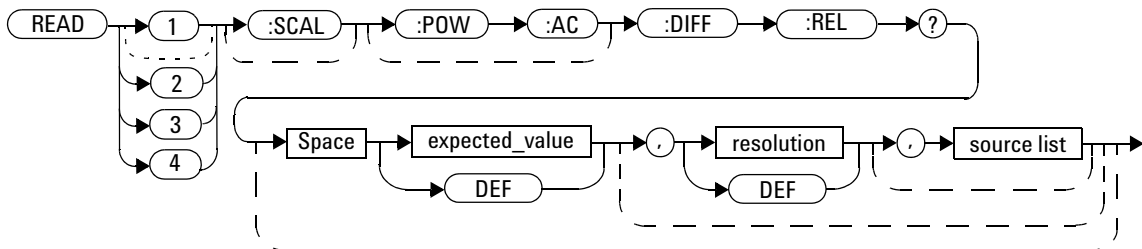
## READ[1] | 2 | 3 | 4[:SCALar][:POWer:AC]:DIFFerence:RELative? [<expected\_value>[,<resolution>[,<source list>]]]

This query sets the measurement type to a difference measurement with relative mode on for the specified window/measurement, aborts then initiates the measurement, calculates the difference measurement result, and places the result on the bus. The result is a ratio-based measurement and is expressed in the unit defined by UNIT[1] | 2 | 3 | 4:POWER:RATio. The relative value used is that set by the CALCulate:RELative:MAGNitude:AUTO command.

### NOTE

INITiate:CONTinuous must be set to OFF, otherwise error -213, "INIT ignored" occurs. If TRIGger:SOURce is set to BUS, error -214, "Trigger deadlock" occurs.

### Syntax



### Parameters

Refer to “Optional parameters” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	The N432A ignores the numeric value entered for this parameter. Any value entered is treated like DEF.	Sensor dependent DEF <sup>[1]</sup>

Item	Description/Default	Range of values
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used. If a value is entered, it should correspond to the current resolution setting otherwise an error occurs.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>
source list	Specifies the channels for calculating the difference value. If unspecified and the current window setup is a difference measurement, then this setup is used, otherwise it defaults to A–A.	(@1),(@1)

[1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.

[2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

## Example

```
READ:DIFF:REL? DEF,4,(@1),(@1)
```

*Queries the relative difference measurement of Channel A – Channel A for the upper window/upper measurement, using the current expected power level and a resolution setting of 4.*

## Error messages

- INITiate:CONTinuous must be set to OFF, otherwise error –213, “INIT ignored” occurs.
- If TRIGger:SOURce is set to BUS or HOLD, error –214, “Trigger deadlock” occurs.
- If the resolution parameter is not the same as the current resolution setting on the specified window/measurement, error –221, “Settings conflict” occurs.

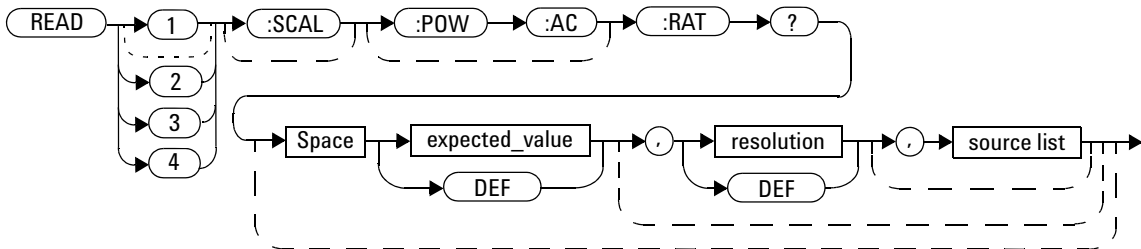
## READ[1|2|3|4[:SCALar][:POWer:AC]:RATio? [<expected\_value>[,<resolution>[,<source list>]]]

This query sets the measurement type to a ratio measurement with relative mode off for the specified window/measurement, aborts then initiates the measurement, calculates the ratio measurement result, and places the result on the bus. The result is a ratio-based measurement and is expressed in the unit defined by UNIT[1|2|3|4:POWer:RATio.

### NOTE

INITiate:CONTinuous must be set to OFF, otherwise error -213, "INIT ignored" occurs. If TRIGger:SOURce is set to BUS, error -214, "Trigger deadlock" occurs.

### Syntax



### Parameters

Refer to “[Optional parameters](#)” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	The N432A ignores the numeric value entered for this parameter. Any value entered is treated like DEF.	Sensor dependent DEF <sup>[1]</sup>



Item	Description/Default	Range of values
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used. If a value is entered, it should correspond to the current resolution setting otherwise an error occurs.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>
source list	Specifies the channels for calculating the ratio value. If unspecified and the current window setup is a ratio measurement, then this setup is used, otherwise it defaults to A/A.	(@1),(@1)

[1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.

[2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

## Example

READ2:RAT? DEF,1,(@1),(@1) *Queries the ratio measurement of Channel A/Channel A with relative mode off for the lower window/upper measurement, using the current expected power level and a resolution setting of 1.*

## Error messages

- INITiate:CONTinuous must be set to OFF, otherwise error -213,“INIT ignored” occurs.
- If TRIGger:SOURce is set to BUS or HOLD, error -214,“Trigger deadlock” occurs.
- If the resolution parameter is not the same as the current resolution setting on the specified window/measurement, error -221,“Settings conflict” occurs.

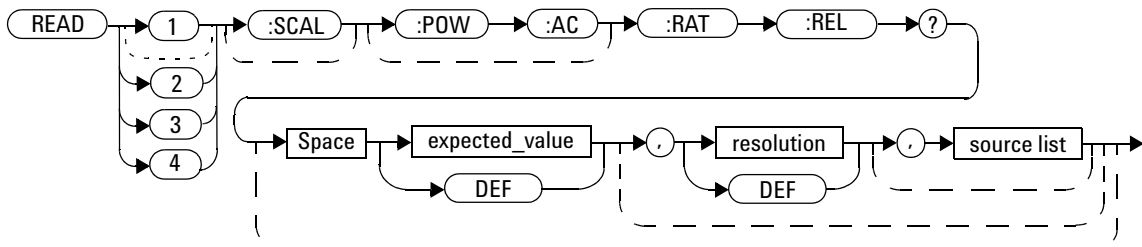
## READ[1] | 2 | 3 | 4[:SCALar][:POWer:AC]:RATio:RELative? [<expected\_value>[,<resolution>[,<source list>]]]

This query sets the measurement type to a ratio measurement with relative mode on for the specified window/measurement, aborts then initiates the measurement, calculates the ratio measurement result using the new sensor data, and places the result on the bus. The result is a ratio-based measurement and is expressed in the unit defined by UNIT[1] | 2 | 3 | 4:POWER:RATio. The relative value used is that set by the CALCulate:RELative:MAGNitude:AUTO command.

### NOTE

INITiate:CONTinuous must be set to OFF, otherwise error -213, "INIT ignored" occurs. If TRIGger:SOURce is set to BUS, error -214, "Trigger deadlock" occurs.

### Syntax



### Parameters

Refer to “[Optional parameters](#)” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	The N432A ignores the numeric value entered for this parameter. Any value entered is treated like DEF.	Sensor dependent DEF <sup>[1]</sup>

Item	Description/Default	Range of values
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used. If a value is entered, it should correspond to the current resolution setting otherwise an error occurs.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>
source list	Specifies the channels for calculating the ratio value. If unspecified and the current window setup is a ratio measurement, then this setup is used, otherwise it defaults to A/A.	(@1),(@1)

[1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.

[2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

## Example

READ:RAT:REL?      *Queries the relative ratio measurement for the upper window/upper measurement.*

## Error messages

- INITiate:CONTinuous must be set to OFF, otherwise error -213, "INIT ignored" occurs.
- If TRIGger:SOURce is set to BUS or HOLD, error -214, "Trigger deadlock" occurs.
- If the resolution parameter is not the same as the current resolution setting on the specified window/measurement, error -221, "Settings conflict" occurs.

## MEASure[1]|2|3|4 Queries

The MEASure? commands configure the N432A to perform a power measurement with the given measurement type, relative mode setting, range, and resolution, then makes the measurement. The format of the result is set by FORMat[:READing][:DATA]. Refer to [Chapter 7, “FORMat Subsystem”](#) on page 233 for further information.

MEASure? is a compound command which is equivalent to:

```
CONFigure  
READ?
```

For the N432A carrying out a single measurement, the MEASure? query is equivalent to:

```
CONFigure  
READ?
```

For the N432A carrying out a difference measurement, the MEASure:DIFFerence? query is equivalent to:

```
CONFigure:DIFFerence  
READ:DIFFerence?
```

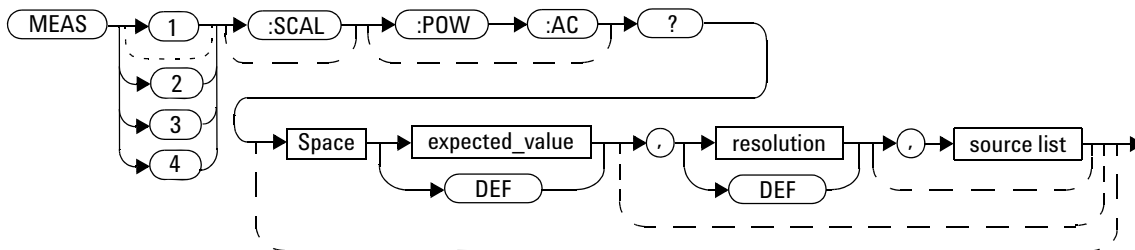
For the N432A carrying out a ratio measurement the MEASure:RATio? query is equivalent to:

```
CONFigure:RATio  
READ:RATio?
```

## MEASure[1] | 2 | 3 | 4[:SCALar][:POWer:AC]? [<expected\_value>[,<resolution>[,<source list>]]]

This query sets the measurement type to single measurement with relative mode off for the specified window/measurement, aborts, configures the window then initiates the measurement, calculates the measurement result, and places the result on the bus.

### Syntax



### Parameters

Refer to “[Optional parameters](#)” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	This parameter can be set to <b>DEF</b> or a numeric value. The units of measurement are dBm and W. The default unit is defined by <b>UNIT:POWER</b> .	Sensor dependent <b>DEF</b> <sup>[1]</sup>
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 <b>DEF</b> <sup>[1]</sup>
source list	The channel which the command is implemented on. If unspecified, the current window setup is used.	(@1)

### 3 MEASurement Commands

- [1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.
- [2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

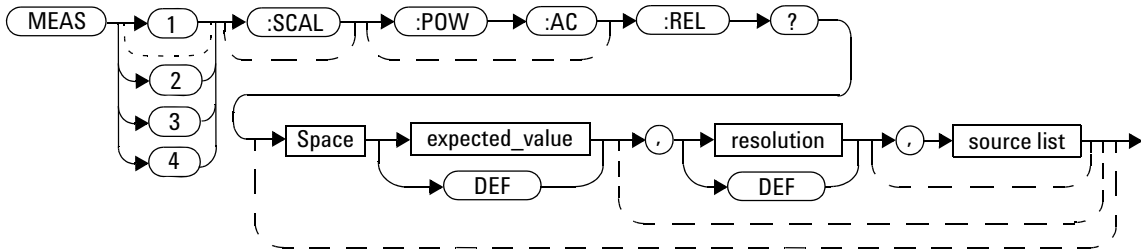
#### Example

MEAS2? -70DBM, 1, (@1)      *Queries the lower window/upper measurement, using an expected power level of -70 dBm and a resolution setting of 1.*

# MEASure[1] | 2 | 3 | 4[:SCALar][:POWer:AC]:RELative? [<expected\_value>[,<resolution>[,<source list>]]]

This query sets the measurement type to single measurement with relative mode on for the specified window/measurement, aborts, configures then initiates the measurement, calculates the measurement result, and places the result on the bus. The result is a ratio-based measurement and is expressed in the unit defined by UNIT[1] | 2 | 3 | 4:POWer:RATio. The relative value used is that set by the CALCulate:RELative:MAGNitude:AUTO command.

## Syntax



## Parameters

Refer to “Optional parameters” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	This parameter can be set to DEF or a numeric value. The units of measurement are dBm and W. The default unit is defined by UNIT:POWER.	Sensor dependent DEF <sup>[1]</sup>
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>

### 3 MEASurement Commands

Item	Description/Default	Range of values
source list	The channel which the command is implemented on. If unspecified, the current window setup is used.	(@1)

- [1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.
- [2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

## Example

```
MEAS:REL? -50DBM, 2, (@1)
```

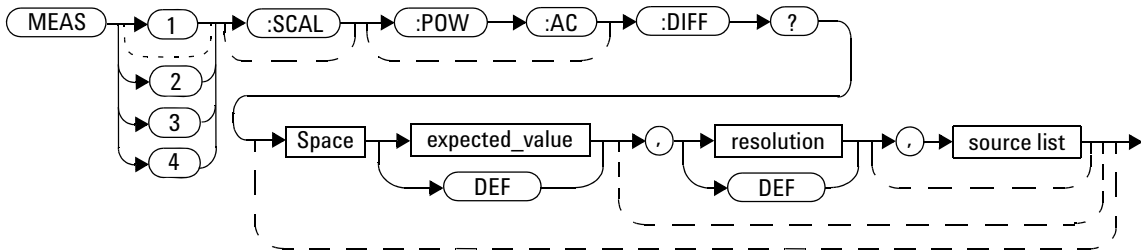
*Queries the relative measurement of the upper window/upper measurement, using an expected power level of -50 dBm and a resolution setting of 2.*



# MEASure[1] | 2 | 3 | 4[:SCALar][:POWer:AC]:DIFFerence? [<expected\_value>[,<resolution>[,<source list>]]]

This query sets the measurement type to a difference measurement with relative mode off for the specified window/measurement, aborts, configures then initiates the measurement, calculates the difference measurement result, and places the result on the bus. The result is a power-based measurement and is expressed in the unit defined by UNIT[1] | 2 | 3 | 4:POWer.

## Syntax



## Parameters

Refer to “Optional parameters” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	The N432A ignores the numeric value entered for this parameter. Any value entered is treated like DEF.	Sensor dependent DEF <sup>[1]</sup>
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>

### 3 MEASurement Commands

Item	Description/Default	Range of values
source list	Specifies the channels for calculating the difference value. If unspecified and the current window setup is a difference measurement, then this setup is used, otherwise it defaults to A–A.	(@1),(@1)

[1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.

[2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

## Example

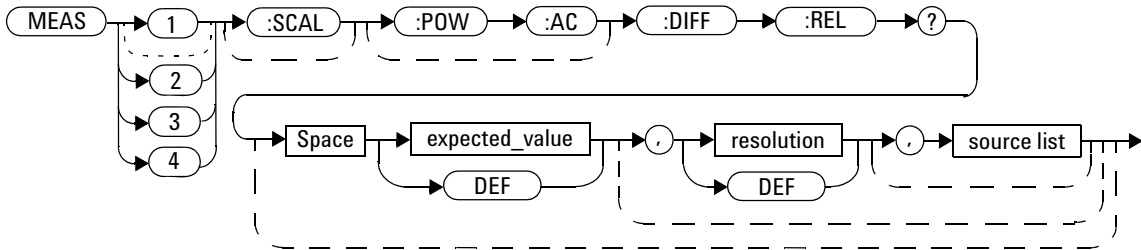
MEAS2:DIFF?

*Queries the difference measurement with relative mode off for the lower window/upper measurement.*

# MEASure[1]|2|3|4[:SCALar][:POWer:AC]:DIFFerence:RELative? [<expected\_value>[,<resolution>[,<source list>]]]

This query sets the measurement type to a difference measurement with relative mode on for the specified window/measurement, aborts, configures then initiates the measurement, calculates the difference measurement result, and places the result on the bus. The result is a ratio-based measurement and is expressed in the unit defined by UNIT[1]|2|3|4:POWer:RATio. The relative value used is that set by the CALCulate:RELative:MAGNitude:AUTO command.

## Syntax



## Parameters

Refer to “[Optional parameters](#)” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	The N432A ignores the numeric value entered for this parameter. Any value entered is treated like DEF.	Sensor dependent DEF <sup>[1]</sup>
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>

Item	Description/Default	Range of values
source list	Specifies the channels for calculating the difference value. If unspecified and the current window setup is a difference measurement, then this setup is used, otherwise it defaults to A–A.	(@1),(@1)

- [1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.
- [2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

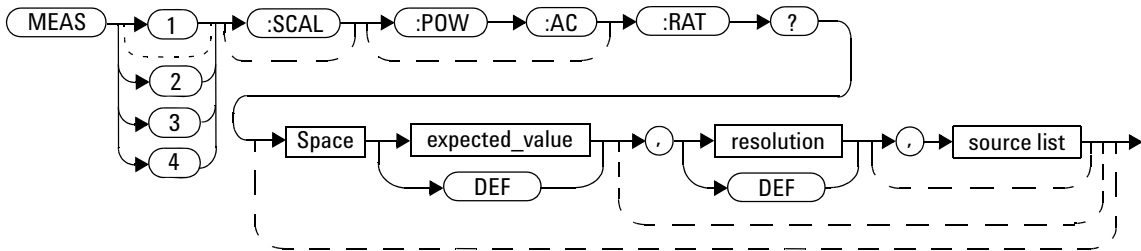
### Example

MEAS:DIFF:REL? DEF,3,(@1),  
 (@1) *Queries the relative difference measurement of Channel A – Channel A for the upper window/upper measurement, using the current expected power level and a resolution setting of 3.*

# MEASure[1] | 2 | 3 | 4[:SCALar][:POWer:AC]:RATio? [<expected\_value>[,<resolution>[,<source list>]]]

This query sets the measurement type to a ratio measurement with relative mode off for the specified window/measurement, aborts, configures then initiates the measurement, calculates the ratio measurement result, and places the result on the bus. The result is a ratio-based measurement and is expressed in the unit defined by UNIT[1] | 2 | 3 | 4:POWer:RATio.

## Syntax



## Parameters

Refer to “Optional parameters” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	The N432A ignores the numeric value entered for this parameter. Any value entered is treated like DEF.	Sensor dependent DEF <sup>[1]</sup>
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>

Item	Description/Default	Range of values
source list	Specifies the channels for calculating the ratio value. If unspecified and the current window setup is a ratio measurement, then this setup is used, otherwise it defaults to A/A.	(@1),(@1)

[1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.

[2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

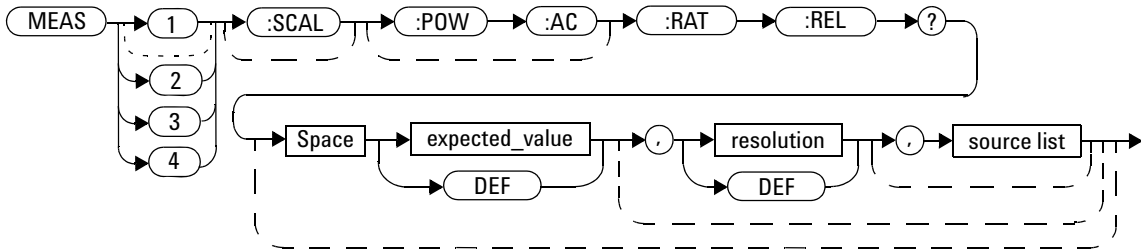
## Example

MEAS2:RAT? DEF,1,(@1),(@1) *Queries the ratio measurement of Channel A/Channel A with relative mode off for the lower window/upper measurement, using the current expected power level and a resolution setting of 1.*

# MEASure[1] | 2 | 3 | 4[:SCALar][:POWer:AC]:RATio:RELative? [<expected\_value>[,<resolution>[,<source list>]]]

This query sets the measurement type to a ratio measurement with relative mode on for the specified window/measurement, aborts, configures then initiates the measurement, calculates the ratio measurement result, and places the result on the bus. The result is a ratio-based measurement and is expressed in the unit defined by UNIT[1] | 2 | 3 | 4:POWER:RATio. The relative value used is that set by the CALCulate:RELative:MAGNitude:AUTO command.

## Syntax



## Parameters

Refer to “Optional parameters” on page 104 for additional details on the parameters in this command.

Item	Description/Default	Range of values
expected_value (for the expected power level)	The N432A ignores the numeric value entered for this parameter. Any value entered is treated like DEF.	Sensor dependent DEF <sup>[1]</sup>
resolution	A numeric value for the resolution. If unspecified, the current resolution setting is used.	1 to 4 <sup>[2]</sup> 1.0, 0.1, 0.01, 0.001 DEF <sup>[1]</sup>

### 3 MEASurement Commands

Item	Description/Default	Range of values
source list	Specifies the channels for calculating the ratio value. If unspecified and the current window setup is a ratio measurement, then this setup is used, otherwise it defaults to A/A.	(@1),(@1)

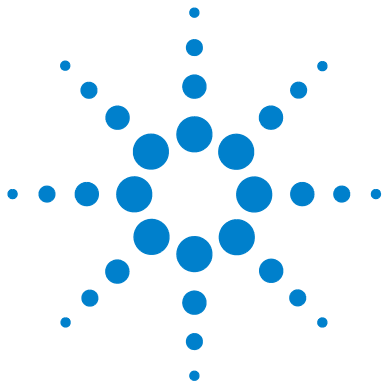
- [1] The mnemonic DEF means DEFault. This is not equivalent to the DEFault parameter used in the command subsystems. The parameters must be entered in the specified order. If parameters are omitted, they default from the right. The parameter DEFault is used as a placeholder. Specifying DEF leaves the parameter value unchanged.
- [2] When the measurement result is linear, this parameter represents the number of significant digits. When the measurement result is logarithmic, 1 to 4 represent 1, 0.1, 0.01, and 0.001 respectively.

## Example

MEAS : RAT : REL ?

*Queries the relative ratio measurement on the upper window/upper measurement.*





## 4 CALCulate Subsystem

CALCulate Subsystem	170
CALCulate[1] 2 3 4:HOLD:STATe <character_data>	173
CALCulate[1] 2 3 4:FEED[1] 2 <string>	175
CALCulate[1] 2 3 4:GAIN Commands	177
CALCulate[1] 2 3 4:GAIN[:MAGNitude] <numeric_value>	178
CALCulate[1] 2 3 4:GAIN:STATe <boolean>	180
CALCulate[1] 2 3 4:LIMit Commands	181
CALCulate[1] 2 3 4:LIMit:CLEar:AUTO <boolean> ONCE	182
CALCulate[1] 2 3 4:LIMit:CLEar[:IMMEDIATE]	184
CALCulate[1] 2 3 4:LIMit:FAIL?	185
CALCulate[1] 2 3 4:LIMit:FCOunt?	186
CALCulate[1] 2 3 4:LIMit:LOWer[:DATA] <numeric_value>	188
CALCulate[1] 2 3 4:LIMit:UPPer[:DATA] <numeric_value>	190
CALCulate[1] 2 3 4:LIMit:STATe <boolean>	192
CALCulate[1] 2 3 4:MATH Commands	193
CALCulate[1] 2 3 4:MATH[:EXPReSSion] <string>	194
CALCulate[1] 2 3 4:MATH[:EXPReSSion]:CATalog?	196
CALCulate[1] 2 3 4:RELative Commands	197
CALCulate[1] 2 3 4:RELative[:MAGNitude]:AUTO <boolean> ONCE	198
CALCulate[1] 2 3 4:RELative:STATe <boolean>	200

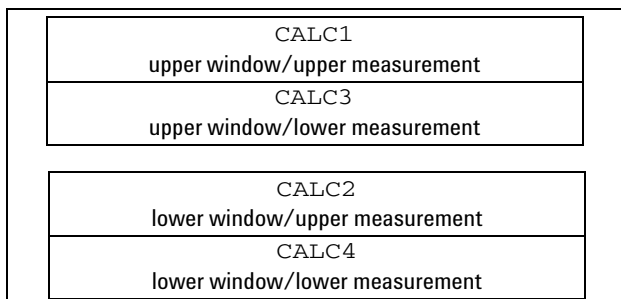
This chapter explains how the CALCulate subsystem is used to perform post-acquisition data processing.



## CALCulate Subsystem

The CALCulate subsystem provides the commands to configure the parameters used in post-acquisition data processing. Functions in the SENSE subsystem are related to data acquisition, while the CALCulate subsystem operates on the data acquired by a SENSE function.

There are four independent CALCulate blocks in the N432A; two for each window, as shown in the figure below. The numeric suffix of the CALCulate command determines which CALCulate block is used and where the measurement result is displayed.



**Figure 4-1** Measurement display CALCulate block window

Data from the SENSE block may feed any or all of the CALCulate blocks via the MATH command. [Figure 4-2](#) details where the commands are applied with in the CALCulate block.

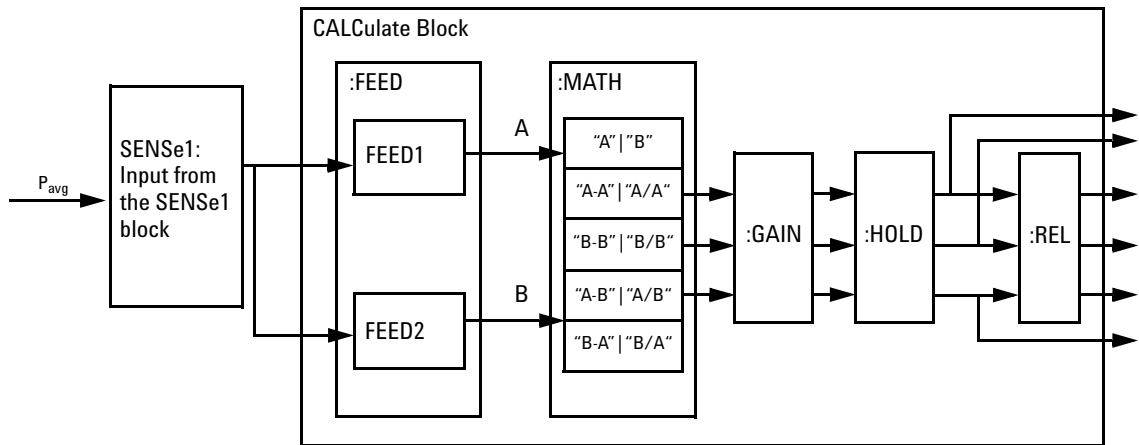


Figure 4-2 CALCulate block

Keyword	Parameter form	Note	Page
CALCulate[1]   2   3   4			
:HOLD			
:STATe	<character_data>		page 173
:FEED[1]   2	<string>		page 175
:GAIN			
[:MAGNitude]	<numeric_value>		page 178
:STATe	<boolean>		page 180
:LIMit			
:CLEar			
:AUTO	<boolean>   ONCE		page 182
[:IMMediate]			page 184
:FAIL?		[query only]	page 185
:FCOunt?		[query only]	page 186
:LOWer			
[:DATA]	<numeric_value>		page 188
:UPPer			
[:DATA]	<numeric_value>		page 190

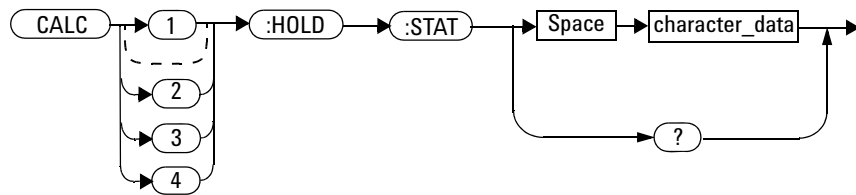
## 4 CALCulate Subsystem

Keyword	Parameter form	Note	Page
:STATe	<boolean>		<a href="#">page 192</a>
:MATH			
[:EXPRession]	<string>		<a href="#">page 194</a>
:CATalog?		[query only]	<a href="#">page 196</a>
:RELative			
[:MAGNitude]			
:AUTO	<boolean>   ONCE		<a href="#">page 198</a>
:STATe	<boolean>		<a href="#">page 200</a>

## CALCulate[1|2|3|4:HOLD:STATe <character\_data>

This command sets the minimum or maximum hold value of the power measured for the specified window/measurement.

### Syntax



### Parameter

Item	Description	Range of values
character_data	Sets the Hold value to Off	OFF
	Sets the Hold value to Minimum	MIN
	Sets the Hold value to Maximum	MAX

### Example

CALC2:HOLD:STAT MIN

*This command sets the hold value to minimum power detected for the lower window/upper measurement.*

### Reset condition

On reset, the hold value state is set to OFF.

### Query

```
CALC[1] | 2 | 3 | 4 : HOLD : STATe?
```

This query returns the current state of the hold value.

### Query example

```
CALC2 : HOLD : STAT?
```

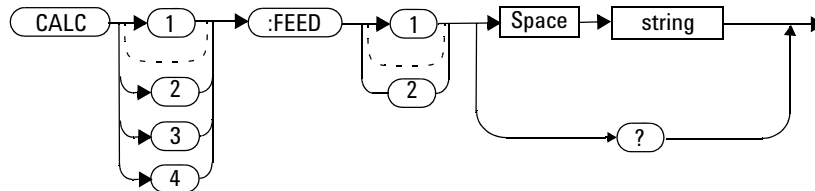
*Queries the state of the hold value for the lower window/upper measurement.*

## CALCulate[1|2|3|4:FEED[1|2] <string>

This command sets the input measurement mode to be fed to the specified input on the CALC block. It is applied to the measurement after the CALC:MATH:EXPR command has been used to specify the measurement type for the feed.

Measurement modes are coupled for combination measurements (for example, ratio measurements).

### Syntax



### Parameter

Item	Description	Range of values
string	<p>The input measurement mode to be fed to the specific input on the CALC block:</p> <ul style="list-style-type: none"> <li>• AVER: average</li> </ul> <p>A feed of "" (empty string) disables the CALC block and switches off that display line.</p>	"POW:AVER"

### Example

`CALC:FEED1 "POW:AVER"` *This command sets the input measurement mode for feed 1 of the CALC1 block to average power.*

### Reset condition

On reset, `<string>` is set to `POW:AVER`.

### Query

`CALCulate[1]|2|3|4:FEED[1]|2?`

This query returns the current input measurement mode for the specified CALC block and feed.

### Query example

`CALC1:FEED2?` *Queries the input measurement mode for feed 2 of the upper window/upper measurement.*

### Error message

If the command is executed when no sensor is attached, error -241, "Hardware missing" occurs.



## CALCulate[1]|2|3|4:GAIN Commands

These commands are used to enter and enable a display offset on the specified window/measurement. The display offset is applied to the measured power after any channel offset or mathematical functions have been included.

The following commands are detailed in this section:

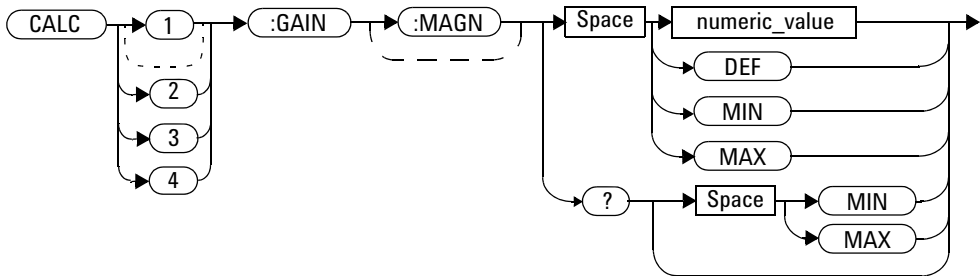
```
CALCulate[1]|2|3|4:GAIN[:MAGNitude] <numeric value>  
CALCulate[1]|2|3|4:GAIN:STATe <boolean>
```

## CALCulate[1|2|3|4:GAIN[:MAGNitude] <numeric\_value>

This command is used to enter a value for the display offset on the specified window/measurement.

Entering a value using this command automatically turns ON the CALCulate[1|2|3|4:GAIN:STATe command.

### Syntax



### Parameter

Item	Description/Default	Range of values
numeric_value	A numeric value for the display offset <ul style="list-style-type: none"> <li>• DEF: The default value is 0 dB</li> <li>• MIN: -100.000 dB</li> <li>• MAX: +100.000 dB</li> </ul>	-100.000 dB to +100.000 dB DEF MIN MAX

## Example

`CALC2:GAIN 20`      *This command sets a display offset of 20 dB for the lower window/upper measurement.*

## Reset condition

On reset, the display offset is set to 0 dB.

## Query

`CALCulate[1]|2|3|4:GAIN[:MAGNitude]? [MIN|MAX]`

This query returns the current setting of the display offset or the value associated with MIN and MAX.

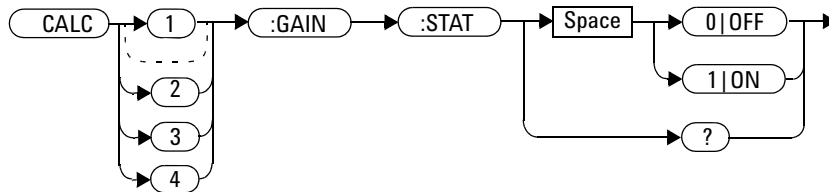
## Query example

`CALC1:GAIN?`      *Queries the current setting of the display offset on the upper window/upper measurement.*

## CALCulate[1]|2|3|4:GAIN:STATe <boolean>

This command is used on the specified window/measurement to enable or disable the display offset set by the CALCulate[1]|2|3|4:GAIN[:MAGNitude] command.

### Syntax



### Example

`CALC2:GAIN:STAT 1`      *This command enables the display offset for the lower window/upper measurement.*

### Reset Condition

On reset, the gain is disabled.

### Query

`CALCulate[1]|2|3|4:GAIN:STATe?`

This query enters a 1 or 0 into the output buffer indicating the status of the display offset.

- 1 is returned when the display offset is enabled
- 0 is returned when the display offset is disabled

### Query example

`CALC:GAIN:STAT?`      *Queries whether the display offset in the upper window/upper measurement is enabled or disabled.*

## CALCulate[1] | 2 | 3 | 4:LIMit Commands

These commands set the limits on the upper and lower windows/measurements enabling you to:

- Set upper and lower level limits
- Query if there has been a failure
- Count the number of failures
- Clear the counter

The following commands are detailed in this section:

```
CALCulate[1] | 2 | 3 | 4:LIMit:CLEar:AUTO <boolean>
```

```
CALCulate[1] | 2 | 3 | 4:LIMit:CLEar[IMMEDIATE]
```

```
CALCulate[1] | 2 | 3 | 4:LIMit:FAIL?
```

```
CALCulate[1] | 2 | 3 | 4:LIMit:FCOUNT?
```

```
CALCulate[1] | 2 | 3 | 4:LIMit:LOWer[:DATA]
```

```
CALCulate[1] | 2 | 3 | 4:LIMit:UPPer[:DATA]
```

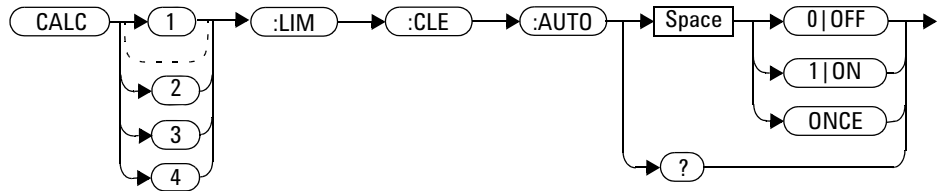
```
CALCulate[1] | 2 | 3 | 4:LIMit:STATE <boolean>
```

## CALCulate[1] | 2 | 3 | 4:LIMit:CLEAr:AUTO <boolean> | ONCE

This command controls when the fail counter (FCO) is cleared of any limit failures for the specified window/measurement. The FCO is used to determine the results returned by the CALCulate[1] | 2 | 3 | 4:LIMit:FAIL? query.

- If ON is specified, the FCO is set to 0 each time a measurement is:
  - Initiated using INITiate[:IMMEDIATE]
  - Initiated using INITiate:CONTinuous ON
  - Measured using MEASure?
  - Read using READ?
- If OFF is specified, the FCO is not cleared by the above commands.
- If ONCE is specified, the FCO is cleared only after the first initialization then starts accumulating any limit failures.

### Syntax



### Example

```
CALC:LIM:CLE:AUTO 1
```

*This command switches on automatic clearing of the FCO for the upper window/upper measurement.*

## Reset condition

On reset, both windows and their measurements are set to ON.

## Query

```
CALCulate[1]|2|3|4:LIMit:CLEar:AUTO?
```

This query enters a 1 or 0 into the output buffer indicating whether limit failures are cleared automatically when a new measurement is initiated on the specified window section.

- 1 is returned if limit failures are cleared automatically when a new measurement is initiated.
- 0 is returned if limit failures are not cleared automatically when a new measurement is initiated.

In the case where limit failures are cleared once, when a query occurs, a 1 is entered into the output buffer if no measurement is initiated. If a measurement is initiated, then 0 is entered.

## Query example

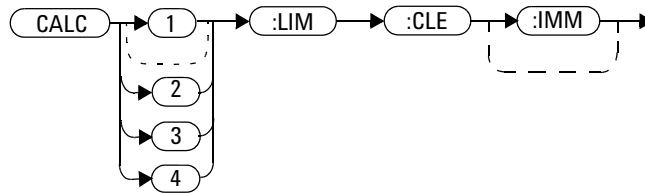
```
CALC4:LIM:CLE:AUTO?
```

*Queries whether the FCO is cleared for the lower window/lower measurement.*

## CALCulate[1] | 2 | 3 | 4:LIMit:CLEar[:IMMEDIATE]

This command immediately clears the fail counter (FCO) of any limit failures for the specified window/measurement. The FCO is used to determine the results returned by the CALCulate[1] | 2 | 3 | 4:LIMit:FAIL? query.

### Syntax



### Example

```
CALC2:LIM:CLE
```

*This command immediately clears the FCO for the lower window/upper measurement.*

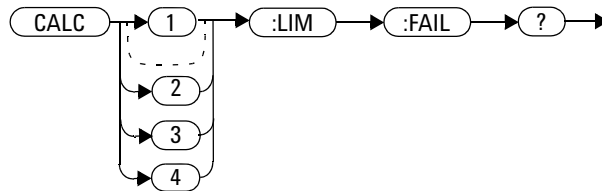


## CALCulate[1]|2|3|4:LIMit:FAIL?

This query enters a 1 or 0 into the output buffer indicating whether there have been any limit failures for the specified window/measurement. A limit failure is defined as `CALC[1]|2|3|4:LIMit:FCO?` being non-zero. The fail counter (FCO) can be zeroed using the `CALC[1]|2|3|4:LIMit:CLear` command.

- 1 is returned if one or more limit failures have occurred
- 0 is returned if no limit failures have occurred

### Syntax



### Example

`CALC1:LIM:FAIL?`

*Queries if there have been any limit failures for the upper window/upper measurement.*

### Reset condition

On reset, the buffer is set to zero for both the upper and lower windows/measurements.

## CALCulate[1]|2|3|4:LIMit:FCOunt?

This query returns the total number of limit failures for the specified window/measurement.

If the appropriate STATE commands are set to ON, each time a measurement is initiated on the specified window/measurement and the result is outside the limits, the counter is incremented by one.

If the measured value is equal to a limit, this is a limit pass.

The counter is reset to zero by any of the following commands:

- \*RST
- CALCulate[1]|2|3|4:LIMit:CLEar[:IMMediate]
- CALCulate[1]|2|3|4:LIMit:CLEar:AUTO ON

When CALCulate[1]|2|3|4:LIMit:CLEar:AUTO is set to ON, the counter is set to zero *each* time a measurement is:

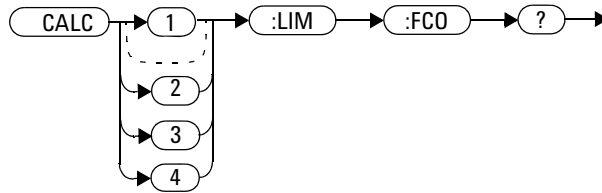
- measured using MEASure?
- read using READ?
- initiated using:
  - INITiate[:IMMediate] or,
  - INITiate:CONTinuous ON

When CALCulate[1]|2|3|4:LIMit:CLEar:AUTO is set to ONCE, the counter is set to zero the *first* time a measurement is:

- measured using MEASure?
- read using READ?
- initiated using:
  - INITiate[:IMMediate] or,
  - INITiate:CONTinuous ON

The maximum number of errors is  $2^{16}-1$ . If more than  $2^{16}-1$  errors are detected, the counter returns to zero.

## Syntax



## Example

`CALC:LIM:FCO?`

*Queries the number of limit failures for the upper window/upper measurement.*

## Reset condition

On reset, the counter is set to zero for both measurements of the upper and lower windows.

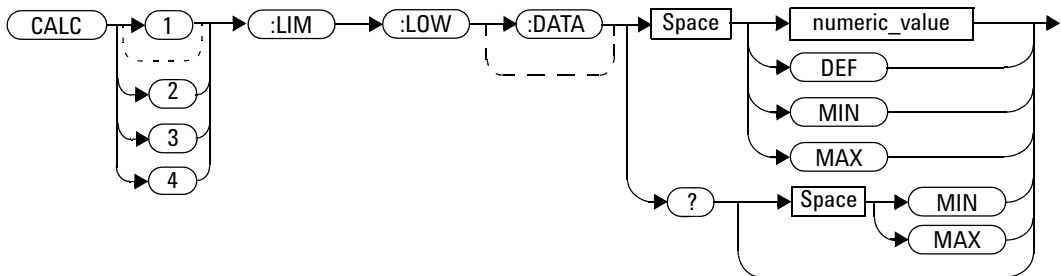
## CALCulate[1]|2|3|4:LIMit:LOWer[:DATA] <numeric\_value>

This command enters a value for the lower test limit for the specified window/measurement used in the CALCulate[1]|2|3|4:LIMit:FAIL? test. The units used are dependent on the current settings of UNIT:POWER and CALCulate:RELative:STATE as shown in Table 4-1. When the measured value is less than the value specified in CALCulate[1]|2|3|4:LIMit:LOWer[:DATA], CALCulate[1]|2|3|4:LIMit:FAIL? reports a fail. When the measured value is greater than or equal to the limit, a fail is not reported.

**Table 4-1** Measurement units

Measurement mode	CALC:REL:STAT OFF		CALC:REL:STAT ON	
	Linear	Log	Linear	Log
Single	Watt	dBm	%	dB
	%	dB	%	dB
Ratio	%	dB	%	dB
Difference	Watt	dBm	%	dB
	%	dB	%	dB

### Syntax



## Parameter

Item	Description/Default	Range of values
numeric_value	A numeric value for the lower test limit <ul style="list-style-type: none"> <li>• DEF: The default is -90.00 dBm or -90 dB</li> <li>• MIN: -150 dBm or -180 dB</li> <li>• MAX: +230 dBm or +200 dB</li> </ul>	-150 to +230 dBm or -180 to +200 dB  DEF  MIN  MAX

## Example

`CALC3:LIM:LOW 0.1` *This command enters a lower limit for the upper window/lower measurement based on the window units as follows:*

*dBm = 0.1 dBm*  
*W = 100 mW*  
*dB = 0.1 dB*  
*% = 0.1%*

## Reset condition

On reset, both measurements of the upper and lower windows are set to -90.00 dBm or -90 dB.

## Query

`CALCulate[1] | 2 | 3 | 4:LIMit:LOWer[:DATA]? [MIN|MAX]`

This query returns the current setting of the lower limit or the values associated with MIN and MAX for the specified window/measurement.

## Query example

`CALC2:LIM:LOW?` *Queries the lower limit setting of the lower window/upper measurement.*

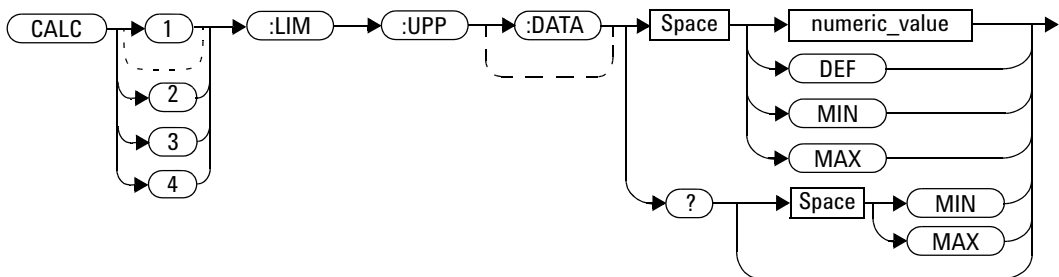
## CALCulate[1|2|3|4:LIMit:UPPer[:DATA] <numeric\_value>

This command enters a value for the upper test limit for the specified window/measurement used in the CALCulate[1|2|3|4:LIMit:FAIL? test. The units used are dependent on the current settings of UNIT:POWER and CALCulate:RELative:STATE as shown in Table 4-2. When the measured power is greater than the value specified in CALCulate[1|2|3|4:LIMit:UPPer[:DATA], CALCulate[1|2|3|4:LIMit:FAIL? reports a fail. When the measured power is less than or equal to the limit, a fail is not reported.

Table 4-2 Measurement units

Measurement mode	CALC:REL:STAT OFF		CALC:REL:STAT ON	
	Linear	Log	Linear	Log
Single	Watt	dBm	%	dB
	%	dB	%	dB
Ratio	%	dB	%	dB
Difference	Watt	dBm	%	dB
	%	dB	%	dB

### Syntax



## Parameter

Item	Description/Default	Range of values
numeric_value	A numeric value for the upper test limit <ul style="list-style-type: none"> <li>• DEF: The default is –90.00 dBm or –90 dB</li> <li>• MIN: –150 dBm or –180 dB</li> <li>• MAX: +230 dBm or +200 dB</li> </ul>	–150 to +230 dBm or –180 to +200 dB  DEF  MIN  MAX

## Example

`CALC2:LIM:UPP 5`      *This command enters an upper limit for the lower window/upper measurement based on the window units as follows:*

*dBm = 5 dBm*  
*W = 5 W*  
*dB = 5 dB*  
*% = 5%*

## Reset condition

On reset, both measurements of the upper and lower windows are set to +90.00 dBm or +90 dB.

## Query

`CALCulate[1]|2|3|4:LIMit:UPPer[:DATA]? [MIN|MAX]`

This query returns the current setting of the upper limit or the values associated with MIN and MAX for the specified window/measurement.

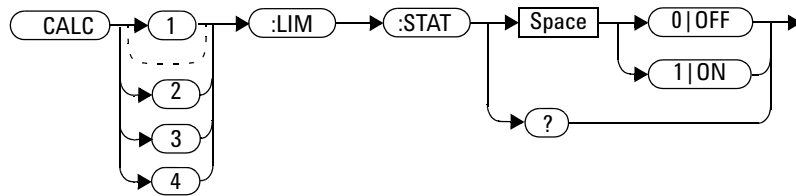
## Query example

`CALC2:LIM:UPP?`      *Queries the upper limit setting of the lower window/upper measurement.*

## CALCulate[1]|2|3|4:LIMit:STATe <boolean>

This command enables or disables the test limits for the specified window/measurement.

### Syntax



### Example

```
CALC:LIM:STAT 1
```

*This command enables the limit- checking function for the upper window/upper measurement.*

### Reset condition

On reset, limit checking is disabled.

### Query

```
CALCulate[1]|2|3|4:LIMit:STATe?
```

This query enters 1 or 0 into the output buffer indicating the status of the limit-checking function for the specified window/measurement.

- 1 is returned if limit checking is enabled.
- 0 is returned if limit checking is disabled.

### Query example

```
CALC:LIM:STAT?
```

*Queries whether limit checking for the upper window/upper measurement is enabled or disabled.*



## CALCulate[1]|2|3|4:MATH Commands

These commands configure and perform the following mathematical transformations on SENSE data:

- Single
- Difference
- Ratio

The following commands are detailed in this section:

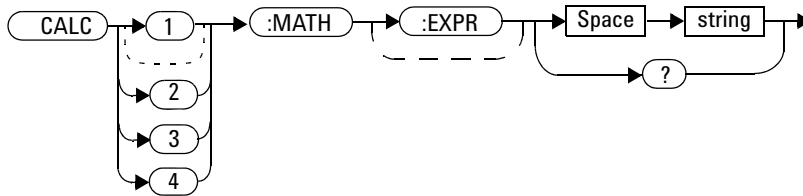
```
CALCulate[1]|2|3|4:MATH[:EXPRession] <string>
```

```
CALCulate[1]|2|3|4:MATH[:EXPRession]:CATalog?
```

# CALCulate[1] | 2 | 3 | 4:MATH[:EXPRession] <string>

This command sets the specified window/measurement to a single, difference, or ratio measurement.

## Syntax



## Parameter

Item	Description/Default	Range of values
string	A single string value specifying the measurement type. The default value is SENS1.	" (SENS1) "[1] " (SENS1-SENS1) "[1],[2] " (SENS1/SENS1) "[1]

[1] Quotes are mandatory. Either single or double quotes may be used.

[2] The mathematical operation will be performed in the linear scale.

## Example

```
CALC2:MATH " (SENS1/SENS1) "
```

*This command sets the lower window/upper measurement to make a Channel A/A ratio measurement.*

## Reset condition

On reset, the upper and lower window measurements are set to "(SENS1)".

## Query

```
CALCulate[1] | 2 | 3 | 4 :MATH[:EXPRession]?
```

This query returns the current math measurement setting of the specified window/measurement.

## Query example

```
CALC:MATH?
```

*Queries the current setting of the math expression for the upper window/upper measurement.*

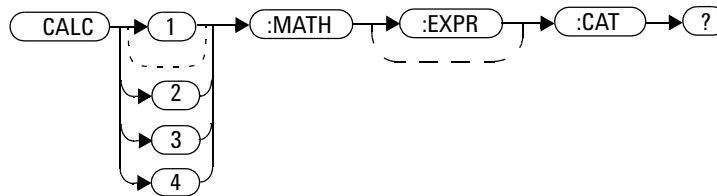
## CALCulate[1|2|3|4:MATH[:EXPRession]:CATalog?]

This query lists all the defined math expressions in the form of comma-separated strings as follows:

" (SENS1) " , " (SENS1/SENS1) " , " (SENS1-SENS1) "

Each string contains a math expression.

### Syntax



### Example

CALC:MATH:CAT? *Lists all the defined math expressions.*

## CALCulate[1]|2|3|4:RELative Commands

These commands enable comparison of a measurement result to a reference value.

Within the CALCulate block, the relative value is applied to the measurement signal after any math calculations and display offsets have been included.

The commands are described in this section:

```
CALCulate[1]|2|3|4:RELative[:MAGNitude]:AUTO <boolean>|ONCE
```

```
CALCulate[1]|2|3|4:RELative:STATE <boolean>
```

# CALCulate[1|2|3|4:RELative[:MAGNitude]:AUTO <boolean> | ONCE

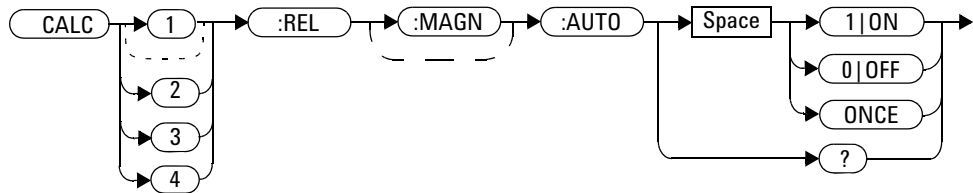
This command sets the reference value to be used in the relative measurement for the specified window/measurement.

The parameter should be set to ONCE to set the reference value to be used in the relative measurement. Specifying ONCE sets the reference value to that of the measurement signal after any math calculations and display offsets have been applied. After the reference value has been set, the command returns to OFF. Setting this command parameter to ONCE turns ON the CALCulate[1|2|3|4:RELative:STATE command.

If 0|OFF is specified, no reference value is applied to the measurement signal. There is no situation in which you would want to send this command with OFF. OFF is only available because it is required for the query response.

If 1|ON is selected, it causes error -224, "Illegal parameter value" to occur.

## Syntax



## Example

```
CALC:REL:AUTO ONCE
```

*This command sets a reference value to be used in the relative measurement on the upper window/upper measurement.*

## Query

```
CALCulate[1] | 2 | 3 | 4:RELative[:MAGNitude]:AUTO?
```

This query always returns OFF.

## Error message

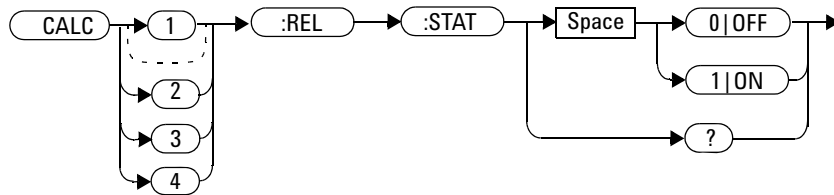
If the command parameter is set to ON, error -224, “Illegal parameter value” occurs.

## CALCulate[1|2|3|4]:RELative:STATe <boolean>

This command enables or disables the relative mode for the specified window/measurement. If the command is:

- disabled, the measurement signal remains unchanged.
- enabled, the current reference value set by CALCulate:RELative:MAGnitude:AUTO is applied to the measurement signal.

### Syntax



### Example

CALC3:REL:STAT OFF

*This command disables the relative mode for the upper window/lower measurement.*

### Reset condition

On reset, relative mode is disabled.



## Query

```
CALCulate[1] | 2 | 3 | 4:RELative:STATe?
```

This query enters a 1 or 0 into the output buffer indicating the status of the relative mode.

- 1 is returned if the relative mode is enabled.
- 0 is returned if the relative mode is disabled.

## Query example

```
CALC:REL:STAT? Queries whether the relative mode is enabled or disabled for the upper window/upper measurement.
```





## 5 CALibration Subsystem

CALibration Subsystem 204

CALibration[1]:RCFactor <numeric\_value> 205

CALibration[1]:ZERO:AUTO <boolean>|ONCE 207

This chapter explains how the CALibration command subsystem is used to perform zeroing and configure the reference calibration factor on the N432A.



## CALibration Subsystem

CALibration commands are used to perform zeroing on the N432A as well as apply the reference calibration factor for the thermistor sensor being used.

The numeric suffix of the CALibration command (CALibration1) represents channel A.

Zeroing of the N432A is recommended:

- when a 5°C change in temperature occurs.
- when connection to the thermistor sensor is established.
- when you change the thermistor sensor.
- every 24 hours.
- prior to measuring low-level signals.

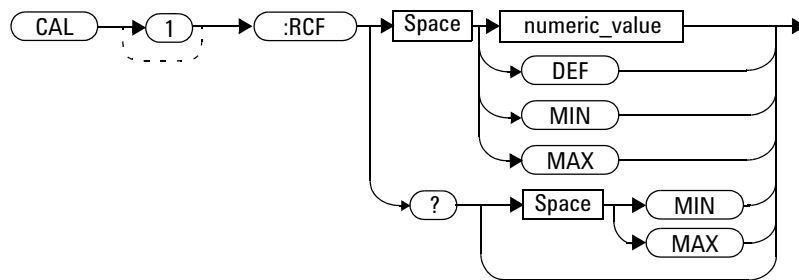
The CAL:ZERO:AUTO command is an overlapped command, where it allows the N432A to continue parsing and executing subsequent commands while it is still executing.

Keyword	Parameter form	Note	Page
CALibration[1]			
:RCFactor	<numeric_value>		<a href="#">page 205</a>
:ZERO			
:AUTO	<boolean> ONCE		<a href="#">page 207</a>

## CALibration[1]:RCFactor <numeric\_value>

This command is used to set the value for the reference calibration factor. The reference calibration factor can also be set from the sensor calibration table. The N432A uses the most recently set reference calibration factor.

### Syntax



### Parameter

Item	Description/Default	Range of values
numeric_value	A numeric value for the reference calibration factor <ul style="list-style-type: none"> <li>• DEF: The default value is 100%</li> <li>• MIN: 1%</li> <li>• MAX: 150%</li> </ul>	1.0 to 150.0 PCT DEF MIN MAX

### Example

CAL:RCF 98 *This command enters a reference calibration factor of 98%.*

## Reset condition

On reset, the reference calibration factor is set to 100%.

## Query

```
CALibration[1]:RCFactor? [MIN|MAX]
```

This query returns the current setting of the reference calibration factor or the values associated with MIN and MAX.

## Query example

```
CAL:RCF?      Queries the current reference calibration factor setting.
```

## CALibration[1]:ZERO:AUTO <boolean> | ONCE

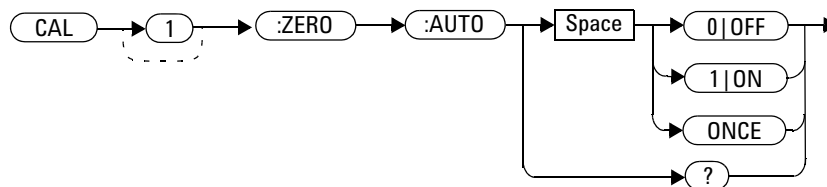
This command causes the N432A to perform its zeroing routine when enabled. This adjusts the N432A for a zero power reading with no power supplied to the thermistor sensor.

1|ON is ignored in this command. Only ONCE can be specified to initiate zeroing on the N432A.

0|OFF is only required for the query response and is ignored in this command.

This command assumes that the sensor is not connected to any power source.

### Syntax



### Example

```
CAL:ZERO:AUTO ONCE
```

*This command causes the N432A to perform a zeroing routine.*

### Reset condition

On reset, automatic zeroing is disabled.

## Query

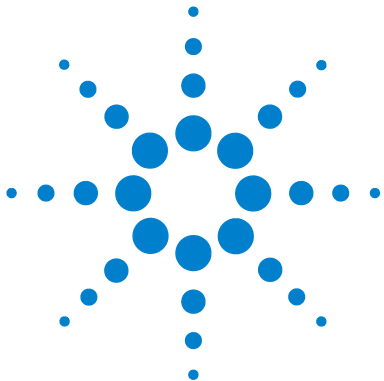
```
CALibration[1]:ZERO:AUTO?
```

This query always returns a value of 0.

## Error messages

- If CAL:ZERO:AUTO is set to ON and an incompatible sensor is connected, error -241, “Hardware missing” occurs.
- If zeroing was not carried out successfully, error -231, “Data Questionable; ZERO ERROR” occurs.
- If there is no sensor connected, error -241, “Hardware missing” occurs.





## 6 DISPlay Subsystem

DISPlay Subsystem	210
DISPlay:ENABle <boolean>	211
DISPlay:SCReen:FORMat <character_data>	212
DISPlay[:WINDow[1] 2] Commands	214
DISPlay[:WINDow[1] 2]:ANALog Commands	215
DISPlay[:WINDow[1] 2]:ANALog:LOWer <numeric_value>	216
DISPlay[:WINDow[1] 2]:ANALog:UPPer <numeric_value>	218
DISPlay[:WINDow[1] 2]:FORMat <character_data>	220
DISPlay[:WINDow[1] 2]:METer Commands	222
DISPlay[:WINDow[1] 2]:METer:LOWer <numeric_value>	223
DISPlay[:WINDow[1] 2]:METer:UPPer <numeric_value>	225
DISPlay[:WINDow[1] 2]:NUMeric[1] 2:RESolution <numeric_value>	227
DISPlay[:WINDow[1] 2]:SELEct[1] 2	229
DISPlay[:WINDow[1] 2][::STATe] <boolean>	231

This chapter explains how the DISPlay subsystem is used to control the selection and presentation of the windows on the N432A display.



## DISPlay Subsystem

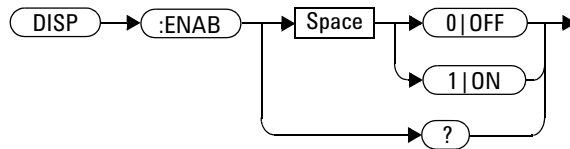
The DISPlay subsystem enables you to configure the windows used on the N432A display.

Keyword	Parameter form	Note	Page
DISPlay			
:ENABle	<boolean>		<a href="#">page 211</a>
:SCREen			
:FORMat	<character_data>		<a href="#">page 212</a>
[:WINDow[1]   2]			
:ANALog			
:LOWer	<numeric_value>		<a href="#">page 216</a>
:UPPer	<numeric_value>		<a href="#">page 218</a>
:FORMat	<character_data>		<a href="#">page 220</a>
:METer			
:LOWer	<numeric_value>		<a href="#">page 223</a>
:UPPer	<numeric_value>		<a href="#">page 225</a>
:NUMeric[1]   2			
:RESolution	<numeric_value>		<a href="#">page 227</a>
:SELect[1]   2			<a href="#">page 229</a>
[:STATE]	<boolean>		<a href="#">page 231</a>

## DISPlay:ENABle <boolean>

This command is used to enable and disable the display. At power up, the display is always enabled.

### Syntax



### Example

DISP:ENAB 0      *This command disables the display.*

### Reset condition

On reset, the display is enabled.

### Query

DISPlay:ENABle?

This query enters a 1 or 0 into the output buffer indicating whether the display is enabled or disabled.

- 1 is returned if the display is enabled
- 0 is returned if the display is disabled

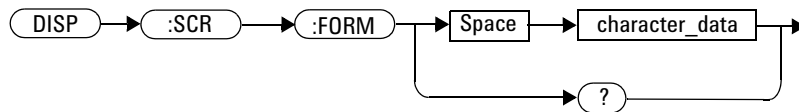
### Query example

DISP:ENAB?      *Queries whether the display is turned on or off.*

## DISPlay:SCReen:FORMat <character\_data>

This command sets the display format.

### Syntax



### Parameter

Item	Description/Default	Range of values
character_data	Sets the display format: <ul style="list-style-type: none"> <li>• <b>WINDowed</b>: The windowed format provides two display windows. Each window can display two measurements.</li> <li>• <b>EXPanded</b>: The expanded format provides one display window which displays a single measurement. This display format provides access to softkeys.</li> <li>• <b>FSCReen</b>: The full screen format provides one display window which displays a single measurement. This display format does not provide access to softkeys.</li> </ul>	WIND EXP FSCR

### Example

DISP:SCR:FORM FSCR

*This command sets the display format to full screen.*

### Reset condition

On reset, the display format is WINDowed.

## Query

DISPlay:SCReen:FORMat?

This query returns the current display format of either WIND, EXP, or FSCR.

## Query example

DISP:SCR:FORM?     *Queries the display format.*

## **DISPlay[:WINDow[1]|2] Commands**

These commands control various characteristics of the display windows. WINDow1 and WINDow2 represent the upper window and lower window respectively.

The following commands are detailed in this section:

```
DISPlay[:WINDow[1]|2]:ANALog:LOWer <numeric_value>
```

```
DISPlay[:WINDow[1]|2]:ANALog:UPPer <numeric_value>
```

```
DISPlay[:WINDow[1]|2]:FORMat <character_data>
```

```
DISPlay[:WINDow[1]|2]:METer:LOWer <numeric_value>
```

```
DISPlay[:WINDow[1]|2]:METer:UPPer <numeric_value>
```

```
DISPlay[:WINDow[1]|2]:NUMeric[1]|2:RESolution <numeric_value>
```

```
DISPlay[:WINDow[1]|2]:SELEct[1]|2
```

```
DISPlay[:WINDow[1]|2][:STATe] <boolean>
```

## DISPlay[:WINDow[1]|2]:ANALog Commands

These commands control the upper and lower scale limits of the analog meter.

The following commands are detailed in this section:

```
DISPlay[:WINDow[1]|2]:ANALog:LOWer <numeric_value>
```

```
DISPlay[:WINDow[1]|2]:ANALog:UPPer <numeric_value>
```

## DISPlay[:WINDow[1] | 2]:ANALog:LOWer <numeric\_value>

This command sets the analog meter lower scale limit.

### NOTE

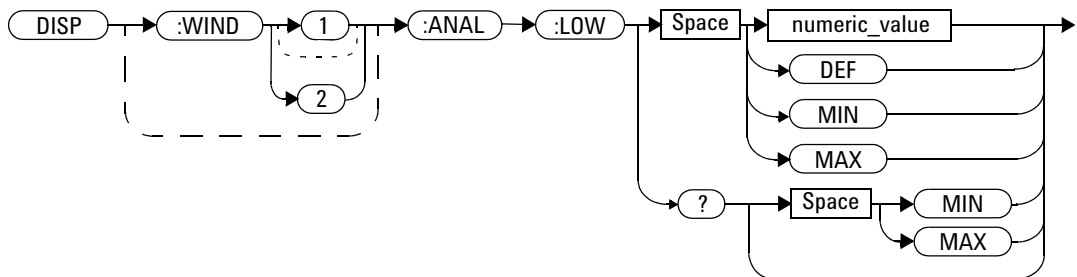
This command has the same purpose as `DISPlay[:WINDow[1] | 2]:METer:LOWer <numeric_value>`.

The units used are dependent on the current settings of `UNIT:POWer` and `CALCulate:RELative:STATE` as shown in the following table.

**Table 6-1** Measurement units

Measurement type	Measurement mode	CALC:REL:STAT OFF		CALC:REL:STAT ON	
		Linear	Log	Linear	Log
<b>Single</b>	$P_{avg}$	Watt	dBm	%	dB
<b>Ratio</b>	$P_{avg}$	%	dB	%	dB
<b>Difference</b>	$P_{avg}$	Watt	dBm	%	dB

### Syntax





## Parameter

Item	Description/Default	Range of values
numeric_value	<p>A numeric value for the analog meter lower scale limit:</p> <ul style="list-style-type: none"> <li>• DEF: The default value is -70 dBm</li> <li>• MIN: -150 dBm</li> <li>• MAX: 230 dBm</li> </ul> <p>The units used are determined by the current settings of UNIT:POWer and CALCulate:RELative:STATE as shown in <a href="#">Table 6-1</a>.</p>	<p>-150 to 230 dBm</p> <p>DEF</p> <p>MIN</p> <p>MAX</p>

## Example

DISP:WIND1:ANAL:LOW -50      *This command sets the analog meter lower scale limit of the upper window to -50 dBm.*

## Reset condition

On reset, the value is set to -70 dBm for both the upper and lower windows.

## Query

DISPlay[:WINDow[1]|2]:ANALog:LOWer? [MIN|MAX]

This query returns the current setting of the analog meter lower scale limit, or the value associated with MIN or MAX. The response format is <NR3>. The units in which the results are returned are determined by the current settings of UNIT:POWer and CALCulate:RELative:STATE as shown in [Table 6-1](#).

## Query example

DISP:WIND1:ANAL:LOW?      *This command queries the lower scale limit set on the analog meter in the upper window.*

## DISPlay[:WINDow[1] | 2]:ANALog:UPPer <numeric\_value>

This command sets the analog meter upper scale limit.

### NOTE

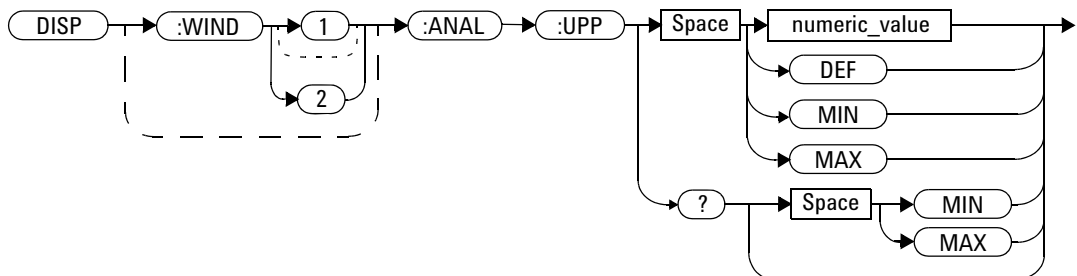
This command has the same purpose as `DISPlay[:WINDow[1] | 2]:METer:UPPer <numeric_value>`.

The units used are dependent on the current settings of `UNIT:POWer` and `CALCulate:RELative:STATe` as shown in the following table.

**Table 6-2** Measurement units

Measurement type	Measurement mode	CALC:REL:STAT OFF		CALC:REL:STAT ON	
		Linear	Log	Linear	Log
<b>Single</b>	$P_{avg}$	Watt	dBm	%	dB
<b>Ratio</b>	$P_{avg}$	%	dB	%	dB
<b>Difference</b>	$P_{avg}$	Watt	dBm	%	dB

### Syntax



## Parameter

Item	Description/Default	Range of values
numeric_value	<p>A numeric value for the analog meter upper scale limit:</p> <ul style="list-style-type: none"> <li>• DEF: The default value is 20 dBm</li> <li>• MIN: -150 dBm</li> <li>• MAX: 230 dBm</li> </ul> <p>The units used are determined by the current settings of UNIT:POWER and CALCulate:RELative:STATE as shown in <a href="#">Table 6-2</a>.</p>	<p>-150 to 230 dBm</p> <p>DEF</p> <p>MIN</p> <p>MAX</p>

## Example

DISP:WIND2:ANAL:UPP 50 *This command sets the analog meter upper scale limit of the lower window to 50 dBm.*

## Reset condition

On reset, the upper scale limit is set to 20 dBm.

## Query

DISPlay: [WINDow[1] | 2]:ANALog:UPPer? [MIN|MAX]

This query returns the current setting of the analog meter upper scale limit, or the value associated with MIN or MAX. The response format is <NR3>. The units in which the results are returned are determined by the current settings of UNIT:POWER and CALCulate:RELative:STATE as shown in [Table 6-2](#).

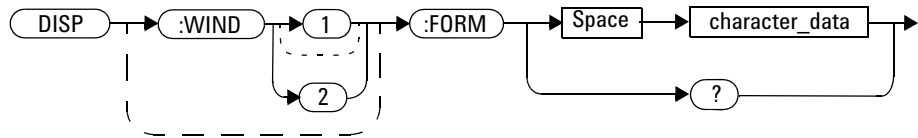
## Query example

DISP:WIND2:ANAL:UPP? *Queries the upper scale limit set on the analog meter in the lower window.*

## DISPlay[:WINDow[1] | 2]:FORMat <character\_data>

This command selects the display format of the selected window.

### Syntax



### Parameter

Item	Description/Default	Range of values
character_data	Sets the window display format: <ul style="list-style-type: none"> <li>• DIGital: Sets the window to a digital display. This setting is the same as SNUMeric.</li> <li>• ANALog: Sets the window to an analog display using the currently selected measurement.</li> <li>• SNUMeric: Sets the window to a single numeric display. The currently selected measurement is displayed. This setting is the same as DIGital.</li> <li>• DNUMeric: Sets the window to a dual numeric display.</li> </ul>	DIGital ANALog SNUMeric DNUMeric

### Example

```
DISP:WIND2:FORM DIG
```

*This command sets the lower window to a digital display.*

## Reset condition

On reset, the upper window is set to a digital display and the lower window is set to an analog display.

## Query

```
DISPlay[:WINDow[1]|2]:FORMat?
```

This query returns the current display format of the selected window.

## Query example

```
DISP:FORM?      Queries the current display format of the upper window.
```

## **DISPlay[:WINDow[1] | 2]:METer Commands**

These commands control the upper and lower scale limits of the analog meter.

The following commands are detailed in this section:

```
DISPlay[:WINDow[1] | 2]:METer:LOWer <numeric_value>
```

```
DISPlay[:WINDow[1] | 2]:METer:UPPer <numeric_value>
```

## DISPlay[:WINDow[1] | 2]:METer:LOWer <numeric\_value>

This command sets the lower scale limit of the analog meter.

### NOTE

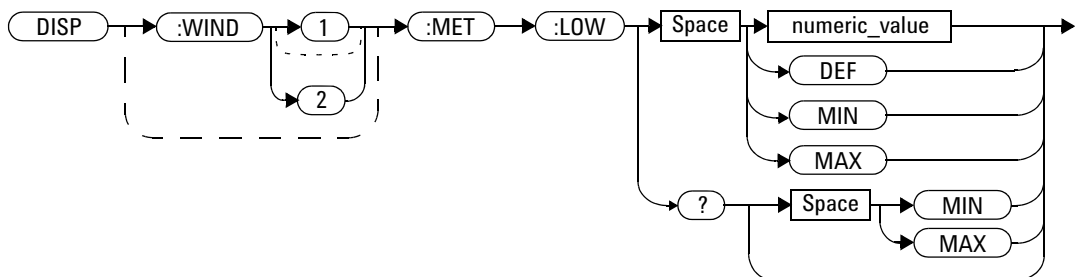
This command has the same purpose as `DISPlay[:WINDow[1] | 2]:ANALog:LOWer <numeric_value>`.

The units used are dependent on the current settings of `UNIT:POWer` and `CALCulate:RELative:STATe` as shown in the following table.

**Table 6-3** Measurement units

Measurement type	Measurement mode	CALC:REL:STAT OFF		CALC:REL:STAT ON	
		Linear	Log	Linear	Log
Single	$P_{avg}$	Watt	dBm	%	dB
Ratio	$P_{avg}$	%	dB	%	dB
Difference	$P_{avg}$	Watt	dBm	%	dB

### Syntax



## Parameter

Item	Description/Default	Range of values
numeric_value	<p>A numeric value for the analog meter lower scale limit:</p> <ul style="list-style-type: none"> <li>DEF: The default value is <math>-70</math> dBm</li> <li>MIN: <math>-150</math> dBm</li> <li>MAX: <math>230</math> dBm</li> </ul> <p>The units used are determined by the current settings of UNIT:POWer and CALCulate:RELative:STATE as shown in <a href="#">Table 6-3</a>.</p>	<p><math>-150</math> to <math>230</math> dBm</p> <p>DEF</p> <p>MIN</p> <p>MAX</p>

## Example

DISP:WIND2:MET:LOW 10 *This command sets the analog meter lower scale limit of the lower window to 10 dBm.*

## Reset condition

On reset, the lower scale limit is set to  $-70$  dBm.

## Query

DISPlay[:WINDow[1]|2]:METer:LOWer? [MIN|MAX]

This query returns the current setting of the analog meter lower scale limit or the value associated with MIN or MAX. The response format is <NR3>. The units in which the results are returned are dependent on the current settings of UNIT:POWer and CALCulate:RELative:STATE as shown in [Table 6-3](#).

## Query example

DISP:MET:LOW? *Queries the lower scale limit set on the analog meter in the upper window.*



## DISPlay[:WINDow[1] | 2]:METer:UPPer <numeric\_value>

This command sets the upper scale limit of the analog meter.

### NOTE

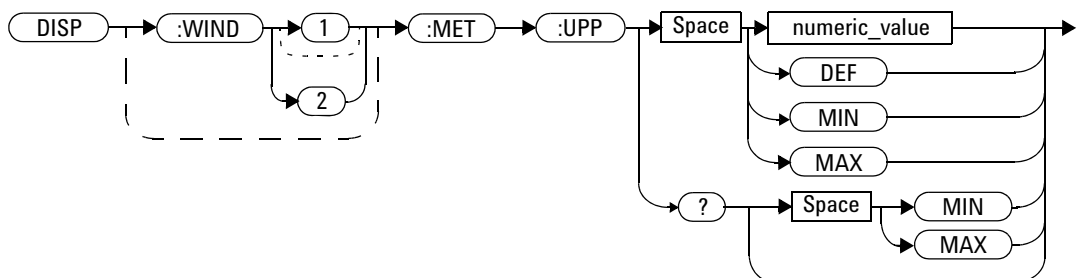
This command has the same purpose as DISPlay[:WINDow[1] | 2]:ANALog:UPPer <numeric\_value>.

The units used are dependent on the current settings of UNIT:POWer and CALCulate:RELative:STATe as shown in the following table.

**Table 6-4** Measurement units

Measurement type	Measurement mode	CALC:REL:STAT OFF		CALC:REL:STAT ON	
		Linear	Log	Linear	Log
Single	P <sub>avg</sub>	Watt	dBm	%	dB
Ratio	P <sub>avg</sub>	%	dB	%	dB
Difference	P <sub>avg</sub>	Watt	dBm	%	dB

### Syntax



## Parameter

Item	Description/Default	Range of values
numeric_value	<p>A numeric value for the analog meter upper scale limit:</p> <ul style="list-style-type: none"> <li>• DEF: The default value is 20 dBm</li> <li>• MIN: -150 dBm</li> <li>• MAX: 230 dBm</li> </ul> <p>The units used are determined by the current settings of UNIT:POWer and CALCulate:RELative:STATE as shown in <a href="#">Table 6-4</a>.</p>	<p>-150 to 230 dBm</p> <p>DEF</p> <p>MIN</p> <p>MAX</p>

## Example

DISP:WIND2:MET:UPP 20      *This command sets the analog meter upper scale limit of the lower window to 20 dBm.*

## Reset condition

On reset, the upper scale limit is set to 20 dBm.

## Query

DISPlay[:WINDow[1]|2]:METer:UPPer? [MIN|MAX]

This query returns the current setting of the analog meter upper scale limit or the value associated with MIN or MAX. The response format is <NR3>. The units in which the results are returned are dependent on the current settings of UNIT:POWer and CALCulate:RELative:STATE as shown in [Table 6-4](#).

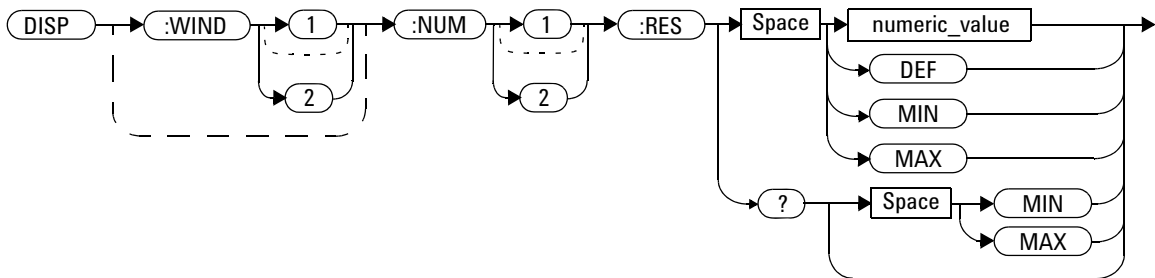
## Query example

DISP:WIND2:MET:UPP?      *Queries the upper scale limit set on the analog meter in the lower window.*

## DISPlay[:WINDow[1] | 2]:NUMeric[1] | 2:RESolution <numeric\_value>

This command sets the resolution of the measurement result in the specified window and measurement line.

### Syntax



### Parameter

Item	Description/Default	Range of values
numeric_value	A numeric value for the window resolution: <ul style="list-style-type: none"> <li>• DEF: 3</li> <li>• MIN: 1</li> <li>• MAX: 4</li> </ul>	1 to 4 DEF MIN MAX

### Example

DISP:WIND2:NUM1:RES 4 *This command sets the lower window/upper measurement resolution to four significant digits if the measurement result is linear, or to 0.001 if the measurement result is logarithmic.*

## Reset condition

On reset, the resolution is set to 3.

## Query

```
DISPlay[:WINDow[1]|2]:NUMeric[1]|2:RESolution? [MIN|MAX]
```

This query returns the current setting of the window/measurement resolution or the value associated with MIN or MAX. The response format is <NR1>.

## Query example

```
DISP:WIND:NUM2:RES?
```

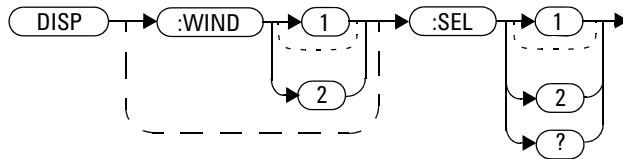
*Queries the resolution setting of the upper window/lower measurement.*

## DISPlay[:WINDow[1] | 2]:SElect[1] | 2

This command is used to select a specific measurement within a specific window.

If the second numeric value is not specified, the upper measurement of the relevant window is selected. This command specifies which measurement is to be used for the analog or single numeric display.

### Syntax



### Example

DISP:WIND2:SEL1

*This command selects the upper measurement of the lower window.*

### Reset condition

On reset, the upper measurement of the upper window is selected.

## Query

```
DISPlay[:WINDow[1]|2]:SElect[1]|2?
```

This query enters a 1 or 0 into the output buffer indicating whether the specified measurement of the specified window is currently selected.

- 1 is returned if the specified measurement of the specified window is selected
- 0 is returned if the specified measurement of the specified window is not selected

## Query example

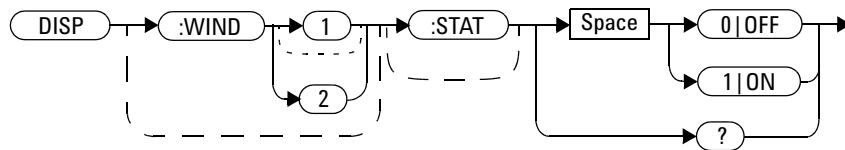
```
DISP:SEL1?
```

*Queries whether the upper measurement of the upper window is selected.*

## DISPlay[:WINDow[1] | 2][:STATe] <boolean>

This command enables or disables the upper or lower window (WINDow1 and WINDow2 respectively) so that the display shows a single window only. The displayed window is presented in the expanded format, showing only a single measurement; either the single measurement that was shown on the window or the currently selected measurement, if two measurements had been shown.

### Syntax



### Examples

DISP:WIND2:STAT OFF

*This command disables the lower window. The upper window is shown in the expanded format, displaying its currently selected measurement.*

DISP:WIND2:STAT ON

*This command enables the lower window so that a dual window display is once more provided.*

### Reset condition

On reset, both the upper and lower windows are enabled.

## Query

```
DISP:WIND[1|2]:STAT?
```

This query enters a 1 or 0 in the output buffer indicating the state of the selected window.

- 1 is returned if the window is enabled.
- 0 is returned if the window is disabled.

## Query example

```
DISP:WIND2:STAT?
```

*Queries whether the lower window is displayed.*





## 7 FORMat Subsystem

FORMat Subsystem [234](#)

FORMat[:READings]:BORDER <character\_data> [235](#)

FORMat[:READings][:DATA] <character\_data> [237](#)

This chapter explains how the FORMat subsystem is used to set the data format for transferring numeric information.



## FORMat Subsystem

The FORMat subsystem sets the data format for transferring numeric information. This data format is used only for response data by the queries that are affected by the FORMat subsystem.

The queries affected are:

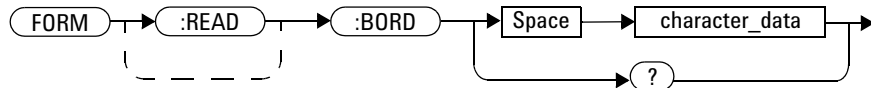
- FETCh?
- READ?
- MEASure?

Keyword	Parameter form	Note	Page
FORMat			
[:READings]			
:BORDer	<character_data>		<a href="#">page 235</a>
[:DATA]	<character_data>		<a href="#">page 237</a>

## FORMat[:READings]:BORDER <character\_data>

This command determines whether the binary data is transferred in the normal or swapped byte order. It is only used when FORMat[:READings][:DATA] is set to REAL.

### Syntax



### Parameter

Item	Description/Default	Range of values
character_data	Byte order of binary data transfer: <ul style="list-style-type: none"> <li>• NORMAl</li> <li>• SWAPped</li> </ul>	NORMAl SWAPped

### Example

FORM:BORDER SWAP      *This command sets the swapped byte order of the binary data transfer.*

### Reset condition

On reset, the byte order is set to NORMAl.

## Query

```
FORMat[:READings]:BORDER?
```

This query returns the current setting of the byte order in the form of NORMAl or SWAPped.

## Query example

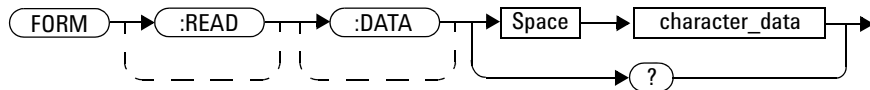
```
FORM:BORD?      Queries the current byte order setting.
```

## FORMat[:READings][:DATA] <character\_data>

This command sets the data format for transferring numeric information to either ASCII or REAL:

- When the format type is ASCII, numeric data is output as ASCII bytes in the <NR3> format.
- When the format type is REAL, numeric data is output as IEEE-754 64-bit floating point numbers in a definite-length block. The result is an 8-byte block per number. Each complete block is terminated by a line feed character.

### Syntax



### Parameter

Item	Description/Default	Range of values
character_data	Data format for transferring data: <ul style="list-style-type: none"> <li>• ASCII</li> <li>• REAL</li> </ul>	ASCII REAL

### Example

FORM REAL      *This command sets the data format to REAL.*

### Reset condition

On reset, the data format is set to ASCII.

## Query

```
FORMat [:READings] [:DATA] ?
```

This query returns the current setting of data format in the form of ASCII or REAL.

## Query example

```
FORM?           Queries the current data format setting.
```



## 8 HCOPy Subsystem

HCOPy Subsystem [240](#)

HCOPy:SDUMp:DATA? [241](#)

HCOPy:SDUMp:DATA:FORMat BMP|PNG [242](#)

This chapter explains how the HCOPy subsystem is used to output the screen image and select the image file format.



## HCOpy Subsystem

The HCOpy subsystem is used to output the N432A front panel display as an image file and to select the file format.

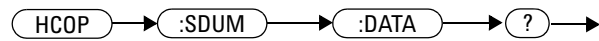
Keyword	Parameter form	Note	Page
HCOpy			
:SDUMp			
:DATA?		[query only]	<a href="#">page 241</a>
:DATA			
:FORMat	BMP   PNG		<a href="#">page 242</a>



## HCOPy:SDUMp:DATA?

This query returns the current screen image from the N432A as a file. The image file is in a format specified by the HCOPy:SDUMp:DATA:FORMat command.

### Syntax



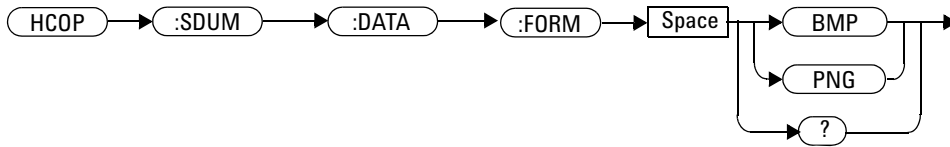
### Example

HCOP:SDUM:DATA?     *Outputs the current screen image.*

## HCOpy:SDUMp:DATA:FORMat BMP | PNG

This command is used to set the format of the screen image file returned by the HCOpy:SDUMp:DATA? query.

### Syntax



### Parameter

Item	Description/Default	Range of values
format	Available image format: <ul style="list-style-type: none"> <li>• Bitmap: BMP</li> <li>• Portable Network Graphics: PNG</li> </ul>	BMP PNG

### Example

HCOpy:SDUM:DATA:FORM BMP

*This command sets the screen image file format to bitmap.*

### Reset condition

On reset, the file format is PNG.

## Query

```
HCOPy:SDUMp:DATA:FORMat?
```

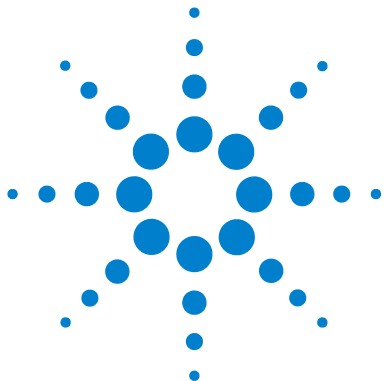
This query returns the current format of the screen image file in the form of BMP or PNG.

## Query example

```
HCOP:SDUM:DATA:FORM?
```

*Queries the current screen image file format.*





## 9 LXI Subsystem

LXI Subsystem [246](#)

LXI:IDENTify[:STATe] <boolean> [247](#)

This chapter describes how the LXI subsystem is used to configure the LXI identification state.



## LXI Subsystem

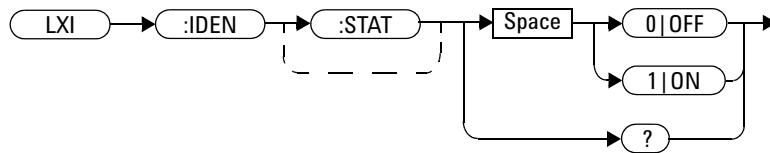
The LXI subsystem is used to enable or disable LXI identification.

Keyword	Parameter form	Note	Page
LXI			
:IDENTify			
[:STATE]	<boolean>		<a href="#">page 247</a>

## LXI:IDENTify[:STATe] <boolean>

This command enables or disables LXI identification.

### Syntax



### Example

`LXI:IDEN 1`      *This command enables LXI identification.*

### Reset condition

On reset, LXI identification is disabled.

### Query

`LXI:IDENTify[:STATe]?`

This query enters a 1 or 0 into the output buffer indicating the status of the LXI identification:

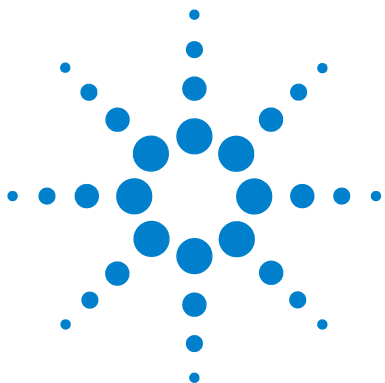
- 1 is returned if LXI identification is enabled.
- 0 is returned if LXI identification is disabled.

### Query example

`LXI:IDEN?`      *Queries whether LXI identification is enabled or disabled.*







## 10 MEMory Subsystem

MEMory Subsystem	250
MEMory[:METer][1]:CATalog Queries	252
MEMory[:METer][1]:CATalog[:ALL]?	253
MEMory[:METer][1]:CATalog:STATe?	255
MEMory[:METer][1]:CATalog:TABLE?	256
MEMory[:METer][1]:CLEar Commands	258
MEMory[:METer][1]:CLEar[:NAME] <character_data>	259
MEMory[:METer][1]:CLEar:TABLE	260
MEMory[:METer][1]:FREE Queries	261
MEMory[:METer][1]:FREE[:ALL]?	262
MEMory[:METer][1]:FREE:STATe?	263
MEMory[:METer][1]:FREE:TABLE?	264
MEMory[:METer][1]:NSTates?	265
MEMory[:METer][1]:STATe Commands	266
MEMory[:METer][1]:STATe:CATalog?	267
MEMory[:METer][1]:STATe:DEFine <character_data>,<numeric_value>	268
MEMory[:METer][1]:TABLE Commands	270
MEMory[:METer][1]:TABLE:FREQuency <numeric_value>{,<numeric_value>}	271
MEMory[:METer][1]:TABLE:FREQuency:POINts?	274
MEMory[:METer][1]:TABLE:GAIN[:MAGNitude] <numeric_value>{,<numeric_value>}	275
MEMory[:METer][1]:TABLE:GAIN[:MAGNitude]:POINts?	277
MEMory[:METer][1]:TABLE:MOVE <character_data>,<character_data>	278
MEMory[:METer][1]:TABLE:SElect <character_data>	279

This chapter explains how the MEMory command subsystem is used to configure sensor tables and save/recall registers.



## MEMory Subsystem

The MEMory command subsystem is used to:

- Edit, review, and store sensor calibration tables.
- Edit, review, and store frequency-dependent offset tables.
- Edit and review save/recall registers

Stored tables remain in the N432A memory during power down. The N432A is capable of storing 20 sensor calibration tables and 10 frequency-dependent offset tables of 80 frequency points each.

Keyword	Parameter form	Note	Page
MEMory			
[ :METer ] [ 1 ]			
:CATalog			
[ :ALL ] ?		[query only]	<a href="#">page 253</a>
:STATe?		[query only]	<a href="#">page 255</a>
:TABLE?		[query only]	<a href="#">page 256</a>
:CLEar			
[ :NAME ]	<character_data>	[no query]	<a href="#">page 259</a>
:TABLE		[no query]	<a href="#">page 260</a>
:FREE			
[ :ALL ] ?		[query only]	<a href="#">page 262</a>
:STATe?		[query only]	<a href="#">page 263</a>
:TABLE?		[query only]	<a href="#">page 264</a>
:NSTates?		[query only]	<a href="#">page 265</a>
:STATe			
:CATalog?		[query only]	<a href="#">page 267</a>
:DEFine	<character_data> [ , <numeric_value> ]		<a href="#">page 268</a>
:TABLE			
:FREQuency	<numeric_value> [ , <numeric_value> ]		<a href="#">page 271</a>

Keyword	Parameter form	Note	Page
	:POINTs?	[query only]	<a href="#">page 274</a>
:GAIN	[ :MAGNitude]		<a href="#">page 275</a>
	<numeric_value> [, <numeric_value>]		
	:POINTs?	[query only]	<a href="#">page 277</a>
:MOVE	<character_data>, <character_data>	[no query]	<a href="#">page 278</a>
:SElect	<character_data>	[no query]	<a href="#">page 279</a>

## MEMory[:METer][1]:CATalog Queries

These queries are used to retrieve information on the current contents of the N432A as follows:

- Sensor calibration tables
- Frequency-dependent offset tables
- Save/recall registers

The following queries are detailed in this section:

```
MEMory[:METer][1]:CATalog[:ALL]?
```

```
MEMory[:METer][1]:CATalog:STATE?
```

```
MEMory[:METer][1]:CATalog:TABLE?
```

## MEMory[:METer][1]:CATalog[:ALL]?

This query returns stored sensor calibration tables, frequency-dependent offset tables, and save/recall registers. The data is returned in the form of two numeric parameters and as many strings as there are stored tables and save/recall registers:

```
<numeric_value>,<numeric_value>{,<string>}
```

- The first numeric parameter indicates the amount of memory (in bytes) used for the storage of tables and registers.
- The second numeric parameter indicates the memory (in bytes) available for the storage of tables and registers.
- Each string parameter returned indicates the name, type, and size of a stored table or save/recall register:

```
<name>,<type>,<size>
```

- <name> indicates the name of the table or save/recall register.
- <type> indicates TABL for sensor calibration and frequency-dependent offset tables, or STAT for a save/recall register.
- <size> indicates the size of the table or save/recall register in bytes.

A sample response may look like the following:

```
198,4664,"DEFAULT,TABL,14","478A,TABL,86","8478B,TABL,98",
"CUSTOM_0,TABL,0","CUSTOM_1,TABL,0","CUSTOM_2,TABL,0","CUS
TOM_3,TABL,0","CUSTOM_4,TABL,0","CUSTOM_5,TABL,0","CUSTOM_
6,TABL,0","CUSTOM_7,TABL,0","CUSTOM_8,TABL,0","CUSTOM_9,TA
BL,0","CUSTOM_10,TABL,0","CUSTOM_11,TABL,0","CUSTOM_12,TAB
L,0","CUSTOM_13,TABL,0","CUSTOM_14,TABL,0","CUSTOM_15,TABL
,0","CUSTOM_16,TABL,0","CUSTOM_A,TABL,0","CUSTOM_B,TABL,0"
,"CUSTOM_C,TABL,0","CUSTOM_D,TABL,0","CUSTOM_E,TABL,0","CU
STOM_F,TABL,0","CUSTOM_G,TABL,0","CUSTOM_H,TABL,0","CUSTOM
_I,TABL,0","CUSTOM_J,TABL,0","State1,STAT,0","State2,STAT,
0","State3,STAT,0","State4,STAT,0","State5,STAT,0","State6
,STAT,0","State7,STAT,0","State8,STAT,0","State9,STAT,0","
State10,STAT,0"
```

The N432A is shipped with a set of predefined sensor calibration tables. The data in these tables is based on statistical averages for a range of Agilent thermistor sensors. These tables can be edited and are as listed below.

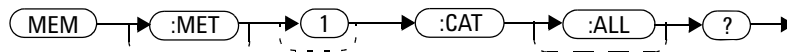
**Table 10-1** Predefined sensor calibration tables

Table	Thermistor sensor	Table name
0	None	DEFAULT
1	478A	478A
2	8478B	8478B

#### NOTE

There are also 17 sensor calibration tables named CUSTOM\_0 through CUSTOM\_16 and ten frequency-dependent offset tables named CUSTOM\_A through CUSTOM\_J which do not contain any data when the N432A is shipped from the factory.

## Syntax



## Example

MEM:CAT?

*Queries the list of tables and save/recall registers.*

## MEMory[:METer][1]:CATalog:STATe?

This query is used to list the save/recall registers.

The N432A returns the data in the form of two numeric parameters and as many strings as there are save/recall registers.

<numeric\_value>,<numeric\_value>{,<string>}

- The first numeric parameter indicates the amount of memory (in bytes) used for the storage of registers.
- The second numeric parameter indicates the memory (in bytes) available for the storage of registers.
- Each string parameter returned indicates the name, type, and size of a save/recall register:

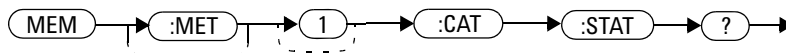
<name>,<type>,<size>

- <name> indicates the name of the save/recall register.
- <type> indicates STAT for the save/recall register.
- <size> indicates the size of the save/recall register in bytes.

A sample response may look like the following:

0,16190,"State1,STAT,0","State2,STAT,0" .....

### Syntax



### Example

MEM:CAT:STAT?      *Queries the list of save/recall registers.*

## MEMory[:METer][1]:CATalog:TABLE?

This query is used to list the stored sensor calibration and frequency-dependent offset tables.

The N432A returns the data in the form of two numeric parameters and as many strings as there are stored tables.

```
<numeric_value>,<numeric_value>{,<string>}
```

- The first numeric parameter indicates the amount of memory (in bytes) used for the storage of tables.
- The second numeric parameter indicates the memory (in bytes) available for the storage of tables.
- Each string parameter returned indicates the name, type, and size of a stored table:

```
<name>,<type>,<size>
```

- <name> indicates the name of the table.
- <type> indicates TABL for a table.
- <size> indicates the size of the table in bytes.

A sample response may look like the following:

```
198,26640,"DEFAULT,TABL,14","478A,TABL,86","8478B,TABL,98","CUSTOM_0,TABL,0","CUSTOM_1,TABL,0","CUSTOM_2,TABL,0","CUSTOM_3,TABL,0","CUSTOM_4,TABL,0","CUSTOM_5,TABL,0","CUSTOM_6,TABL,0","CUSTOM_7,TABL,0","CUSTOM_8,TABL,0","CUSTOM_9,TABL,0","CUSTOM_10,TABL,0","CUSTOM_11,TABL,0","CUSTOM_12,TABL,0","CUSTOM_13,TABL,0","CUSTOM_14,TABL,0","CUSTOM_15,TABL,0","CUSTOM_16,TABL,0","CUSTOM_A,TABL,0","CUSTOM_B,TABL,0","CUSTOM_C,TABL,0","CUSTOM_D,TABL,0","CUSTOM_E,TABL,0","CUSTOM_F,TABL,0","CUSTOM_G,TABL,0","CUSTOM_H,TABL,0","CUSTOM_I,TABL,0","CUSTOM_J,TABL,0"
```



The N432A is shipped with a set of predefined sensor calibration tables. The data in these tables is based on statistical averages for a range of Agilent thermistor sensors. These tables can be edited and are as listed below.

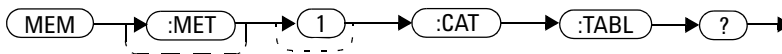
**Table 10-2** Predefined sensor calibration tables

Table	Thermistor sensor	Table name
0	None	DEFAULT
1	478A	478A
2	8478B	8478B

#### NOTE

There are also 17 sensor calibration tables named CUSTOM\_0 through CUSTOM\_16 and ten frequency-dependent offset tables named CUSTOM\_A through CUSTOM\_J which do not contain any data when the N432A is shipped from the factory.

## Syntax



## Example

`MEM:CAT:TABLE?`      *Queries the list of stored tables.*

## MEMory[:METer][1]:CLEar Commands

These commands are used to remove the contents stored in the sensor calibration tables, frequency-dependent offset tables, and save/recall registers. This subsystem removes the data contents but does not affect the name of the associated table or save/recall register.

The following commands are detailed in this section:

```
MEMory[:METer][1]:CLEar[:NAME] <character_data>
```

```
MEMory[:METer][1]:CLEar:TABLE
```

### NOTE

The contents cleared using these commands are non-recoverable.

---

## MEMory[:METer][1]:CLEar[:NAME] <character\_data>

This command clears the contents of a specified sensor calibration table, frequency-dependent offset table, or save/recall register.

Although the table remains, a MEMory[:METer][1]:TABLE:

FREQuency|GAIN:POINts? query returns a 0 as there are no contents in the table.

For sensor calibration tables and frequency-dependent offset tables, this command is an alternative form of the MEMory:CLEar:TABLE command, the only difference being the method in which the table is selected.

### NOTE

The contents cleared using this command are non-recoverable.

### Syntax



### Parameter

Item	Description/Default	Range of values
character_data	Contains an existing table name or save/recall register	Any existing table name or save/recall register

### Example

MEM:CLE "478A" *This command clears the contents of sensor calibration table 478A.*

### Error message

If the table or save/recall register name does not exist, error -224, "Illegal parameter value" occurs.

## MEMory[:METer][1]:CLEAr:TABLE

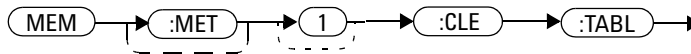
This command is used to clear the contents of the table currently selected using `MEMory:TABLE:SElect`. Although the table remains, a `MEMory:TABLE:FREQuency|GAIN:POINts?` query returns a 0 as the table contents are empty.

This command is an alternative form of the `MEMory:CLEAr[:NAME]` command. The difference is the method in which the table is selected.

### NOTE

The contents cleared using this command are non-recoverable.

### Syntax



### Example

```
MEM:CLE:TABLE
```

*This command clears the contents of the currently selected table.*

### Error message

If no table is selected, error -221, “Settings conflict” occurs.

## MEMory[:METer][1]:FREE Queries

These queries are used to retrieve information on the amount of free memory space available for sensor calibration tables, frequency-dependent offset tables, and save/recall registers.

The following queries are described in this section:

```
MEMory[:METer][1]:FREE[:ALL]?
```

```
MEMory[:METer][1]:FREE:STATe?
```

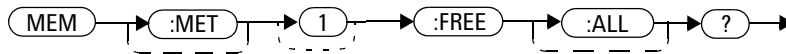
```
MEMory[:METer][1]:FREE:TABLE?
```

## MEMory[:METer][1]:FREE[:ALL]?

This query returns the amount of free memory space for sensor calibration tables, frequency-dependent offset tables, and save/recall registers. The response format is:

<bytes\_available>, <bytes\_in\_use>

### Syntax



### Example

MEM:FREE?

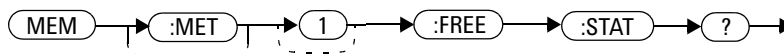
*Queries the total amount of free memory space.*

## MEMory[:METer][1]:FREE:STATe?

This query returns the amount of free memory space for save/recall registers. The response format is:

<bytes\_available>,<bytes\_in\_use>

### Syntax



### Example

`MEM:FREE:STAT?`

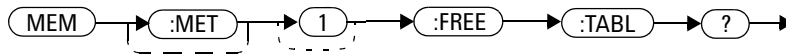
*Queries the amount of free memory space for save/recall registers.*

## MEMory[:METer][1]:FREE:TABLE?

This query returns the amount of free memory space for sensor calibration and frequency-dependent offset tables. The response format is:

<bytes\_available>,<bytes\_in\_use>

### Syntax



### Example

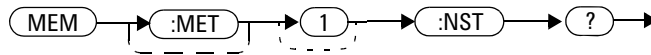
`MEM:FREE:TABL?`      *Queries the amount of free memory space for tables.*



## MEMory[:METer][1]:NSTates?

This query returns the number of registers that are available for save/recall. As there are ten registers, this query always returns ten.

### Syntax



### Example

MEM:NST?

*Queries the number of registers available for save/recall.*

## MEMory[:METer][1]:STATe Commands

These commands are used to query and define register names.

The following commands are described in this section:

```
MEMory[:METer][1]:STATe:CATalog?
```

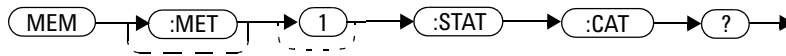
```
MEMory[:METer][1]:STATe:DEFine
```

## MEMory[:METer][1]:STATe:CATalog?

This query returns a list of the save/recall register names in the ascending order of the register number. The response format is:

<string>,<string>,...,<string>

### Syntax



### Example

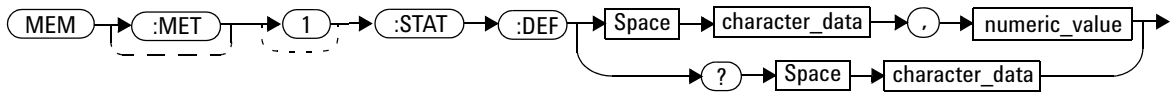
MEM:STAT:CAT?

*Queries the register names.*

## MEMory[:METer][1]:STATe:DEFine <character\_data>,<numeric\_value>

This command is used to associate a name with a save/recall register number.

### Syntax



### Parameters

Item	Description/Default	Range of values
character_data	Specifies the register name. A maximum of 12 characters can be used.	A to Z (uppercase) a to z (lowercase) 0 to 9 _ (underscore)
numeric_value	A numeric value (<NR1>) for the register number	0 to 9

### Example

MEM:STAT:DEF "SETUP1",4      *This command names register 4 SETUP1.*

### Query

MEMory[:METer][1]:STATe:DEFine? <character\_data>

The query returns the register number for the given register name.

## Query example

MEM:STAT:DEF? "SETUP1"     *Queries the register number of SETUP1.*

## Error messages

- If the register number is out of range, error -222, "Data out of range" occurs.
- If the name is invalid, error -224, "Illegal parameter value" occurs.
- If a register or sensor calibration table with the same name already exists, error -257, "File name error" occurs (command only).

## MEMory[:METer][1]:TABLE Commands

These commands are used to define a sensor calibration table or a frequency-dependent offset table, and to write to and read data from it.

The following commands are described in this section:

```
MEMory[:METer][1]:TABLE:FREQuency
<numeric_value>{,<numeric_value>}
```

```
MEMory[:METer][1]:TABLE:FREQuency:POINTs?
```

```
MEMory[:METer][1]:TABLE:GAIN[:MAGNitude]
<numeric_value>{,<numeric_value>}
```

```
MEMory[:METer][1]:TABLE:GAIN[:MAGNitude]:POINTs?
```

```
MEMory[:METer][1]:TABLE:MOVE <character_data>,<character_data>
```

```
MEMory[:METer][1]:TABLE:SElect <character_data>
```

## MEMory[:METer][1]:TABLE:FREQuency <numeric\_value>{,<numeric\_value>}

This command is used to enter frequency data into the currently selected table. Any previous frequency list is cleared before the new frequency list is stored. The frequencies must be entered in the ascending order. Entries in the frequency lists correspond to entries in the calibration/offset factor lists.

### NOTE

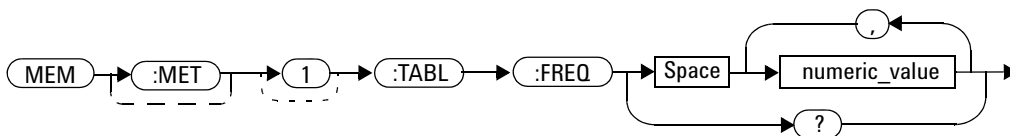
For sensor calibration tables only, the first calibration factor entered using the `MEMory:TABLE:GAIN` command is used as the reference calibration factor.

For sensor calibration tables, the number of frequency points must be one less than the number of calibration factor points. This is verified when the sensor calibration table is selected using `SENSe:CORRection:CSET:SElect <string>`.

Ensure that the frequency points you use cover the frequency range of the signals that you want to measure. If you measure a signal with a frequency outside the frequency range defined in the table, then the N432A uses the highest or lowest point in the table to calculate the calibration factor/offset.

Depending on available memory, the N432A is capable of storing 20 sensor calibration tables and 10 frequency-dependent offset tables, each containing 80 points.

### Syntax



## Parameter

Item	Description/Default	Range of values
numeric_value	A numeric value for the frequency. The default unit is Hz.	1 kHz to 1000.0 GHz <sup>[1], [2]</sup>

[1] The following measurement units can be used:

Hz  
 kHz ( $10^3$ )  
 MHz ( $10^6$ )  
 GHz ( $10^9$ )

[2] All frequencies are truncated to a multiple of 1 kHz.

## Example

`MEM:TABLE:FREQ 200kHz,600kHz`      *This command enters frequencies of 200 kHz and 600 kHz into the currently selected table.*

## Query

`MEMory[:METer][1]:TABLE:FREQuency?`

This query returns a list of frequency points in Hz for the currently selected table.



## Query example

MEM:TABL:FREQ?

*Queries the frequency points in the currently selected table.*

## Error messages

- If there are more than 80 frequency points in the list, error -108, "Parameter not allowed" occurs.
- If the frequencies are not entered in the ascending order, error -220, "Parameter error; Frequency list must be in ascending order" occurs.
- If a table has not been specified using the MEMory:TABLE:SElect command, the data cannot be entered into the table and error -221, "Settings conflict" occurs.
- If a frequency is sent which is outside of the allowed frequency range, error -222, "Data out of range" occurs.

## MEMory[:METer][1]:TABLe:FREQuency:POINts?

This query returns the number of frequency points for the currently selected table. The response format is <NRf>. If no frequency values have been set, this command returns a 0. If no table is selected, this command returns NAN.

### Syntax



### Example

```
MEM:TABL:FREQ:POIN?
```

*Queries the number of frequency points in the current table.*

## MEMory[:METer][1]:TABLE:GAIN[:MAGNitude] <numeric\_value>{,<numeric\_value>}

This command is used to enter calibration factors into the sensor calibration table or offsets into the frequency-dependent offset table, currently selected using `MEMory[:METer][1]:TABLE:SElect`. Any previous calibration factor list, or offset list is cleared before the new calibration factors/offsets are stored.

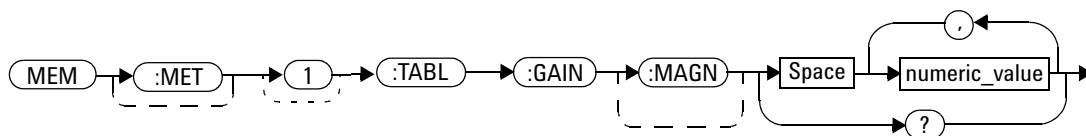
A maximum of 81 parameters for sensor calibration tables and 80 parameters for frequency-dependent offset tables can be sent with this command. For sensor calibration tables only, the first parameter is the reference calibration factor and each subsequent parameter is a calibration factor point in the sensor calibration table.

Entries in the frequency lists correspond to entries in the calibration factor/offset lists as shown in the following table:

Frequency	Calibration factor/offset
–	Reference calibration factor (For sensor calibration tables only)
Frequency 1	Calibration factor/Offset 1
“	“
Frequency 80	Calibration factor/Offset 80

For sensor calibration tables, the number of frequency points must be one less than the number of calibration factor data points. This is verified when the sensor calibration table is selected using `SENSE:CORREction:CSET1:SElect <string>`.

### Syntax



## Parameter

Item	Description/Default	Range of values
numeric_value	A numeric value for the calibration factor/offset. The unit is PCT.	1.0 to 150.0

## Example

```
MEM:TABLE:SEL "Sensor_1"
MEM:TABLE:GAIN 97,99.5,97.4
```

*This command enters a reference calibration factor of 97% and calibration factors of 99.5% and 97.4% into the selected sensor calibration table.*

## Query

```
MEMory[:METer][1]:TABLE:GAIN[:MAGNitude]?
```

This query returns a list of calibration factor/offset points for the currently selected table.

## Query example

```
MEM:TABLE:GAIN? Queries the calibration factor/offset in the current table.
```

## Error messages

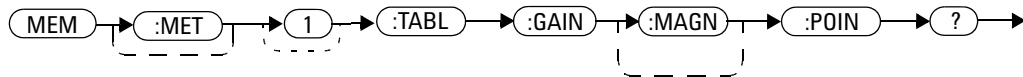
- If more than 81 calibration factors for sensor calibration tables, or 80 offsets for frequency-dependent offset tables are in the list, error -108, "Parameter not allowed" occurs.
- If a table is not specified using the MEMory:TABLE:SElect command, the data cannot be entered and error -221, "Settings conflict" occurs.
- If any of the calibration factors/offsets are outside of the allowed range, error -222, "Data out of range" occurs.

## MEMory[:METer][1]:TABLE:GAIN[:MAGNitude]:POINts?

This query is used to return the number of calibration factor/offset points for the currently selected table. If the currently selected table is a sensor calibration table, the reference calibration factor is included.

If no values have been set, 0 is returned. If no table is selected, NAN is returned.

### Syntax



### Example

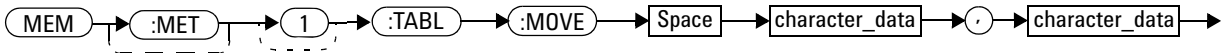
```
MEM:TABL:GAIN:POIN?
```

*Queries the number of calibration factor/offset points in the current table.*

## MEMory[:METer][1]:TABLe:MOVE <character\_data>,<character\_data>

This command is used to rename a sensor calibration table or a frequency-dependent offset table.

### Syntax



### Parameters

Item	Description/Default	Range of values
character_data (1st parameter)	Contains the existing table name	Existing table name
character_data (2nd parameter)	Specifies the new table name. A maximum of 12 characters can be used.	A to Z (uppercase) a to z (lowercase) 0 to 9 _ (underscore)

### Example

```
MEM:TABL:MOVE "tab1", "tab1a"
```

*This command renames a table named tab1 to tab1a.*

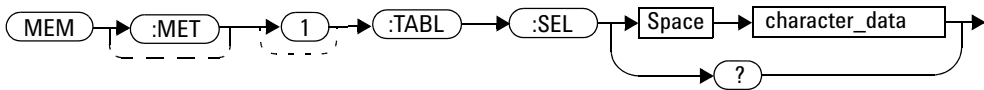
### Error messages

- If either of the table names specified is invalid, error -224, “Illegal parameter value” occurs.
- If the first parameter does not match an existing table name, error -256, “File name not found” occurs.
- If the second parameter matches an existing table name or save/recall register, error -257, “File name error” occurs.

## MEMory[:METer][1]:TABLE:SElect <character\_data>

This command is used to activate either a sensor calibration table, or a frequency-dependent offset table. A table must be activated before any operation can be performed on it.

### Syntax



### Parameter

Item	Description/Default	Range of values
character_data	Specifies the table name. A maximum of 12 characters can be used.	A to Z (uppercase) a to z (lowercase) 0 to 9 _ (underscore)

### Example

```
MEM:TABL:SEL "Sensor1"
```

*This command selects a sensor calibration table named "Sensor1".*

### Query

```
MEMory[:METer][1]:TABLE:SElect?
```

This query returns the name of the currently selected table.

### Query example

```
MEM:TABL:SEL? Queries the name of the current table.
```







## 11 OUTPut Subsystem

OUTPut Subsystem	282
OUTPut:RECOder[1]:FEED <data_handle>	283
OUTPut:RECOder[1]:LIMit:AUTO <boolean>	285
OUTPut:RECOder[1]:LIMit:LOWer <numeric_value>	286
OUTPut:RECOder[1]:LIMit:UPPer <numeric_value>	288
OUTPut:RECOder[1]:STATe <boolean>	290
OUTPut:ROSCillator[:STATe] <boolean>	291

This chapter explains how the `OUTPut` command subsystem is used to configure the N432A recorder and POWER REF outputs.



## OUTPut Subsystem

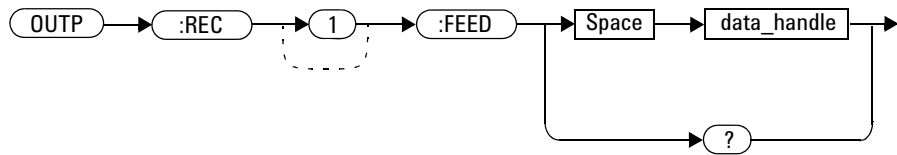
The OUTPut command subsystem is used to control the recorder output, and switch on and off the POWER REF output.

Keyword	Parameter form	Note	Page
OUTPut			
:REcorder[1]			
:FEED	<data_handle>		<a href="#">page 283</a>
:LIMit			
:AUTO	<boolean>		<a href="#">page 285</a>
:LOWer	<numeric_value>		<a href="#">page 286</a>
:UPPer	<numeric_value>		<a href="#">page 288</a>
:STATe	<boolean>		<a href="#">page 290</a>
:ROScillator			
[:STATe]	<boolean>		<a href="#">page 291</a>

## OUTPut:RECOOrder[1]:FEED <data\_handle>

This command determines which measurement is sent to the recorder output.

### Syntax



### Parameter

Item	Description/Default	Range of values
data_handle	The CALC block specifying the measurement to be sent to the recorder output	"CALC1" or "CALC" "CALC2" "CALC3" "CALC4"

### Example

OUTPut:RECOOrder:FEED "CALC1"

*This command sends the CALC1 measurement to the recorder output.*

### Reset condition

On reset, data\_handle is set to its previous value.

### Query

```
OUTPut:REcOrder[1]:FEED?
```

This query returns the current value of `data_handle` for the recorder output.

### Query example

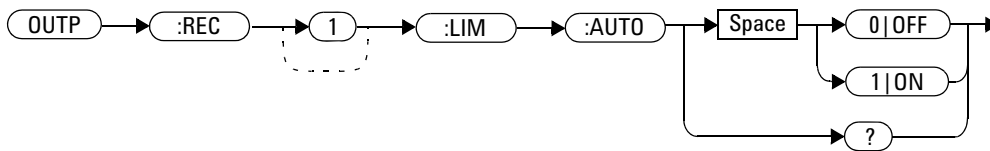
```
OUTP:REC:FEED?
```

*Queries the value of `data_handle` for the recorder output.*

## OUTPut:RECOOrder[1]:LIMit:AUTO <boolean>

This command enables or disables autoscaling for the recorder output. Presetting the N432A disables the autoscaling.

### Syntax



### Example

```
OUTP:REC:LIM:AUTO 1
```

*This command enables autoscaling for the recorder output.*

### Query

```
OUTPut:RECOOrder[1]:LIMit:AUTO?
```

This query enters a 1 or 0 into the output buffer indicating whether autoscaling is enabled or disabled for the recorder output:

- 1 is returned when autoscaling is enabled.
- 0 is returned when autoscaling is disabled.

### Query example

```
OUTP:REC:LIM:AUTO?
```

*Queries the current autoscaling state for the recorder output.*

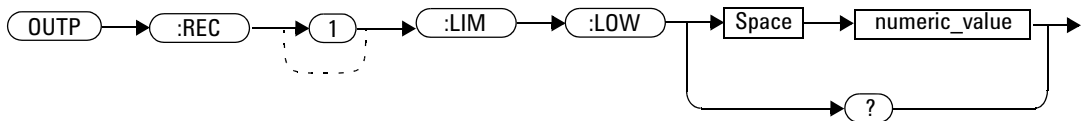
### Error message

If `OUTPut:RECOOrder[1]:LIMit:AUTO` is set to ON while `CALCulate[1]|2|3|4:MATH[:EXPRession]` is set to a difference or ratio measurement, error -221, “Settings conflict” occurs.

## OUTPut:RECOOrder[1]:LIMit:LOWer <numeric\_value>

This command sets the minimum scaling value for the recorder output. The unit used is dependent on the unit currently set for the CALC block specified in `OUTPut:RECOOrder[1]:FEED <data_handle>`.

### Syntax



### Parameter

Item	Description/Default	Range of values
numeric_value	A numeric value for the minimum scaling value. The unit used, dBm, W, or %, is dependent on the unit currently set for the CALC block specified in <code>OUTPut:RECOOrder[1]:FEED &lt;data_handle&gt;</code> .	-150 to +230 dBm 1 aW to 100 XW 0% to 999%

### Example

`OUTPut:RECOOrder[1]:LIMit:LOW -90`     *This command sets the minimum scaling value to -90 dBm for the recorder output.*

### Reset condition

On reset, the minimum scaling value is set to -150 dBm.

## Query

```
OUTPut:REcorder[1]:LIMit:LOWer?
```

This query returns the minimum scaling value for the recorder output.

## Query example

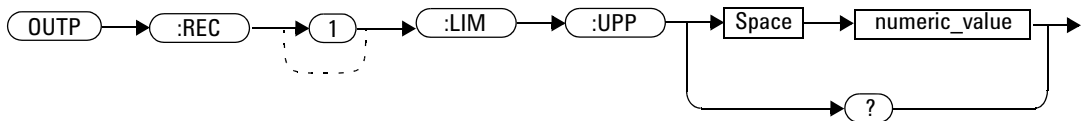
```
OUTP:REC:LIM:LOW?
```

*Queries the minimum scaling value for the recorder output.*

## OUTPut:RECOOrder[1]:LIMit:UPPer <numeric\_value>

This command sets the maximum scaling value for the recorder output. The unit used is dependent on the unit currently set for the CALC block specified in `OUTPut:RECOOrder[1]:FEED <data_handle>`.

### Syntax



### Parameter

Item	Description/Default	Range of values
numeric_value	A numeric value for the maximum scaling value. The unit used, dBm, W, or %, is dependent on the unit currently set for the CALC block specified in <code>OUTPut:RECOOrder[1]:FEED &lt;data_handle&gt;</code> .	-150 to +230 dBm 1 aW to 100 XW 0 % to 999 %

### Example

`OUTPut:RECOOrder[1]:LIMit:UPP 10`      *This command sets the maximum scaling value to 10 for the recorder output.*

### Reset condition

On reset, the maximum scaling value is set to 20 dBm.



## Query

```
OUTPut:RECOder[1]:LIMit:UPPer?
```

This query returns the maximum scaling value for the recorder output.

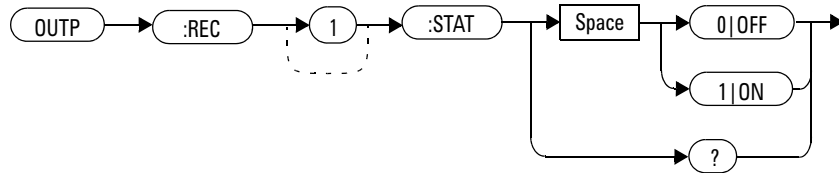
## Query example

```
OUTP:REC:LIM:UPP?    Queries the maximum scaling value for the  
recorder output.
```

## OUTPut:RECOOrder[1]:STATe <boolean>

This command enables or disables the recorder output.

### Syntax



### Example

OUTP:REC:STAT 1      *This command enables the recorder output.*

### Reset condition

On reset, the recorder output is disabled.

### Query

OUTPut:RECOOrder[1]:STATe?

This query enters a 1 or 0 into the output buffer indicating whether the recorder output is enabled or disabled:

- 1 is returned when the recorder output is turned ON.
- 0 is returned when the recorder output is turned OFF.

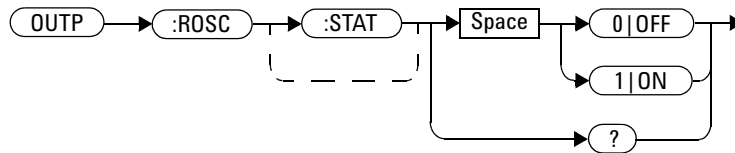
### Query example

OUTP:REC:STAT?      *Queries the status of the recorder output.*

## OUTPut:ROSCillator[:STATe] <boolean>

This command enables or disables the POWER REF output.

### Syntax



### Example

OUTPut:ROSCillator:STAT 1      *This command enables the POWER REF output.*

### Reset condition

On reset, the POWER REF output is disabled.

### Query

OUTPut:ROSCillator[:STATe]?

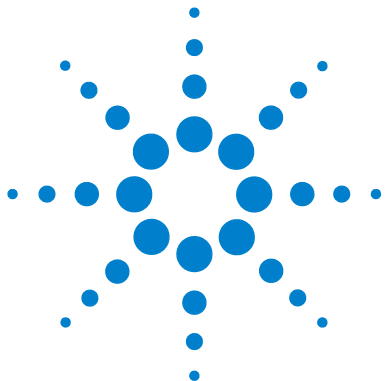
This query enters a 1 or 0 into the output buffer indicating whether the POWER REF output is enabled or disabled.

- 1 is returned when the POWER REF output is enabled.
- 0 is returned when the POWER REF output is disabled.

### Query example

OUTPut:ROSC?      *Queries the status of the POWER REF output.*





## 12 SENSe Subsystem

SENSe Subsystem	294
[SENSe[1]:]AVERAge Commands	296
[SENSe[1]:]AVERAge:COUNT:VOLTage <character_data>	297
[SENSe[1]:]AVERAge:SDETECT <boolean>	299
[SENSe[1]:]BRESistance <character_data>	301
[SENSe[1]:]RSELECTION <character_data>	303
[SENSe[1]:]RVALue <numeric_value>	305
[SENSe[1]:]CORRection Commands	307
[SENSe[1]:]CORRection:CFACtor   GAIN[1][:INPut][:MAGNitude] <numeric_value>	308
[SENSe[1]:]CORRection:CSET[1] Commands	310
[SENSe[1]:]CORRection:CSET2 Commands	311
[SENSe[1]:]CORRection:CSET[1][:SElect] <string>	312
[SENSe[1]:]CORRection:CSET2[:SElect] <string>	314
[SENSe[1]:]CORRection:CSET[1]:STATe <boolean>	316
[SENSe[1]:]CORRection:CSET2:STATe <boolean>	318
[SENSe[1]]   2   3   4:CORRection:DCYCLE   GAIN3 Commands	320
[SENSe[1]:]CORRection:DCYCLE   GAIN3[:INPut][:MAGNitude] <numeric_value>	321
[SENSe[1]:]CORRection:DCYCLE   GAIN3:STATe <boolean>	323
[SENSe[1]:]CORRection:FDOFFset   GAIN4[:INPut][:MAGNitude]? <boolean>	325
[SENSe[1]:]CORRection:FDOFFset:UNIT <character_data>	326
[SENSe[1]:]CORRection:GAIN2 Commands	327
[SENSe[1]:]CORRection:GAIN2:STATe <boolean>	328
[SENSe[1]:]CORRection:GAIN2[:INPut][:MAGNitude] <numeric_value>	329
[SENSe[1]:]FREQuency[:CW]   :FIXed <numeric_value>	331

This chapter explains how the `SENSe` command subsystem is used to configure device-specific settings for making measurements.



## SENSE Subsystem

The SENSE command subsystem directly affects device-specific settings used to make measurements. The SENSE subsystem is optional since this is the primary function of the N432A. The high-level command CONFIGure uses the SENSE commands to prepare the N432A for making measurements. At a lower level, SENSE enables you to change the AVERage, BRESistance, RSElection, RVALue, FREQuency, and CORRection parameters without completely reconfiguring the N432A.

The SENSE command subsystem also allows you to select a sensor calibration table and a frequency-dependent offset table.

The numeric suffix of the SENSE command (SENSE1) represents channel A.

Keyword	Parameter form	Note	Page
[SENSE[1]:]			
AVERage			
:COUNT			
:VOLTage	<character_data>		<a href="#">page 297</a>
:SDETECT	<boolean>		<a href="#">page 299</a>
BRESistance	<character_data>		<a href="#">page 301</a>
RSEL	<character_data>		<a href="#">page 303</a>
RVALue	<numeric_value>		<a href="#">page 305</a>
CORRection			
:CFACTOR GAIN[1]			
[:INPut]			
[:MAGNitude]	<numeric_value>		<a href="#">page 308</a>
:CSET[1]			
[:SElect]	<string>		<a href="#">page 312</a>
:STATE	<boolean>		<a href="#">page 316</a>
:CSET2			
[:SElect]	<string>		<a href="#">page 314</a>
:STATE	<boolean>		<a href="#">page 318</a>

Keyword	Parameter form	Note	Page
:FDOffset   GAIN4 [:INPut] [:MAGNitude]?		[query only]	page 325
:FDOffset :UNIT	<character_data>		page 326
:GAIN2 :STATE [:INPut] [:MAGNitude]	<boolean>  <numeric_value>		page 328  page 329
FREQuency [:CW :FIXed]	<numeric_value>		page 331

## [SENSE[1]:]AVERAGE Commands

These commands allow you to configure the voltage measurement averaging used in incident power calculation and to set the step detection.

The following commands are detailed in this section:

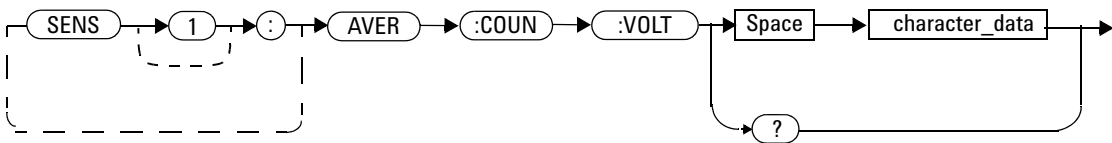
```
[SENSE[1]:]AVERAGE:COUNT:VOLTage <string>  
[SENSE[1]:]AVERAGE:SDETECT <boolean>
```



## [SENSe[1]:]AVERage:COUNT:VOLTage <character\_data>

This command allows you to set a voltage average value to be used in the calculation of the incident power.

### Syntax



### Parameter

Item	Description/Default	Range of values
character_data	Character data representing a voltage average value where: <ul style="list-style-type: none"> <li>• AVC4 indicates a value of 4</li> <li>• AVC8 indicates a value of 8</li> <li>• AVC16 indicates a value of 16</li> <li>• AVC32 indicates a value of 32</li> <li>• AVC64 indicates a value of 64</li> <li>• AVC128 indicates a value of 128</li> </ul>	AVC4 AVC8 AVC16 AVC32 AVC64 AVC128

### Reset condition

On reset, the voltage average value is set to AVC16.

## Example

AVER:COUN:VOLT AVC16     *This command specifies a voltage average value of 16.*

## Query

[SENSE[1]:]AVERage:COUNT:VOLTage?

This query returns the current setting of the voltage average.

## Query example

AVER:COUN:VOLT?     *Queries the voltage average setting.*

## Error message

If the value specified for the <character\_data> parameter is invalid, error -224,"Illegal parameter value" occurs.

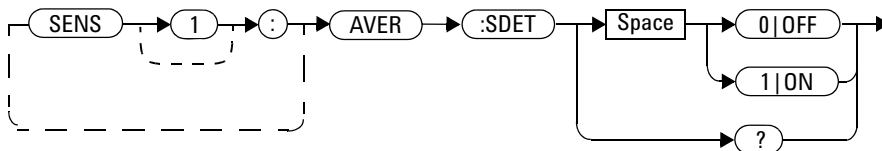
## [SENSe[1]:]AVERage:SDEct <boolean>

This command enables or disables step detection. Step detection shortens the filter time when the input power changes substantially, for the filter output to get to its final value. Note that this result appears to settle faster, although true settling to the final value is unaffected.

### NOTE

- Step detection is automatically disabled when TRIG : DEL : AUTO is set to ON and the trigger mode is set to free run. Under these circumstances, the value of SENS : AVER : SDET is ignored. Note also that SENS : AVER : SDET is not set by the N432A (that is, SENS : AVER : SDET retains its current setting which may indicate that step detection is turned ON).
- With certain pulsing signals, step detection may operate on the pulses, preventing the final average from being completed and making the results unstable. Under these conditions, step detection should be turned OFF.

### Syntax



### Reset condition

On reset, step detection is enabled.

### Example

```
AVER:SDET ON
```

*This command enables step detection.*

### Query

```
[SENSe[1]:]AVERage:SDETECT?
```

This query enters a 1 or 0 into the output buffer indicating the status of step detection:

- 1 is returned when step detection is enabled
- 0 is returned when step detection is disabled

### Query example

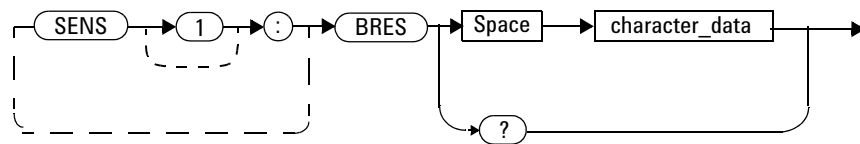
```
AVER:SDET?
```

*Queries whether step detection is turned on or off.*

## [SENSe[1]:]BRESistance <character\_data>

This command sets the value of the bridge resistance for the thermistor sensor.

### Syntax



### Parameter

Item	Description/Default	Range of values
character_data	Character data representing a bridge resistance value where: <ul style="list-style-type: none"> <li>• R100 indicates 100 <math>\Omega</math></li> <li>• R200 indicates 200 <math>\Omega</math></li> <li>• R300 indicates 300 <math>\Omega</math></li> <li>• R400 indicates 400 <math>\Omega</math></li> </ul>	R100 R200 R300 R400

### Reset condition

On reset, the bridge resistance is not affected.

## Example

BRES R100      *This command sets the bridge resistance to 100  $\Omega$*

## Query

[SENSe[1]:]BRESistance?

This query returns the current setting of the bridge resistance in the ohms unit.

## Query example

BRES?      *Queries the bridge resistance setting.*

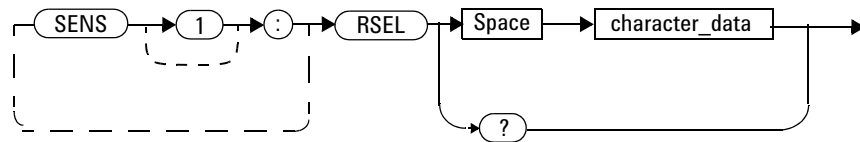
## Error message

If the value specified for the <character\_data> parameter is invalid, error -224,"Illegal parameter value" occurs.

## [SENSe[1]:]RSElection <character\_data>

This command selects the resistance type of Rmeas or Ruser for the thermistor sensor. When Rmeas is selected, an externally measured bridge resistance value stored in EEPROM will be used. When Ruser is selected, you can enter a resistance value to be used in power measurement.

### Syntax



### Parameter

Item	Description/Default	Range of values
character_data	Character data representing the resistance type.	MEAS USER

### Reset condition

On reset, the resistance type is set to Rmeas.

### Example

RSEL MEAS      *This command sets the resistance type to Rmeas.*

### Query

```
[SENSe[1]:]RSElection?
```

This query returns the current resistance type.

### Query example

```
RSEL?           Queries the resistance type.
```

### Error message

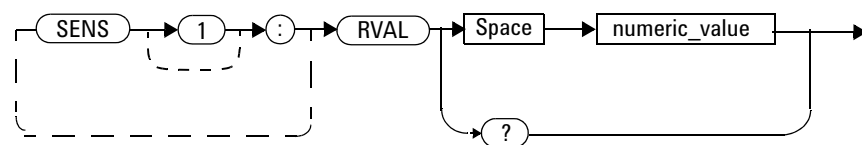
If the value specified for the <character\_data> parameter is invalid, error -224,"Illegal parameter value" occurs.



## [SENSe[1]:]RVALue <numeric\_value>

This command sets the resistance value for Ruser. Both the Rmeas and Ruser values correspond to the current bridge resistance setting. The value specified for Ruser must be within a  $\pm 10\%$  tolerance level of the bridge resistance. If this value exceeds the tolerance level, it will be clipped to the upper/lower limit within tolerance. To measure the value for Ruser, refer to the *N432A Thermistor Power Meter User's Guide*.

### Syntax



### Parameter

Item	Description/Default	Range of values
numeric_value	A numeric value for the Ruser resistance setting	Within a $\pm 10\%$ tolerance level of the current bridge resistance setting

### Reset condition

On reset, the resistance value of Ruser is not affected.

## Example

RVAL 110      *This command sets the resistance value for Ruser to 110  $\Omega$*

## Query

[SENSe[1]:]RVALue?

This query returns the current resistance value of Ruser or Rmeas, depending on which type is being currently used.

## Query example

RVAL?      *Queries the resistance value of the current resistance type.*

## Error messages

- The [SENSe[1]:]RVALue command is only applicable for Ruser. If this command is sent when the resistance type is Rmeas, error -221,"Settings conflict" occurs.
- If the <numeric\_value> parameter exceeds the  $\pm 10\%$  tolerance level of the bridge resistance, error -222,"Data out of range" occurs.

## [SENSe[1]:]CORRection Commands

These commands provide for changes to be applied to the measurement result. They are used to enter duty cycle values, calibration factors and other external gains and losses.

The following commands are detailed in this section:

```
[SENSe[1]:]CORRection:CFACTOR|GAIN[1][:INPut][:MAGNitude]
<numeric_value>
```

```
[SENSe[1]:]CORRection:CSET[1]|CSET2[:SElect] <string>
```

```
[SENSe[1]:]CORRection:CSET[1]|CSET2:STATe <boolean>
```

```
[SENSe[1]:]CORRection:DCYClE|GAIN3[:INPut][:MAGNitude]
<numeric_value>
```

```
[SENSe[1]:]CORRection:DCYClE|GAIN3:STATe <boolean>
```

```
[SENSe[1]:]CORRection:FDOFfset|GAIN4[:INPut][:MAGNitude]?
```

```
[SENSe[1]:]CORRection:FDOFfset:UNIT <character_data>
```

```
[SENSe[1]:]CORRection:GAIN2:STATe <boolean>
```

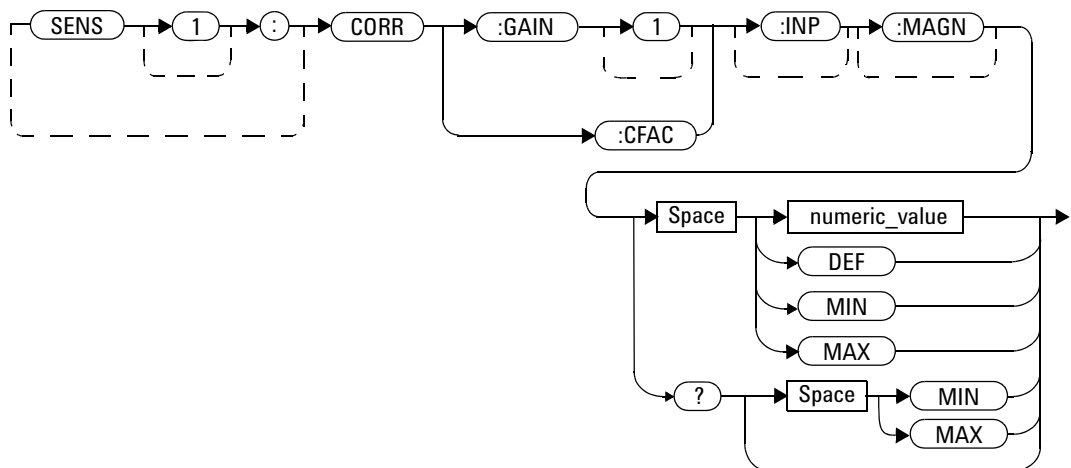
```
[SENSe[1]:]CORRection:GAIN2[:INPut][:MAGNitude] <numeric_value>
```

## [SENSe[1]:]CORRection:CFACTOR | GAIN[1][:INPut][:MAGNitude] <numeric\_value>

This command is used to enter a gain correction value for the calibration factor. The N432A corrects every measurement by this factor to compensate for the gain.

Either CFACTOR or GAIN1 can be used in the command; both have an identical result. Using GAIN1 complies with the SCPI standard, whereas CFACTOR does not; this may make your program easier to understand.

### Syntax



### Parameter

Item	Description/Default	Range of values
numeric_value	A numeric value for CFACTOR or GAIN1 <ul style="list-style-type: none"> <li>• DEF: The default value is 100%</li> <li>• MIN: 1%</li> <li>• MAX: 150%</li> </ul>	1 to 150 PCT <sup>[1]</sup> DEF MIN MAX

[1] For example, a gain of 60 % corresponds to a multiplier of 0.6 and a gain of 150 % corresponds to a multiplier of 1.5.

## Example

`CORR:GAIN 100 PCT`      *This command sets a gain correction of 100%.*

## Reset condition

On reset, `CFACTOR|GAIN1` is set to 100%.

## Query

```
[SENSe[1]:]CORRection:CFACTOR|GAIN[1][:INPut][:MAGNitude]?
[MIN|MAX]
```

This query returns the current gain correction setting or the values associated with MIN and MAX.

## Query example

`CORR:GAIN?`      *Queries the current gain correction setting.*

## Error message

If the `[SENSe[1]:]CORRection:CFACTOR|GAIN1` command is used with a sensor calibration table selected, error -221, “Settings Conflict” occurs.

## [SENSE[1]:]CORRection:CSET[1] Commands

These commands are used to select the active sensor calibration table.

The following commands are detailed in this section:

```
[SENSE[1]:]CORRection:CSET[1][:SElect] <string>
```

```
[SENSE[1]:]CORRection:CSET[1]:STATe <boolean>
```

## [SENSe[1]:]CORRection:CSET2 Commands

These commands are used to select the active frequency-dependent offset table.

The following commands are detailed in this section:

```
[SENSe[1]:]CORRection:CSET2[:SElect] <string>
```

```
[SENSe[1]:]CORRection:CSET2:STATE <boolean>
```

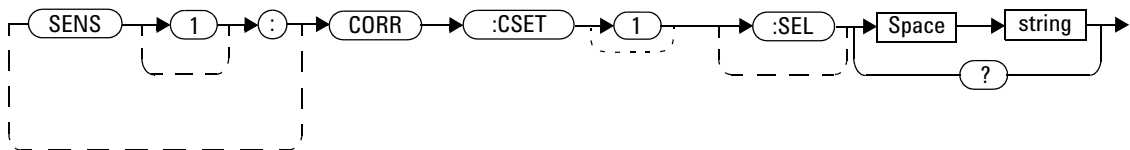
## [SENSe[1]:]CORRection:CSET[1][:SElect] <string>

This command enters the name of the required sensor calibration table. The calibration factor is interpolated from the table using the setting for [SENSe[1]:]FREQuency.

### NOTE

If [SENSe[1]:]CORRection:CSET[1]:STATE is set to OFF, the selected sensor calibration table is not being used.

### Syntax



### Parameter

Item	Description/Default	Range of values
string	String data representing a sensor calibration table name	Any existing table name (Existing table names can be listed using MEMORY:CATalog:TABLE?).

### Example

`CORR:CSET `PW1``      *This command enters the name of the sensor calibration table as PW1.*

### Reset condition

On reset, the selected table is not affected.



## Query

```
[SENSe[1]:]CORRection:CSET[1]:[SELEct]?
```

This query returns the name of the selected sensor calibration table as a quoted string. If no table is selected, an empty string is returned.

## Query example

```
CORR:CSET?      Queries the sensor calibration table currently used.
```

## Error messages

- If the value entered for <string> is not valid, error -224, “Illegal parameter value” occurs.
- If a table with the name specified by <string> does not exist, error -256, “File name not found” occurs.
- When a sensor calibration table is selected, the N432A verifies that the number of calibration points defined is one more than the number of frequency points defined. When a frequency-dependent offset table is selected, the N432A verifies that the number of offset points defined is equal to the number of frequency points defined. If this is not the case, error -226, “Lists not the same length” occurs.

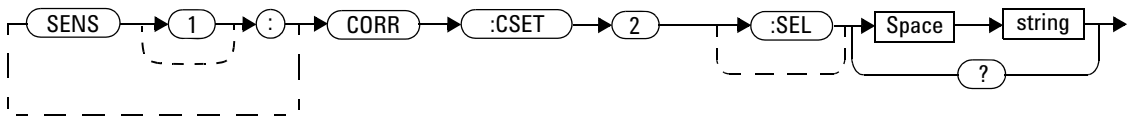
## [SENSe[1]:]CORRection:CSET2[:SElect] <string>

This command enters the name of the required frequency-dependent offset table.

**NOTE**

If [SENSe[1]:]CORRection:CSET2:STATE is set to OFF, the selected frequency-dependent offset table is not being used.

### Syntax



### Parameter

Item	Description/Default	Range of values
string	String data representing a frequency-dependent offset table name	Any existing table name (Existing table names can be listed using MEMory:CATalog:TABLE?).

### Example

CORR:CSET2 `PW1` *This command enters the name of the frequency-dependent offset table as PW1.*

## Reset condition

On reset, the selected table is not affected.

## Query

```
[SENSe[1]:]CORRection:CSET2[SELEct]?
```

This query returns the name of the selected frequency-dependent offset table as a quoted string. If no table is selected, an empty string is returned.

## Query example

```
CORR:CSET2?    Queries the frequency- dependent offset table currently used.
```

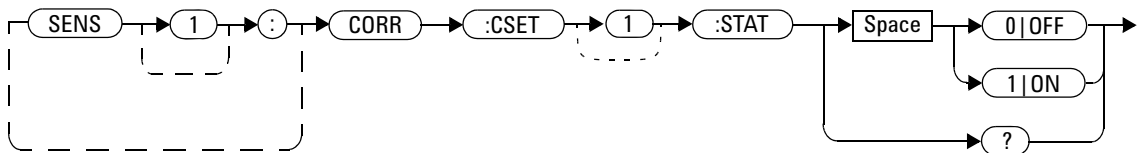
## Error messages

- If the value entered for <string> is not valid, error -224, “Illegal parameter value” occurs.
- If a table with a name specified by <string> does not exist, error -256, “File name not found” occurs.
- When a sensor calibration table is selected, the N432A verifies that the number of calibration points defined is one more than the number of frequency points defined. When a frequency-dependent offset table is selected, the N432A verifies that the number of offset points defined is equal to the number of frequency points defined. If this is not the case, error -226, “Lists not the same length” occurs.

## [SENSe[1]:]CORRection:CSET[1]:STATe <boolean>

This command is used to enable or disable the currently selected sensor calibration table. When a table has been selected and enabled, the calibration factors stored in it can be used by specifying the required frequency using the [SENSe[1]:]FREQuency command.

### Syntax



### Example

`CORR:CSET1:STAT 1` *This command enables the currently selected sensor calibration table.*

### Reset condition

On reset, the sensor calibration table is not affected.

### Query

`[SENSe[1]:]CORRection:CSET[1]:STATe?`

This query enters a 1 or 0 into the output buffer indicating whether a currently selected sensor calibration table is enabled or disabled.

- 1 is returned if the table is enabled.
- 0 is returned if the table is disabled.

## Query example

```
CORR:CSET1:STAT?
```

*Queries whether a currently selected sensor calibration table is enabled or disabled.*

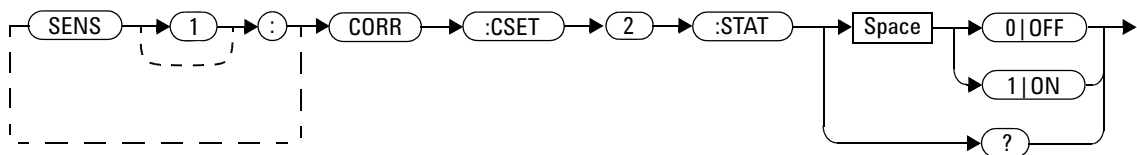
## Error message

If you attempt to set this command to ON and no table has been selected using [SENSe[1]:]CORRection:CSET[1]:[SElect], then error -221, "Settings conflict" will occur and [SENSe[1]:]CORRection:CSET[1]:STATe remains OFF.

## [SENSe[1]:]CORRection:CSET2:STATe <boolean>

This command is used to enable or disable the currently selected frequency-dependent offset table. When a table has been selected and enabled, the offsets stored in it can be used by specifying the required frequency using the [SENSe[1]:]FREQuency command.

### Syntax



### Example

`CORR:CSET2:STAT 1`      *This command enables the currently selected frequency-dependent offset table.*

### Reset condition

On reset, the frequency-dependent offset table is not affected.

### Query

`[SENSe[1]:]CORRection:CSET2:STATe?`

This query enters a 1 or 0 into the output buffer indicating whether a currently selected frequency-dependent offset table is enabled or disabled.

- 1 is returned if the table is enabled.
- 0 is returned if the table is disabled.

## Query example

`CORR:CSET2:STAT?`

*Queries whether a currently selected frequency-dependent offset table is enabled or disabled.*

## Error message

If this command is set to ON and no table has been selected using `[SENSe[1]:]CORRection:CSET2:[SElect]`, then error -221, “Settings conflict” will occur and `[SENSe[1]:]CORRection:CSET2:STATe` remains OFF.

## [SENSE[1]] | 2 | 3 | 4:CORREction:DCYCLE | GAIN3 Commands

These commands control the pulse power measurement feature of the N432A.

The following commands are detailed in this section:

```
[SENSE[1]:]CORREction:DCYCLE|GAIN3[:INPut][:MAGNitude]  
<numeric_value>  
[SENSE[1]:]CORREction:DCYCLE|GAIN3:STATe <boolean>
```

### NOTE

You can use either DCYCLE or GAIN3 in these commands; both perform the same function. Using GAIN3 complies with the SCPI standard whereas DCYCLE does not but it may make your program more understandable.

---



## [SENSe[1]:]CORRection:DCYClE | GAIN3[:INPut][:MAGNitude] <numeric\_value>

This command is used to set the duty cycle for the pulse power measurement feature of the N432A. Pulse power measurements average out any deviations in the pulse, such as, overshoot or ringing. The result returned for a pulse power measurement is a mathematical representation of the pulse power rather than an actual measurement. The N432A measures the average power in the pulsed input signal and then divides the result by the duty cycle value to obtain a pulse power reading.

Entering a value using this command automatically turns the [SENSe[1]:]CORRection:DCYClE|GAIN3:STATe command to ON.

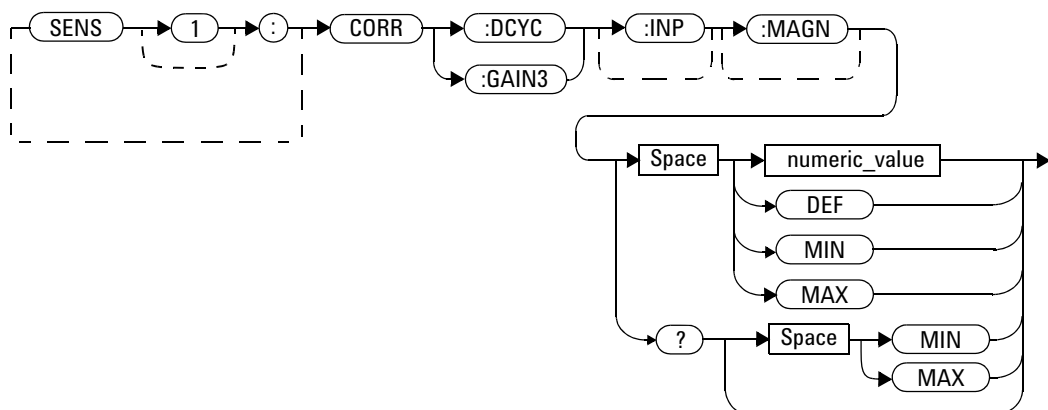
### NOTE

Pulse power averages out any deviations in the pulse such as overshoot or ringing. Hence, it is called pulse power and not peak power or peak pulse power.

To ensure accurate pulse power readings, the input signal must be pulsed with a rectangular pulse. Other pulse shapes (such as triangle, chirp, or Gaussian) will cause incorrect results.

The pulse power on/off ratio must be much greater than the duty cycle ratio.

### Syntax



## Parameter

Item	Description/Default	Range of values
numeric_value	A numeric value for the duty cycle <ul style="list-style-type: none"> <li>• DEF: Default value is 1%</li> <li>• MIN: 0.001%</li> <li>• MAX: 99.999%</li> </ul> The unit is PCT, and is optional	0.001 to 99.999 PCT DEF MIN MAX

## Example

`CORR:DCYC 90PCT`      *This command sets a duty cycle of 90%.*

## Reset condition

On reset, the duty cycle is 1%.

## Query

```
[SENSE[1]:]CORREction:DCYcle|GAIN3[:INPut][:MAGNitude]?
[MIN|MAX]
```

This query returns the current setting of the duty cycle or the values associated with MIN and MAX.

## Query example

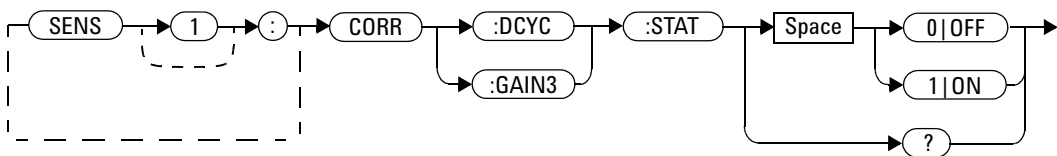
`CORR:GAIN3?`      *Queries the current setting of the duty cycle.*

## [SENSe[1]:]CORRection:DCYClE | GAIN3:STATe <boolean>

This command is used to enable or disable the pulse power measurement feature.

The [SENSe[1]:]CORRection:DCYClE|GAIN3 command should be used to enter the duty cycle of the signal you want to measure.

### Syntax



### Example

`CORR:DCYC:STAT 1` *This command enables pulse power measurement.*

### Reset condition

On reset, the pulse power measurement feature is disabled.

### Query

`[SENSe[1]:]CORRection:DCYClE|GAIN3:STATe?`

This query enters a 1 or 0 into the output buffer indicating the status of the pulse power measurement feature.

- 1 is returned when enabled.
- 0 is returned when disabled.

### Query example

`CORR:GAIN3:STAT?` *Queries whether pulse power measurement is enabled or disabled.*

## [SENSE[1]:]CORRection:FDOFFset Commands

These commands are used to set the unit for the frequency-dependent offset as well as retrieve the currently applied offset value.

The following commands are detailed in this section:

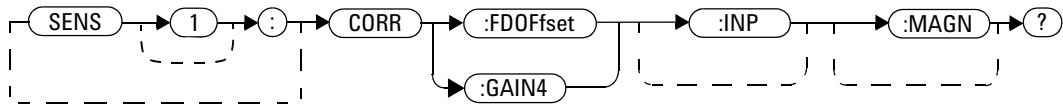
```
[SENSE[1]:]CORRection:FDOFFset|GAIN4[:INPut][:MAGNitude]?
```

```
[SENSE[1]:]CORRection:FDOFFset:UNIT <character_data>
```

## [SENSe[1]:]CORRection:FDOFFset | GAIN4[:INPut][:MAGNitude]?

This query is used to return the currently applied frequency-dependent offset value.

### Syntax



### Example

CORR:GAIN4?

*Queries the frequency- dependent offset being currently applied.*

### Reset condition

On reset, the frequency-dependent offset is not affected.

## [SENSe[1]:]CORRection:FDOFfset:UNIT <character\_data>

This command allows you to set the unit for the frequency-dependent offset.

### Syntax



### Parameter

Item	Description/Default	Range of values
character_data	Specifies a unit of either % or dB	PCT DB

### Example

CORR:FDOF:UNIT PCT *This command sets the unit to percent (%).*

### Query

[SENSe[1]:]CORRection:FDOFfset:UNIT?

This query returns the current unit for the frequency-dependent offset.

### Query example

CORR:FDOF:UNIT? *Queries the current frequency- dependent offset unit.*

## [SENSe[1]:]CORRection:GAIN2 Commands

These commands provide a simple correction to a measurement for an external gain/loss.

The following commands are detailed in this section:

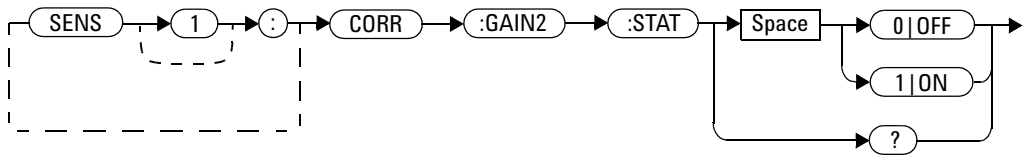
```
[SENSe[1]:]CORRection:GAIN2:STATe <boolean>
```

```
[SENSe[1]:]CORRection:GAIN2[:INPut][:MAGNitude]  
<numeric_value>
```

## [SENSe[1]:]CORRection:GAIN2:STATe <boolean>

This command is used to enable or disable a channel offset for the N432A setup. The [SENSe[1]:]CORRection:GAIN2[:INPut][:MAGNitude] command is used to enter the loss/gain value.

### Syntax



### Example

`CORR:GAIN2:STAT ON`      *This command enables a channel offset.*

### Reset condition

On reset, the channel offset is disabled.

### Query

`[SENSe[1]:]CORRection:GAIN2:STATe?`

This query enters a 1 or 0 into the output buffer indicating the status of the channel offset:

- 1 is returned if a channel offset is enabled.
- 0 is returned if a channel offset is disabled.

### Query example

`CORR:GAIN2:STAT?`      *Queries whether a channel offset is enabled or disabled.*

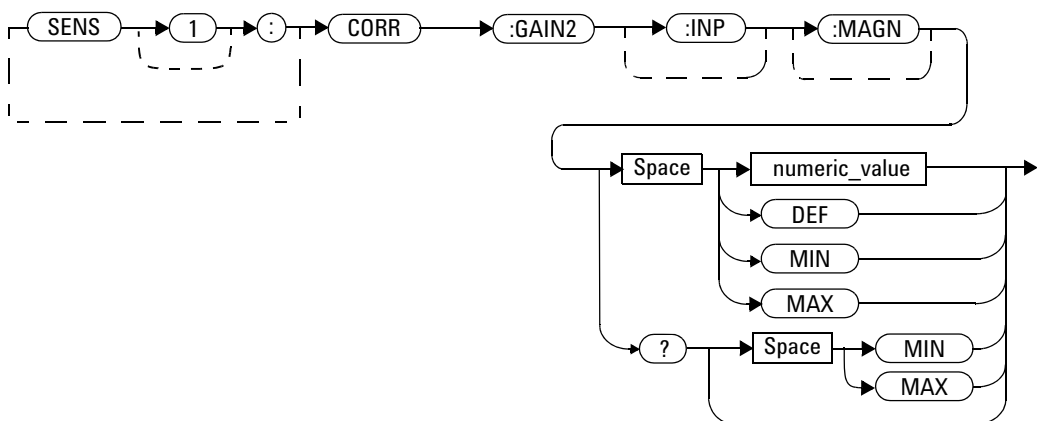


## [SENSe[1]:]CORRection:GAIN2[:INPut][:MAGNitude] <numeric\_value>

This command is used to enter a channel offset value for the N432A setup. The N432A then corrects every measurement by this value to compensate for the gain/loss.

Entering a value for this command automatically turns ON the [SENSe[1]:]CORRection:GAIN2:STATE command.

### Syntax



### Parameter

Item	Description/Default	Range of values
numeric_value	A numeric value for the channel offset <ul style="list-style-type: none"> <li>• DEF: The default value is 0.00 dB</li> <li>• MIN: -100 dB</li> <li>• MAX: +100 dB</li> </ul>	-100 to +100 dB DEF MIN MAX

### Example

`CORR:GAIN2 50`      *This command sets a channel offset of 50 dB.*

### Reset condition

On reset, the channel offset is set to 0.00 dB.

### Query

`[SENSE[1]:]CORREction:GAIN2[:INPut][:MAGNitude]? [MIN|MAX]`

This query returns the current setting of the channel offset or the values associated with MIN and MAX.

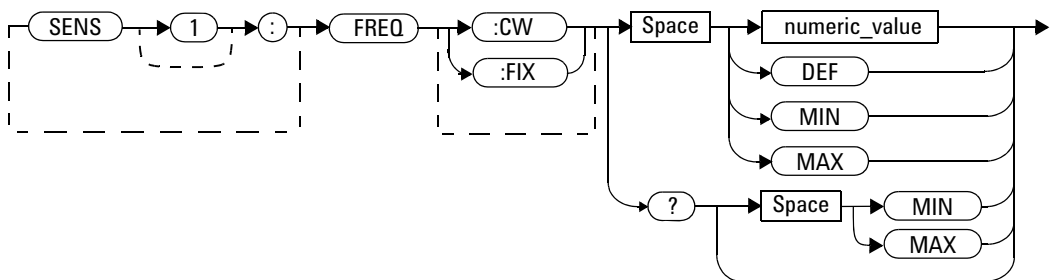
### Query example

`CORR:GAIN2?`      *Queries the current channel offset setting.*

## [SENSe[1]:]FREQuency[:CW | :FIXed] <numeric\_value>

This command is used to enter the frequency of the RF signal to measure. If the frequency does not correspond directly to a frequency in the sensor calibration table, the N432A calculates the calibration factor using linear interpolation if [SENSe[1]:]CORREction:CSET:STATe is ON.

### Syntax



### Parameter

Item	Description/Default	Range of values
numeric_value	A numeric value for the frequency <ul style="list-style-type: none"> <li>• DEF: The default value is 50 MHz</li> <li>• MIN: 1 kHz</li> <li>• MAX: 1000.0 GHz</li> </ul> The default unit is Hz	1 kHz to 1000.0 GHz <sup>[1]</sup> DEF MIN MAX

[1] The following measurement units can be used:

- Hz
- kHz ( $10^3$ )
- MHz ( $10^6$ )
- GHz ( $10^9$ )

### Example

FREQ 500kHz     *This command sets the frequency to 500 kHz.*

### Reset condition

On reset, the frequency is set to 50 MHz.

### Query

[SENSE[1]:]FREQUENCY[:CW|:FIXed]? [MIN|MAX]

This query returns the current frequency setting or the values associated with MIN and MAX. The unit in which the results are returned is Hz.

### Query example

FREQ?     *Queries the current frequency setting.*



## 13 SERvice Subsystem

SERvice Subsystem	334
SERvice:BACKlight:BRIGhtness <numeric_value>	336
SERvice:BIST:CALibrator <boolean>	338
SERvice:BIST:VRF <boolean>	339
SERvice:BIST:VCOM <boolean>	340
SERvice:BIST:V0 <boolean>	341
SERvice:CALibrator:ADJust:COUR <numeric_value>	342
SERvice:CALibrator:ADJust:FINE <numeric_value>	344
SERvice:DISPlay:BSCReen <boolean>	346
SERvice:DISPlay:BSCReen:SECure:ACTivation <numeric_value>	347
SERvice:DISPlay:BSCReen:SECure:DEACTivation <numeric_value>	348
SERvice:FAN:FULL <boolean>	350
SERvice:LAN:PHOStname	351
SERvice:SECure:ERASe	352
SERvice:SENSor[1]:TYPE?	353
SERvice:SNUMber?	354
SERvice:STATe <boolean>	355
SERvice:VERSion:PROCCessor?	356
SERvice:VERSion:SYSTem?	357

This chapter explains how the `SERvice` command subsystem is used to configure service-related parameters for the N432A.



## SERVICE Subsystem

The SERVICE command subsystem allows you to acquire service-related information of the N432A as well as configure the parameters for servicing the N432A.

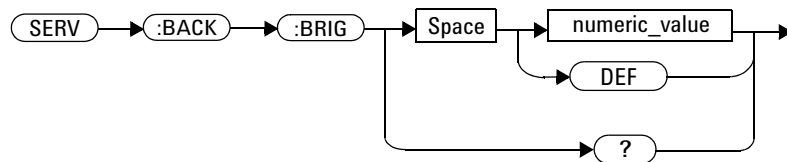
Keyword	Parameter form	Note	Page
SERVICE			
:BACKlight			
:BRiGhtness	<numeric_value>		page 336
:BIST			
:CALibrator	<boolean>		page 338
:VRF	<boolean>		page 339
:VCOM	<boolean>		page 340
:V0	<boolean>		page 338
:CALibrator			
:ADJ			
:COUR	<numeric_value>		page 342
:FINE	<numeric_value>		page 344
:DISP			
:BScreen		[no query]	page 346
:SECure			
:ACTivation	<string>		page 347
:DEACTivation	<string>		page 348
:FAN			
:FULL	<boolean>		
:LAN			
:PHOSername			page 351
:SECure			
:ERASe			page 352
:SENSor[1]			
:TYPE?		[query only]	page 353

<b>Keyword</b>	<b>Parameter form</b>	<b>Note</b>	<b>Page</b>
:SNUMber?		[query only]	<a href="#">page 354</a>
:STATe	<boolean>		<a href="#">page 355</a>
:VERSion			
:PROcEssor?		[query only]	<a href="#">page 356</a>
:SYSTem?		[query only]	<a href="#">page 357</a>

## SERVICE:BACKlight:BRIGhtness <numeric\_value>

This command sets the intensity of the front panel backlight.

### Syntax



### Parameter

Item	Description/Default	Range of values
numeric_value	Intensity level of the backlight. DEF: The default value is 80	15 to 100 DEF

### Example

SERV:BACK:BRIG 70     *This command sets the backlight intensity to 70.*

### Reset condition

On reset, the backlight intensity level is 80.



## Query

```
SERvice:BACKlight:BRIGhtness?
```

This query returns the current intensity level of the front panel backlight.

## Query example

```
SERV:BACK:BRIG?      Queries the backlight intensity level.
```

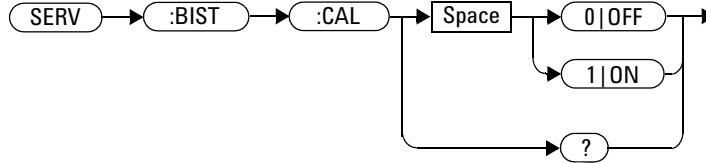
## Error messages

- If the backlight is not turned on, error -310, “System error” occurs.
- If the intensity level is set outside the valid range, error -222, “Data out of range” occurs.

## SERVICE:BIST:CALibrator <boolean>

This command enables or disables the reference calibrator self-test during power up. It can be used to disable the self-test if it incorrectly indicates a failure. If a load, for example, a sensor, is connected to the reference calibrator port, this could cause the self-test to fail. Also, if it fails the self-test, a pop-up message will be displayed for 5 seconds, stating “*If Ref Calibrator test fails disconnect any load attached to it and re-try test*”.

### Syntax



### Example

`SERV:BIST:CAL OFF`      *This command disables the reference calibrator self-test during power up.*

### Query

`SERVICE:BIST:CALibrator?`

This query enters a 1 or 0 into the output buffer indicating the status of the reference calibrator self-test:

- 1 is returned if the self-test is enabled.
- 0 is returned if the self-test is disabled.

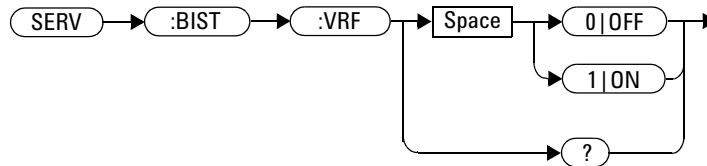
### Query example

`SERV:BIST:CAL?`      *Queries whether the reference calibrator self-test is enabled or disabled.*

## SERvice:BIST:VRF <boolean>

This command enables or disables the  $V_{RF}$  self-test.

### Syntax



### Reset condition

On reset, the  $V_{RF}$  self-test is enabled.

### Example

`SERvice:BIST:VRF ON`      *This command enables the  $V_{RF}$  self-test.*

### Query

`SERvice:BIST:VRF?`

This query enters a 1 or 0 into the output buffer indicating the status of the  $V_{RF}$  self-test:

- 1 is returned if the self-test is enabled.
- 0 is returned if the self-test is disabled.

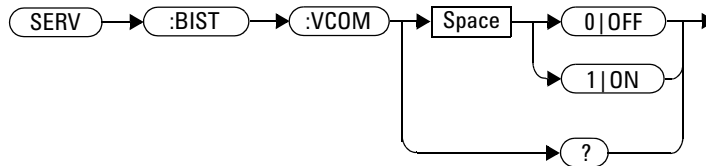
### Query example

`SERvice:BIST:VRF?`      *Queries whether the  $V_{RF}$  self-test is enabled or disabled.*

## SERVICE:BIST:VCOM <boolean>

This command enables or disables the  $V_{COMP}$  self-test.

### Syntax



### Reset condition

On reset, the  $V_{COMP}$  self-test is enabled.

### Example

`SERV:BIST:VCOM OFF` *This command disables the  $V_{COMP}$  self-test.*

### Query

`SERVICE:BIST:VCOM?`

This query enters a 1 or 0 into the output buffer indicating the status of the  $V_{COMP}$  self-test:

- 1 is returned if the self-test is enabled.
- 0 is returned if the self-test is disabled.

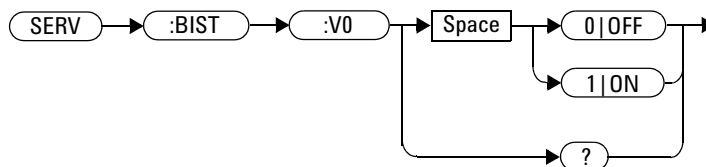
### Query example

`SERV:BIST:VCOM?` *Queries whether the  $V_{COMP}$  self-test is enabled or disabled.*

## SERvice:BIST:V0 <boolean>

This command enables or disables the  $V_0$  self-test.

### Syntax



### Reset condition

On reset, the  $V_0$  self-test is enabled.

### Example

SERV:BIST:V0 ON      *This command enables the  $V_0$  self-test.*

### Query

SERVice:BIST:V0?

This query enters a 1 or 0 into the output buffer indicating the status of the  $V_0$  self-test:

- 1 is returned if the self-test is enabled.
- 0 is returned if the self-test is disabled.

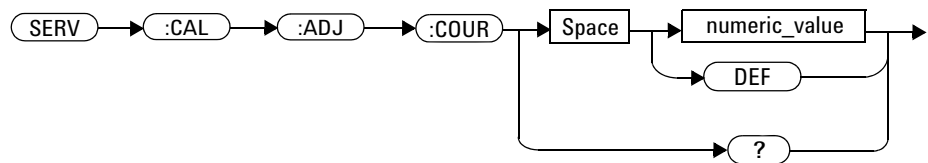
### Query example

SERV:BIST:V0?      *Queries whether the  $V_0$  self-test is enabled or disabled.*

## SERVICE:CALibrator:ADJust:COUR <numeric\_value>

This command adjusts the 1 mW reference calibrator output in the coarse scale.

### Syntax



### Parameter

Item	Description/Default	Range of values
numeric_value	Adjusts the 1 mW power reference level increment coarse by 1	0 to 1023 (Unsigned Integer 16 format)

### Example

SERV:CAL:ADJ:COUR 1000 *This command sets the power reference level to 1000.*

### Reset condition

On reset, the power reference level is not affected.

## Query

```
SERvice:CALibrator:ADJust:COUR?
```

This query returns the current power reference level in the unsigned integer 16 format.

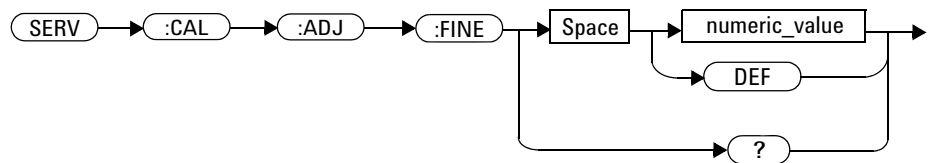
## Query example

```
SERV:CAL:ADJ:COUR? Returns the reference calibrator power level in  
unsigned integer 16.
```

## SERVICE:CALibrator:ADJust:FINE <numeric\_value>

This command adjusts the 1 mW reference calibrator output in the fine scale.

### Syntax



### Parameter

Item	Description/Default	Range of values
numeric_value	Adjusts the 1 mW power reference level increment fine by 1	0 to 1023 (Unsigned Integer 16 format)

### Example

`SERV:CAL:ADJ:FINE 500`      *This command sets the power reference level to 500.*

### Reset condition

On reset, the power reference level is not affected.



## Query

```
SERvice:CALibrator:ADJust:FINE?
```

This query returns the current power reference level in the unsigned integer 16 format.

## Query example

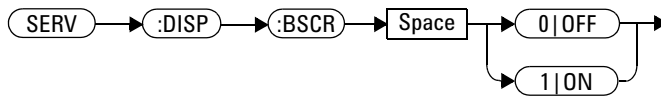
```
SERV:CAL:ADJ:FINE?
```

*Returns the Reference Calibrator power level in unsigned integer 16.*

## SERVICE:DISPLAY:BSCREEN <boolean>

This command enables or disables the blank screen feature of the front panel display.

### Syntax



### Examples

```
SERV:DISPLAY:BSCR ON|1
```

*This command blanks the screen.*

```
SERV:DISPLAY:BSCR OFF|0
```

*This command restores the screen.*

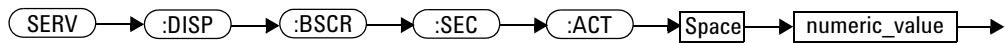
### Reset condition

On reset, the blank screen feature is disabled.

## SERvice:DISPlay:BSCReen:SECure:ACTivation <numeric\_value>

This command activates the secure blank feature using a 6-digit user-defined password.

### Syntax



### Parameter

Type	Description/Default	Range of values
numeric_value	A 6-digit password to activate the secure blank feature	"000000" to "999999"

### Example

`SERV:DISP:BSCR:SEC:ACT "123456"` *This command activates the secure blank feature.*

### Remark

A double quote "xxxxxx" is only applicable for remote operations.

### Error messages

- If the secure blank feature is already activated, error -221, "Settings conflict" occurs.
- If the password is not made up of six digits, error -222, "Data out of range; password is not 6 digits; please try again" occurs.

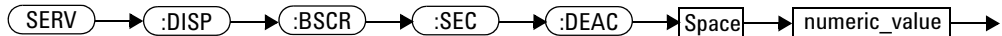
## SERVICE:DISPLAY:BSCREEN:SECURE:DEACTIVATION <numeric\_value>

This command deactivates the secure blank feature with the 6-digit user-defined password that was used to activate the secure blank feature. You are allowed up to three attempts to deactivate using the correct password; failing which you will only be allowed to retry deactivation after two hours.

### NOTE

The retry timer is a 2-hour duration with the N432A turned on. If the N432A is turned off at any time within this duration, the retry timer will be restarted.

### Syntax



### Parameter

Type	Description/Default	Range of values
numeric_value	A 6-digit password to deactivate the secure blank feature	"000000" to "999999"

### Remark

This command is only effective when the secure blank feature is activated.

## Example

SERV:DISP:BSCR:SEC:DEAC "123456"      *This command restores the screen if the correct 6- digit password has been entered.*

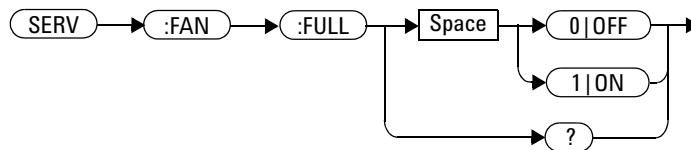
## Error messages

- If the secure blank feature is not activated, error -221, "Settings conflict" occurs.
- If the deactivation attempt password does not match the activation password, error -120, "Numeric data error; password is incorrect; please try again" occurs.
- If the number of attempts has been exceeded and the retry timer has not expired, error -120, "Numeric data error; password is incorrect; please try again" occurs.

## SERVICE:FAN:FULL <boolean>

This command sets the fan controller to run at full speed. At the first power up, the default state is ON. If the state is set to OFF and power is cycled, the state will be saved. On the next power cycle, the state will be set to OFF.

### Syntax



### Examples

SERV:FAN:FULL 1	<i>This command sets the fan controller to run at full speed.</i>
SERV:FAN:FULL OFF	<i>This command turns off the fan controller.</i>

### Query

```
SERVICE:FAN:FULL?
```

This query returns the current state of the fan controller.

### Query example

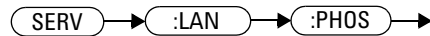
SERV:FAN:FULL?	<i>Queries whether the fan controller is turned on or off.</i>
----------------	--

## SERVice:LAN:PHOSname

This command presets the LAN host name to its default value.

The format of the default value is “A-” + model number + “-” + last five digits of the serial number; for example: A-N432A-01346

### Syntax



### Example

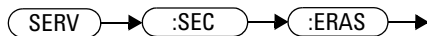
SERV:LAN:PHOS

*The command presets the LAN host name to its default value.*

## SERVICE:SECure:ERASe

This command erases the N432A memory, for example, before you return the N432A to Agilent Technologies for repair or calibration, of all data stored in it which includes the save/recall and last power-on states.

### Syntax



### Example

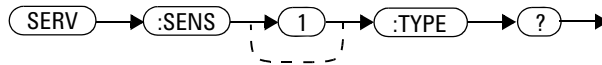
SERV:SEC:ERAS      *The command erases the N432A memory.*



## SERvice:SENSor[1]:TYPE?

This query identifies the type of sensor connected to the N432A input channel A.

### Syntax



### Example

SERV:SENS:TYPE?      *This query returns the connected sensor type.*

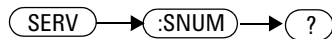
### Error message

If there is no sensor connected, error -241, "Hardware missing" occurs.

## SERVICE:SNUMBER?

This query returns the N432A serial number in the form of MY12345678.

### Syntax



### Example

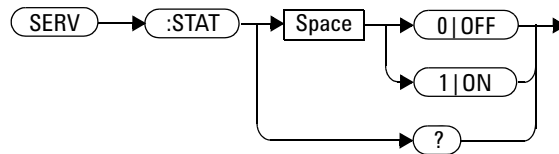
SERV:SNUM?

*This query returns the N432A serial number.*

## SERvice:STATe <boolean>

This command enables or disables the warm start feature. The warm start feature allows you to retain the N432A current states and settings upon power cycle or in the event of interrupted power. The default state is ON.

### Syntax



### Example

`SERvice:STAT 1`      *This command enables the warm start feature.*

### Remark

This parameter is not affected by any reset operation and can only be changed by direct user access.

### Query

`SERvice:STATe?`

This query returns the current state of the warm start feature.

### Query example

`SERvice:STAT?`      *Queries whether warm start is enabled or disabled.*

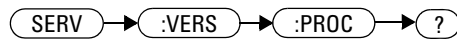
### Error message

If there is no non-volatile RAM on board, error -310, "System error" occurs.

## SERVICE:VERSION:PROCESSOR?

This query returns the current processor board revision version.

### Syntax



### Example

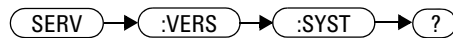
SERV:VERS:PROC?

*Queries the processor board revision version of the N432A.*

## SERVice:VERSion:SYSTem?

This query returns the current system version number.

### Syntax



### Example

SERV:VERS:SYST?      *Queries the system version number of the N432A.*





## 14 STATus Subsystem

STATus Subsystem	360
Status Register Set Commands	362
Device Status Register Set	366
Operation Status Registers Sets	368
STATus:OPERation	369
STATus:OPERation:CALibrating[:SUMMARY]	370
STATus:OPERation:LLFail[:SUMMARY]	371
STATus:OPERation:MEASuring[:SUMMARY]	372
STATus:OPERation:TRIGger[:SUMMARY]	373
STATus:OPERation:ULFail[:SUMMARY]	374
STATus:PRESet	375
Questionable Status Register Sets	376
STATus:QUEStionable	377
STATus:QUEStionable:CALibration[:SUMMARY]	378
STATus:QUEStionable:POWer[:SUMMARY]	379
Status Block Diagram	381

This chapter explains how the *STATus* command subsystem enables you to examine the status of the N432A by monitoring the “Device Status Register”, “Operation Status Register”, and “Questionable Status Register”.



## STATus Subsystem

**NOTE**

Refer to “[Status Block Diagram](#)” on page 381 to view the relationship between various registers in the N432A SCPI status system.

The STATus command subsystem enables you to examine the status of the N432A by monitoring the following status registers:

- Device status register
- Operation status register
- Questionable status register

The contents of these and other registers in the N432A are determined by one or more status registers.

The following table summarizes the effects of various commands and events on the status registers:

**Table 14-1** Commands and events affecting the status registers

Status register	*RST	*CLS	Power on	STATus: PRESet
SCPI Transition Filters (NTR and PTR registers)	none	none	preset	preset
SCPI Enable Registers	none	none	preset	preset
SCPI Event Registers	none	clear	clear	none
SCPI Error/Event Queue Enable	none	none	preset	preset
SCPI Error/Event Queue	none	clear	clear	none
IEEE-488.2 Registers ESE SRE	none	none	clear	none
IEEE-488.2 Registers ESR STB	none	clear	clear	none



The contents of the status registers are examined using the following status register set commands:

```
:CONDition?
:ENABle <NRf>|<non-decimal numeric>
[:EVENT?]
:NTRAnsition <NRf>|<non-decimal numeric>
:PTRAnsition <NRf>|<non-decimal numeric>
```

Each of these can be used to examine any of the following status registers:

```
STATus:DEVIce (page 366)
STATus:OPERation (page 369)
STATus:OPERation:CALibrating[:SUMMARY] (page 370)
STATus:OPERation:LLFail[:SUMMARY] (page 371)
STATus:OPERation:MEASuring[:SUMMARY] (page 372)
STATus:OPERation:TRIGger[:SUMMARY] (page 373)
STATus:OPERation:ULFail[:SUMMARY] (page 374)
STATus:QUESTionable (page 377)
STATus:QUESTionable:CALibration[:SUMMARY] (page 378)
STATus:QUESTionable:POWER[:SUMMARY] (page 379)
```

## Examples

Using the `:CONDition?` query to examine the `STATus:DEVIce` register:

```
STATus:DEVIce:CONDition?
```

Using the `:NTRAnsition` command to examine the `STATus:OPERation:MEASuring[:SUMMARY]` register:

```
STATus:OPERation:MEASuring[:SUMMARY]:NTRAnsition
```

## Status Register Set Commands

This section describes the five status register set commands. Each can be used to examine all of the eleven status registers listed on [page 361](#).

To apply a command to a specific register, prefix the command with the name of the appropriate register. For example, to apply the :ENABle command to the STATus:QUEStionable register, use the following command:

```
STATus:QUEStionable:ENABle
```

The status register set commands detailed in this section are:

Keyword	Parameter form	Note	Page
:CONDition?		[query only]	<a href="#">page 362</a>
:ENABle	<NRf>   <non-decimal numeric>		<a href="#">page 363</a>
[ :EVENT? ]		[query only]	<a href="#">page 363</a>
:NTRansition	<NRf>   <non-decimal numeric>		<a href="#">page 364</a>
:PTRansition	<NRf>   <non-decimal numeric>		<a href="#">page 365</a>

### :CONDition?

This query returns a 16-bit decimal-weighted number representing the bits set in the Condition register of the SCPI register set you require to control. The return format is <NR1> in the range of 0 to 32767 ( $2^{15}-1$ ). The contents of the Condition register remain unchanged after they are read.

#### Syntax

```
.. → :COND → ? →
```

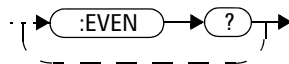
## [ :EVENT? ]

This query returns a 16-bit decimal-weighted number representing the bits set in the Event register of the SCPI register set you require to control. The return format is <NR1> in the range of 0 to 32767 ( $2^{15}-1$ ). This query clears all bits in the register to 0.

**NOTE**

[ :EVENT? ] is the default command if the STATus commands are not accompanied by any of the status register set commands (:COND?, :ENAB, :NTR, and :PTR).

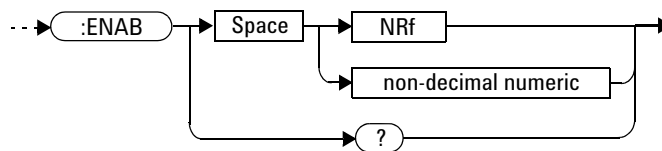
**Syntax**



## :ENABLE <NRf> | <non-decimal numeric>

This command sets the Enable register of a particular SCPI register set you require to control. The parameter value, when rounded to an integer and expressed in base 2 has its first 15 bits written into the Enable register of the specific SCPI register set. The last bit (bit 15) is always set to 0.

**Syntax**



**Parameters**

Item	Description	Range of values
NRf	The value used to set the Enable register	0 to $2^{15}-1$
non-decimal numeric		

**Query**

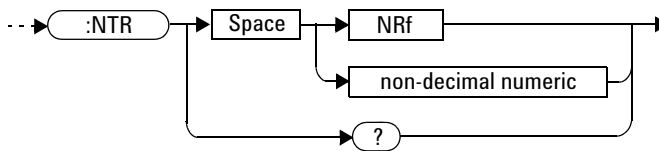
:ENABle?

This query returns a 15-bit decimal-weighted number representing the contents of the Enable register of the SCPI register set being queried. The return format is <NR1> in the range of 0 to 32767 ( $2^{15}-1$ ).

**:NTRansition <NRf> | <non-decimal numeric>**

This command sets the Negative Transition register of the SCPI register set you require to control. The parameter value, when rounded to an integer and expressed in base 2 has its first 15 bits written into the Negative Transition register of the specific SCPI register set. The last bit (bit 15) is always set to 0.

**Syntax**



**Parameters**

Item	Description	Range of values
NRf	The value used to set the NTR register	0 to $2^{15}-1$
non-decimal numeric		

**Query**

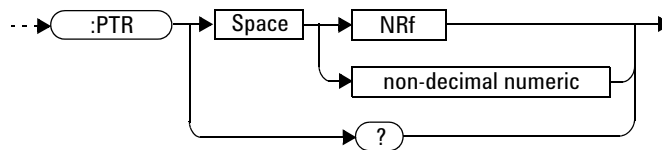
:NTRansition?

This query returns a 15-bit decimal-weighted number representing the contents of the Negative Transition register of the SCPI register set being queried. The return format is <NR1> in the range of 0 to 32767 ( $2^{15}-1$ ).

## :PTRansition <NRf> | <non-decimal numeric>

This command is used to set the Positive Transition register of the SCPI register set you require to control. The first 15 bits of the input parameter are written into the Positive Transition register of the specific SCPI register set. The last bit (bit 15) is always set to 0.

### Syntax



### Parameters

Item	Description	Range of values
NRf	The value used to set the PTR register	0 to $2^{15}-1$
non-decimal numeric		

### Query

:PTRansition?

This query returns a 15-bit decimal-weighted number representing the contents of the Positive Transition register of the SCPI register set being queried. The return format is <NR1> in the range of 0 to 32767 ( $2^{15}-1$ ).

## Device Status Register Set

The device status register set contains the information which provides the device status information. The contents of the individual registers of this register set can be accessed by appending the commands listed in “[Status Register Set Commands](#)”.

The following descriptions detail the SCPI register you require to control and some examples on the register set commands.

The one device status register set is:

STATus:DEvice

The following bits in this register are used by the N432A:

Bit	Weight/Decimal value	Definition
0	1	Not used
1	2	Channel A sensor connected
2 to 11	–	Not used
12	4096	Over temperature
13	8192	Fan failure
14	16384	Front panel keypress
15	32767	Always 0

The channel A sensor connected bit (bit 1), when queried by STATus:DEvice:CONDition?, is set to:

- 1, when a thermistor sensor is connected.
- 0, when no thermistor sensor is connected.

The channel A sensor connected bit (bit 1), when queried by `STATus:DEVIce:EVENT?`, indicates whether a thermistor sensor has been connected or disconnected depending on the state of the corresponding bits of `STATus:DEVIce:NTRansition` and `STATus:DEVIce:PTRansition`. If the corresponding bit in:

- `STATus:DEVIce:NTRansition` is 1, then bit 1 is set when a sensor is disconnected.
- `STATus:DEVIce:PTRansition` is 1, then bit 1 is set when a sensor is connected.

**NOTE**

Querying `STATus:DEVIce:EVENT?` clears the Device Status Event register.

---

The front panel keypress bit (bit 14), when queried by `STATus:DEVIce:EVENT?` indicates whether any front panel keys have been pressed since power up or since you last queried the device status register. This bit ignores the `:NTRansition`, and `:PTRansition` registers and a `:CONDition?` query always returns a 0.

## Operation Status Registers Sets

The following register sets contain information which is part of the N432A normal operation. The contents of the individual registers of these register sets can be accessed by appending the commands listed in “[Status Register Set Commands](#)”.

The seven operation status register sets are:

STATUS:OPERation

STATUS:OPERation:CALibrating[:SUMMARY]

STATUS:OPERation:LLFail[:SUMMARY]

STATUS:OPERation:MEASuring[:SUMMARY]

STATUS:OPERation:TRIGger[:SUMMARY]

STATUS:OPERation:ULFail[:SUMMARY]

Further information on these register sets is provided on the following pages.



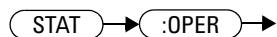
## STATus:OPERation

The operation status register set contains conditions which are a part of the N432A operation as a whole.

The following bits in the register set are used by the N432A:

Bit	Weight/Decimal value	Definition
0	1	Not used
1	2	Self-test started
2 to 3	–	Not used
4	16	MEASure summary
5	32	Waiting for TRIGger summary
6 to 10	–	Not used
11	2048	Lower limit fail summary
12	4096	Upper limit fail summary
13 to 15	–	Not used (bit 15 is always 0)

### Syntax



## STATus:OPERation:CALibrating[:SUMMary]

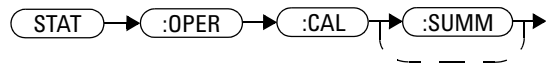
The operation status calibration summary register contains information on the zeroing status of the N432A.

The following bit in the register is used by the N432A:

Bit	Weight/Decimal value	Definition
0	1	Not used
1	2	Channel A CALibration (zeroing) status
2 to 15	–	Not used

This bit is set at the beginning of zeroing (CALibration:ZERO:AUTO ONCE) and cleared at the end of zeroing.

### Syntax



## STATus:OPERation:LLFail[:SUMM]ary]

The operation status lower limit fail summary register contains information on the lower limit fail status of the N432A.

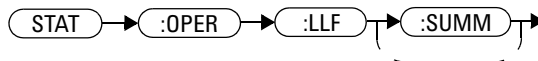
The following bits in the register are used by the N432A:

Bit	Weight/Decimal value	Definition
0 to 2	–	Not used
3	8	Upper window upper measurement LLFail status
4	16	Lower window upper measurement LLFail status
5	32	Upper window lower measurement LLFail status
6	64	Lower window lower measurement LLFail status
7 to 15	–	Not used

The appropriate bits are set if a window lower limit test fails.

These bits are cleared if a measurement is made and the test is enabled and passes.

### Syntax



## STATus:OPERation:MEASuring[:SUMM]ary]

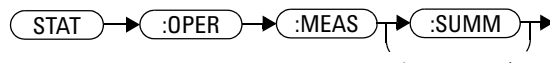
The operation status measure summary register contains information on the measurement status of the N432A.

The following bit in the register is used by the N432A:

Bit	Weight/Decimal value	Definition
0	1	Not used
1	2	Channel A MEASurement status
2 to 15	–	Not used

This bit is set when the N432A is taking a measurement and cleared when the measurement has completed.

### Syntax



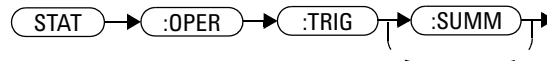
## STATus:OPERation:TRIGger[:SUMM]ary

The operation status trigger summary register contains information on the triggering status of the N432A.

The following bit in the register is used by the N432A:

Bit	Weight/Decimal value	Definition
0	1	Not used
1	2	Channel A waiting for TRIGger status
2 to 15	–	Not used

### Syntax



## STATUS:OPERation:ULFail[:SUMMARY]

The operation status upper limit fail summary register contains information on the upper limit fail status of the N432A.

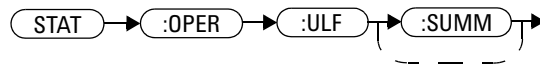
The following bits in the register are used by the N432A:

Bit	Weight/Decimal value	Definition
0 to 2	–	Not used
3	8	Upper window upper measurement ULFail status
4	16	Lower window upper measurement ULFail status
5	32	Upper window lower measurement ULFail status
6	64	Lower window lower measurement ULFail status
7 to 15	–	Not used

The appropriate bits are set if a window upper limit test fails.

These bits are cleared if a measurement is made and the test is enabled and passes.

### Syntax



## STATus:PRESet

PRESet sets a number of the status registers to their preset values as shown below; all other registers are unaffected. Bit 15 is always 0.

Register	Filter/Enable	PRESet value
OPERation	ENABle	all zeros
	PTR	all ones
	NTR	all zeros
QUEStionable	ENABle	all zeros
	PTR	all ones
	NTR	all zeros
All others	ENABle	all ones
	PTR	all ones
	NTR	all zeros

### Syntax



## Questionable Status Register Sets

The questionable status register sets contain information which gives an indication of the quality of the data produced by the N432A. The contents of the individual registers in these register sets may be accessed by appending the commands listed in “[Status Register Set Commands](#)”.

The three questionable status register sets are:

```
STATus:QUESTionable
```

```
STATus:QUESTionable:CALibration[:SUMMARY]
```

```
STATus:QUESTionable:POWER[:SUMMARY]
```



## STATus:QUEStionable

The questionable status register set contains bits that indicate the quality of various aspects of signals processed by the N432A.

The following bits in the register set are used by the N432A:

Bit	Weight/Decimal value	Definition
0 to 2	–	Not used
3	8	POWer summary
4 to 7	–	Not used
8	256	CALibration summary
9	512	Power-on self-test
10 to 15	–	Not used (bit 15 is always 0)

Bit 3 is set by the logical OR outputs of the `STATus:QUEStionable:POWer:SUMMARY` register.

Bit 8 is set by the logical OR outputs of the `STATus:QUEStionable:CALibration:SUMMARY` register.

Bit 9 is set if the power-on self-test fails, and cleared if it passes.

### Syntax



## STATus:QUEStionable:CALibration[:SUMMARY]

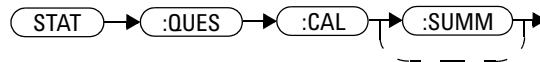
The questionable status calibration summary register contains bits which give an indication of the quality of the data produced by the N432A due to its calibration (zeroing) status.

The following bit in the register is used by the N432A:

Bit	Weight/Decimal value	Definition
0	1	Not used
1	2	Channel A CALibration (zeroing) summary
2 to 15	–	Not used

This bit is set when error -231, “Data questionable; ZERO ERROR ChA” occurs. It is cleared when this error condition succeeds and no errors are placed on the error queue.

### Syntax



## STATus:QUEStionable:POWer[:SUMMary]

The questionable status power summary register contains bits that indicate the quality of the power data being acquired by the N432A.

The following bits in the register are used by the N432A:

Bit	Weight/Decimal value	Definition
0	1	Not used
1	2	Channel A POWer
2	4	Not used
3	8	Upper window upper measurement POWer
4	16	Lower window upper measurement POWer
5	32	Channel A Please Zero
6	64	Not used
7	128	Upper window lower measurement POWer
8	256	Lower window lower measurement POWer
9 to 15	–	Not used

Bit 1 is set when any of the following errors occur:

- Error -231,“Data questionable; Input Overload”
- Error -231,“Data questionable; Input Overload ChA”

Bits 3 is set when the following errors occur:

- Error -230,“Data corrupt or stale”
- Error -231,“Data questionable; Upper window log error”

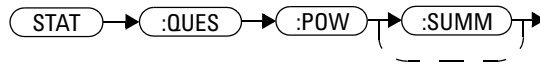
Bit 4 is set when the following errors occur:

- Error -230,“Data corrupt or stale”
- Error -231,“Data questionable; Lower window log error”

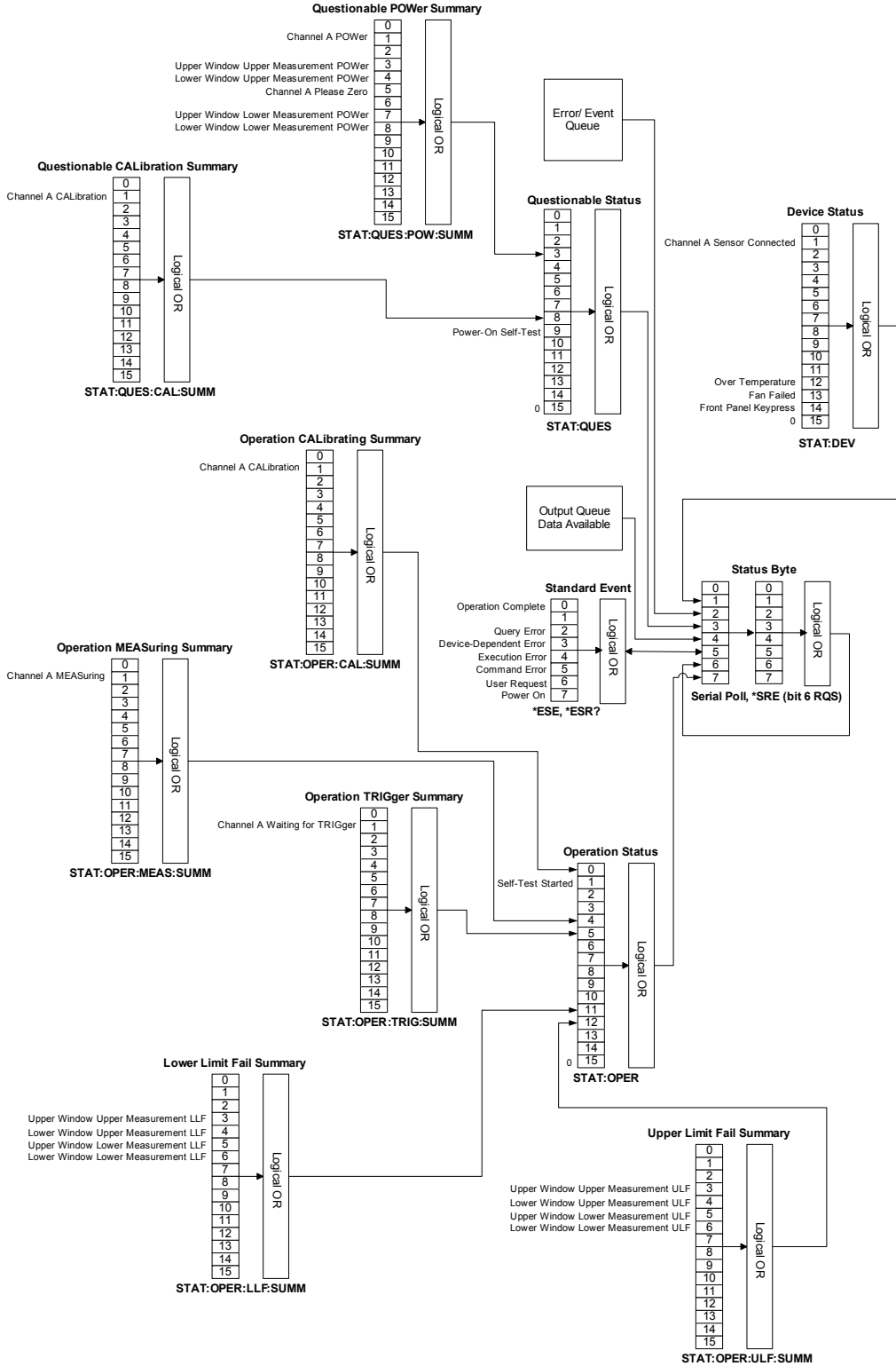
Bit 5 is set when channel A requires zeroing.

These bits are cleared when no errors or events are detected by the N432A during a measurement covering the causes given for it to set.

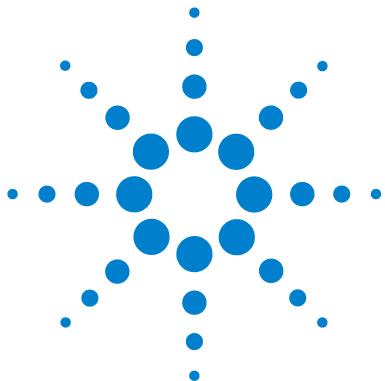
## Syntax



# Status Block Diagram







## 15 SYSTEM Subsystem

SYSTEM Subsystem	384
SYSTEM:COMMunicate:GPIB[:SELF]:ADDRESS <numeric_value>	386
SYSTEM:COMMunicate:LAN:AIP[:STATe] <boolean>	388
SYSTEM:COMMunicate:LAN:CURRent:ADDRESS?	389
SYSTEM:COMMunicate:LAN:CURRent:DGATeway?	390
SYSTEM:COMMunicate:LAN:CURRent:DNAMe?	391
SYSTEM:COMMunicate:LAN:CURRent:SMASK?	392
SYSTEM:COMMunicate:LAN:ADDRESS <character_data>	393
SYSTEM:COMMunicate:LAN:DGATeway <character_data>	394
SYSTEM:COMMunicate:LAN:DHCP[:STATe] <boolean>	395
SYSTEM:COMMunicate:LAN:DNAMe <character_data>	396
SYSTEM:COMMunicate:LAN:HNAME <character_data>	397
SYSTEM:COMMunicate:LAN:MAC?	398
SYSTEM:COMMunicate:LAN:REStart	399
SYSTEM:COMMunicate:LAN:SMASK <character_data>	400
SYSTEM:COMMunicate:TCPIP:CONTRol?	401
SYSTEM:DISPlay:BMP?	402
SYSTEM:ERRor?	403
SYSTEM:LOCal	411
SYSTEM:PRESet	412
SYSTEM:REMOte	416
SYSTEM:RWLock	417
SYSTEM:VERSion?	418

This chapter explains how to use the `SYSTEM` command subsystem to configure the remote settings, preset the N432A, return error numbers and messages from the N432A, and query the SCPI version.



## SYSTem Subsystem

The SYSTem command subsystem is used to perform system-related configurations such as:

- Setting the GPIB address
- Setting the LAN settings
- Presetting the N432A
- Returning error numbers and messages from the N432A
- Querying the SCPI version

Keyword	Parameter form	Note	Page
SYSTem			
:COMMunicate			
:GPIB			
[:SELF]			
:ADDRess	<numeric_value>		<a href="#">page 386</a>
:LAN			
:AIP			
[:STATe]	<boolean>		<a href="#">page 388</a>
:CURRent			
:ADDRess?		[query only]	<a href="#">page 389</a>
:DGATeway?		[query only]	<a href="#">page 390</a>
:DNAME?		[query only]	<a href="#">page 391</a>
:SMASk?		[query only]	<a href="#">page 392</a>
:ADDRess	<character_data>		<a href="#">page 393</a>
:DGATeway	<character_data>		<a href="#">page 394</a>
:DHCP			
[:STATe]	<boolean>		<a href="#">page 395</a>
:DNAME	<character_data>		<a href="#">page 396</a>
:HNAME	<character_data>		<a href="#">page 397</a>
:MAC?		[query only]	<a href="#">page 398</a>
:REStart		[no query]	<a href="#">page 399</a>

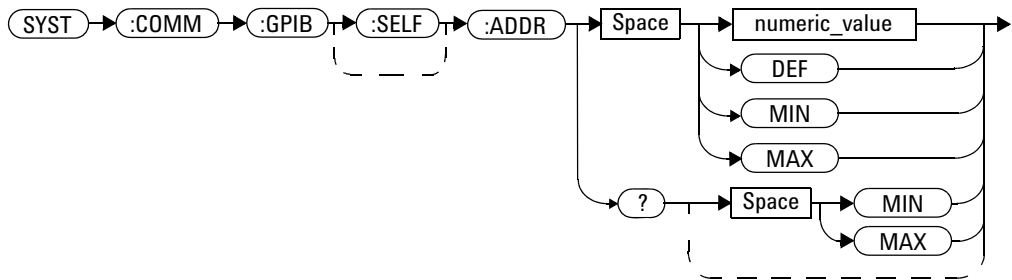


Keyword	Parameter form	Note	Page
:SMASK	<character_data>		<a href="#">page 400</a>
:TCPip			
:CONTROL?		[query only]	<a href="#">page 401</a>
:DISPLAY			
:BMP?		[query only]	<a href="#">page 402</a>
:ERROR?			<a href="#">page 403</a>
:LOCAL		[no query]	<a href="#">page 411</a>
:PRESET		[no query]	<a href="#">page 412</a>
:REMOTE		[no query]	<a href="#">page 416</a>
:RWLOCK		[no query]	<a href="#">page 417</a>
:VERSION?		[query only]	<a href="#">page 418</a>

## SYSTem:COMMunicate:GPIB[:SELF]:ADDRESS <numeric\_value>

This command sets the GPIB address of the N432A.

### Syntax



### Parameter

Item	Description/Default	Range of values
numeric_value	A numeric value for the address <ul style="list-style-type: none"> <li>• DEF: The default value is 10</li> <li>• MIN: 0</li> <li>• MAX: 30</li> </ul>	0 to 30 DEF MIN MAX

### Example

SYST:COMM:GPIB:ADDR 13 *This command sets the GPIB address to 13.*

## Query

```
SYSTem:COMMunicate:GPIB[:SELF]:ADDRess? [MIN|MAX]
```

This query returns the current setting of the GPIB address or the values associated with MIN and MAX.

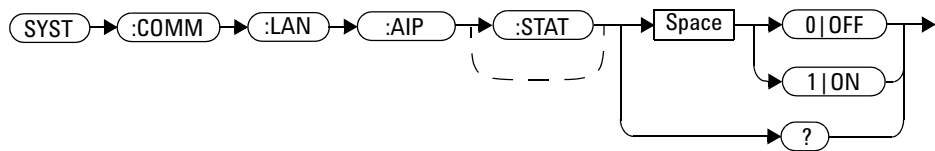
## Query example

```
SYST:COMM:GPIB:ADDR? Queries the current GPIB address.
```

## SYSTem:COMMunicate:LAN:AIP[:STATe] <boolean>

This command enables the auto IP protocol to dynamically assign the IP address when connecting to the N432A in an isolated (non-site) LAN network (for example, PC to N432A).

### Syntax



### Example

SYST:COMM:LAN:AIP ON

*This command enables auto IP.*

### Query

SYSTem:COMMunicate:LAN:AIP?

This query enters a 1 or 0 into the output buffer indicating whether auto IP is enabled or disabled.

- 1 is returned if auto IP is enabled
- 0 is returned if auto IP is disabled

### Query example

SYST:COMM:LAN:AIP?

*Queries the auto IP state.*

## SYSTem:COMMunicate:LAN:CURRent:ADDRess?

This query returns the current setting of the IP address in use by the N432A.

**NOTE**

If DHCP or auto IP is enabled successfully, then one of these IP address modes assigns the IP address, otherwise it is a static IP address.

---

### Syntax



### Example

SYST:COMM:LAN:CURR:ADDR?      *Queries the current IP address.*

## SYSTem:COMMunicate:LAN:CURRent:DGATeway?

This query returns the current setting of the LAN IP router/gateway address in use by the N432A.

**NOTE**

If DHCP or auto IP is enabled successfully, then one of these IP address modes assigns the LAN IP router/gateway address, otherwise it is a static LAN IP router/gateway address.

---

### Syntax



### Example

SYST:COMM:LAN:CURR:DGAT? *Queries the current LAN IP router/gateway address.*

## SYSTem:COMMunicate:LAN:CURRent:DNAMe?

This query returns the current setting of the LAN domain name in use by the N432A.

**NOTE**

If DHCP or auto IP is successfully enabled, then one of these IP address modes assigns the LAN domain name, otherwise it is a static LAN domain name.

---

### Syntax



### Example

SYST:COMM:LAN:CURR:DNAM?      *Queries the current LAN domain name.*

## SYSTem:COMMunicate:LAN:CURRent:SMASk?

This query returns the current setting of the LAN subnet mask in use by the N432A.

**NOTE**

If DHCP or auto IP is successfully enabled, then one of these IP address modes assigns the LAN subnet mask, otherwise it is a static LAN subnet mask.

---

### Syntax



### Example

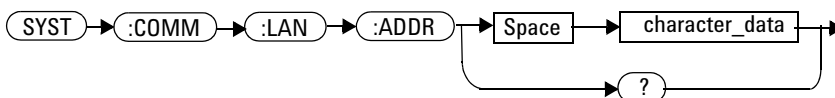
SYST:COMM:LAN:CURR:SMAS?      *Queries the current LAN subnet mask.*



## SYSTem:COMMunicate:LAN:ADDRess <character\_data>

This command sets the LAN (IP) address of the N432A.

### Syntax



### Parameter

Item	Description	Range of values
character_data	Numeric character values for the address. Up to 15 characters, formatted as A.B.C.D where each A, B, C, and D = 0 to 255	0 to 255 (no embedded spaces)

### Example

SYST:COMM:LAN:ADDR `130.015.156.255`      *This command sets the LAN IP address to 130.015.156.255.*

### Query

SYSTem:COMMunicate:LAN:ADDRess?

This query returns the current setting of the LAN address.

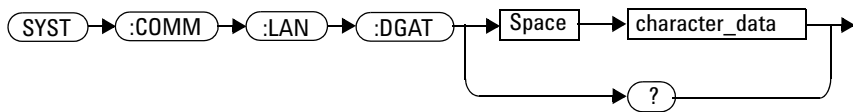
### Query example

SYST:COMM:LAN:ADDR?      *Queries the LAN IP address setting.*

## SYSTem:COMMunicate:LAN:DGATeway <character\_data>

This command sets the LAN IP router/gateway address for the N432A.

### Syntax



### Parameter

Item	Description	Range of values
character_data	Numeric character values for the address. Up to 15 characters, formatted as A.B.C.D where each A, B, C, and D = 0 to 255	0 to 255 (no embedded spaces)

### Example

```
SYST:COMM:LAN:DGAT '130.2.6.200'
```

*This command sets the gateway address to 130.2.6.200.*

### Query

```
SYSTem:COMMunicate:LAN:DGAT?
```

This query returns the current setting of the LAN gateway address.

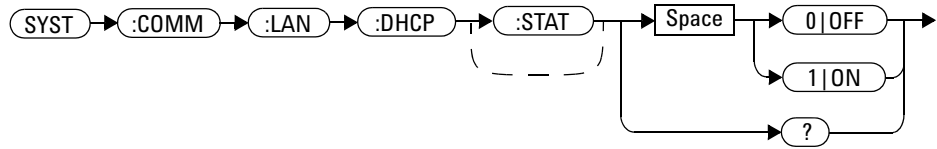
### Query example

```
SYST:COMM:LAN:DGAT? Queries the gateway address setting.
```

## SYSTem:COMMunicate:LAN:DHCP[:STATe] <boolean>

This command enables the dynamic host configuration protocol (DHCP).

### Syntax



### Example

SYST:COMM:LAN:DHCP ON *This command enables the DHCP.*

### Query

SYSTem:COMMunicate:LAN:DHCP?

This query enters a 1 or 0 into the output buffer indicating whether the DHCP is enabled or disabled.

- 1 is returned if DHCP is enabled.
- 0 is returned if DHCP is disabled.

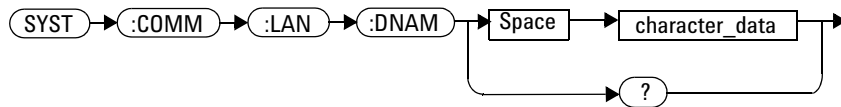
### Query example

SYST:COMM:LAN:DHCP? *Queries the current state of the DHCP.*

## SYSTem:COMMunicate:LAN:DNAMe <character\_data>

This command sets the domain name for the N432A.

### Syntax



### Parameter

Item	Description	Range of values
character_data	Character values of up to 16 characters	Maximum of 16 characters

### Example

```
SYST:COMM:LAN:DNAM 'myco.com'
```

*This command sets the domain name to “myco.com”.*

### Query

```
SYSTem:COMMunicate:LAN:DNAM?
```

This query returns the current setting of the LAN domain name.

### Query example

```
SYST:COMM:LAN:DNAM?
```

*Queries the domain name setting.*

## SYSTem:COMMunicate:LAN:HNAME <character\_data>

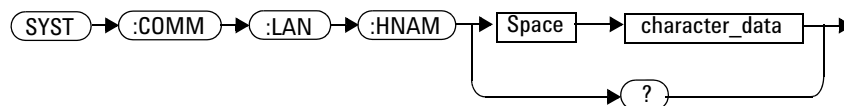
This command sets the host name for the N432A.

The factory default setting of the host name is in this format:

**A- + product number + - + suffix 5 digits of serial number**

For example, A-N432A-00204

### Syntax



### Parameter

Item	Description	Range of values
character_data	Character values of up to 15 characters	Maximum of 15 characters

### Example

SYST:COMM:LAN:HNAME 'PowerMeter1'      *This command sets the host name to PowerMeter 1.*

### Query

SYSTem:COMMunicate:LAN:HNAME?

The query returns the current setting of the LAN host name.

### Query example

SYST:COMM:LAN:HNAME?      *Queries the host name setting.*

## SYSTem:COMMunicate:LAN:MAC?

This query returns the LAN MAC address.

### Syntax



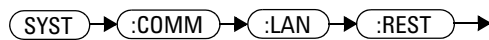
### Example

SYST:COMM:LAN:MAC?      *Queries the MAC address setting.*

## SYSTem:COMMunicate:LAN:REStart

This command restarts the N432A network stack; any LAN configuration changes can only take effect after this is performed.

### Syntax



### Example

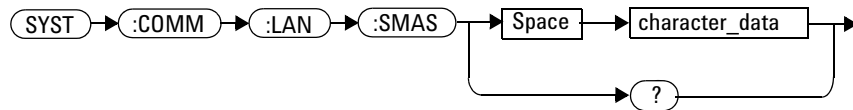
```
SYST:COMM:LAN:REST
```

*This command restarts the LAN network with a new configuration.*

## SYSTem:COMMunicate:LAN:SMASk <character\_data>

This command sets the subnet mask of the N432A.

### Syntax



### Parameter

Item	Description	Range of Values
character_data	Numeric character values for the address. Up to 15 characters, formatted as A.B.C.D where each A, B, C, and D = 0 to 255	0 to 255 (no embedded spaces)

### Example

```
SYST:COMM:LAN:SMAS
'255.255.248.0'
```

*This command sets the subnet mask to 255.255.248.0.*

### Query

```
SYSTem:COMMunicate:LAN:SMASk?
```

This query returns the current setting of the LAN subnet mask.

### Query example

```
SYST:COMM:LAN:SMAS?
```

*Queries the LAN subnet mask setting.*



## SYSTem:COMMunicate:TCPIp:CONTrol?

This query returns the socket number of control from the SCPI/socket connection. This query should be sent only when the SCPI/socket connection is used.

### Syntax



### Example

SYST:COMM:TCP:CONT?      *Queries the socket number.*

### Error message

If the connection type is incorrect, error -310, "System error" occurs.

## SYSTem:DISPlay:BMP?

This query returns the display image in the bitmap format. This query is limited to a maximum of five image returns per second.

### Syntax



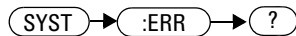
### Example

`SYST:DISP:BMP?` *Returns the display image in the bitmap format.*

## SYSTem:ERRor?

This query returns error numbers and messages from the N432A error queue. When an error is generated by the N432A, it stores an error number and its corresponding message in the error queue. One error is removed from the error queue each time this query is sent. The errors are cleared in the first-in first-out order, where the oldest errors are cleared first. To clear all errors from the error queue, execute the \*CLS command. When the error queue is empty, subsequent SYSTem:ERRor? queries will return +0, "No error". The error queue has a maximum capacity of 30 errors.

### Syntax



### Example

SYST:ERR?      *Queries the oldest error message stored in the N432A error queue.*

### Reset condition

On reset, the error queue is unaffected.

### Error message

If the error queue overflows, the last error stored in the queue (the most recent error) is replaced with -350, "Queue overflow". No additional errors are stored until you remove errors from the queue.

## Error list

-101	Invalid character
	<p>An invalid character was found in the command string. You may have inserted a character such as #, \$, or % in the command header or within a parameter.          Example: LIM:LOW 0#</p>
-102	Syntax error
	<p>Invalid syntax was found in the command string          Example: LIM:CLE:AUTO, 1 or LIM:CLE: AUTO 1</p>
-103	Invalid separator
	<p>An invalid separator was found in the command string. You may have used a comma instead of a colon, semicolon, or blank space; or you may have used a blank space instead of a comma.          Example: OUP:ROSC,1</p>
-105	GET not allowed
	<p>A Group Execute Trigger (GET) is not allowed within a command string</p>
-108	Parameter not allowed
	<p>More parameters were received than expected for the command. You may have entered an extra parameter, or added a parameter to a command that does not accept a parameter.          Example: CAL 10</p>
-109	Missing parameter
	<p>Fewer parameters were received than expected for the command. You may have omitted one or more parameters that are required for this command.          Example: AVER:COUN:VOLT</p>
-112	Program mnemonic too long
	<p>A command header was received which contained more than the maximum 12 characters allowed          Example: SENSEAVERageCOUNtVOLTage AVC8</p>

-113	Undefined header
	<p>A command was received that is not valid for the N432A. You may have misspelled the command, it may not be a valid command, or you may have selected the wrong interface. If you are using the short form of the command, remember that it may contain up to four letters.          Example: TRIG:SOU IMM</p>
-121	Invalid character in number
	<p>An invalid character was found in the number specified for a parameter value          Example: SENS:BRES R100#H</p>
-123	Exponent too large
	<p>A numeric parameter was found whose exponent was larger than 32000          Example: CAL:RCF 1E34000</p>
-124	Too many digits
	<p>A numeric parameter was found whose mantissa contained more than 255 digits, excluding leading zeros</p>
-128	Numeric data not allowed
	<p>A numeric value was received within a command which does not accept a numeric value          Example: MEM:CLE 24</p>
-131	Invalid suffix
	<p>A suffix was incorrectly specified for a numeric parameter. You may have misspelled the suffix.          Example: SENS:FREQ 200KZ</p>
-134	Suffix too long
	<p>A suffix used contained more than 12 characters          Example: SENS:FREQ 2MHZZZZZZZZZZ</p>
-138	Suffix not allowed
	<p>A suffix was received following a numeric parameter which does not accept a suffix          Example: INIT:CONT 0Hz</p>

-148	Character data not allowed
	<p>A discrete parameter was received but a character string or a numeric parameter was expected. Check the list of parameters to verify that you have used a valid parameter type.          Example: MEM:CLE CUSTOM_1</p>
-151	Invalid string data
	<p>An invalid string was received. Check to see if you have enclosed the character string in single or double quotes.          Example; MEM:CLE "CUSTOM_1</p>
-158	String data not allowed
	<p>A character string was received but is not allowed for the command. Check the list of parameters to verify that you have used a valid parameter type.          Example: LIM:STAT 'ON'</p>
-161	Invalid block data
	<p>A block data element was expected but was invalid for some reason, for example, *DDT #15FET. The 5 in the string indicates that five characters should follow, whereas in this example there are only three.</p>
-168	Block data not allowed
	<p>A legal block data element was encountered but not allowed by the N432A at this point          Example: SYST:COMM:GPIB:ADDR #15FETC?</p>
-178	Expression data not allowed
	<p>A legal expression data was encountered but not allowed by the N432A at this point          Example: FORM:BORD (5+2)</p>
-211	Trigger ignored
	<p>Indicates that &lt;GET&gt; or *TRG, or TRIG:IMM was received and recognized by the device but was ignored because the N432A was not in the wait-for-trigger state</p>

-213	<p>Init ignored</p> <p>Indicates that a request for a measurement initiation was ignored as the N432A was already initiated</p> <p>Example: INIT:CONT ON INIT</p>
-214	<p>Trigger deadlock</p> <p>TRIG:SOUR was set to HOLD or BUS and a READ? or MEASure? was attempted, expecting TRIG:SOUR to be set to IMMEDIATE</p>
-220	<p>Parameter error; Frequency list must be in ascending order</p> <p>Indicates that the frequencies entered using the MEMory:TABLE:FREQuency command are not in the ascending order</p>
-221	<p>Settings conflict</p> <p>This message occurs under a variety of conflicting conditions. The following list gives a few examples of where this error may occur:</p> <ul style="list-style-type: none"> <li>• If the READ? parameters do not match the current settings</li> <li>• Trying to clear a sensor calibration table when none is selected</li> </ul>
-222	<p>Data out of range</p> <p>A numeric parameter value is outside the valid range for the command</p> <p>Example: SENS:FREQ 2 HZ</p>
-224	<p>Illegal parameter value</p> <p>A discrete parameter was received which was not a valid choice for the command. You may have used an invalid parameter choice.</p> <p>Example: TRIG:SOUR EXT</p>
-226	<p>Lists not same length</p> <p>This occurs when SENSE:CORREction:CSET[1] CSET2:STATE is set to ON and the frequency and calibration/offset lists do not correspond in length</p>
-230	<p>Data corrupt or stale</p> <p>This occurs when a FETC? is attempted and either a reset has been received or the N432A state has changed such that the current measurement is invalidated (for example, a change of frequency setting or triggering conditions)</p>

-231	Data questionable;Input Overload The power input to the channel exceeds the maximum range of the sensor
-231	Data questionable;Input Overload ChA The power input to Channel A exceeds the maximum range of the sensor
-231	Data questionable;Lower window log error This indicates that a difference measurement in the lower window has given a negative result when the units of measurement were logarithmic
-231	Data questionable;Upper window log error This indicates that a difference measurement in the upper window has given a negative result when the units of measurement were logarithmic
-231	Data questionable;ZERO ERROR N432A zeroing failed. The most likely cause is attempting to zero when some power signal is being applied to the sensor.
-231	Data questionable;ZERO ERROR ChA N432A zeroing failed on Channel A. The most likely cause is attempting to zero when some power signal is being applied to the sensor.
-231	Data questionable;Measurement Over Range The voltage measurement failed as the measured data has exceeded the maximum range
-231	Data questionable;Measurement Under Range The voltage measurement failed as the measured data is below the minimum range
-241	Hardware missing The N432A is unable to execute the command because either no sensor is connected or it expects a compatible thermistor sensor and one is not connected



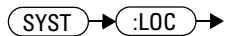
-310	System error
	This indicates that there is a problem with the N432A operation. This error may occur under certain conditions, for example, the front panel backlight is not turned on when you set the backlight intensity or there is no non- volatile RAM on board when you configure the warm start state.
-321	Out of memory
	The N432A required more memory than was available to run an internal operation
-330	Self-test Failed;
	The -330, "Self-test Failed" errors indicate that you have a problem with your N432A. Refer to the <i>N432A Thermistor Power Meter User's Guide</i> for details on how to manage your faulty N432A.
-330	Self-test Failed;Measurement Channel Fault
-330	Self-test Failed;Measurement Channel A Fault
-330	Self-test Failed;Calibrator Fault
	Refer to the <i>N432A Thermistor Power Meter User's Guide</i> if you require a description of the calibrator test
-330	Self-test Failed;ROM Check Failed
-330	Self-test Failed;RAM Check Failed
-350	Queue overflow
	The error queue is full and another error has occurred which could not be recorded
-361	Parity error in program
	The serial port receiver has detected a parity error and consequently, data integrity cannot be guaranteed
-362	Framing error in program
	The serial port receiver has detected a framing error and consequently, data integrity cannot be guaranteed

-363	<p>Input buffer overrun</p> <p>The serial port receiver has been overrun and consequently, data has been lost</p>
-410	<p>Query INTERRUPTED</p> <p>A command which sends data to the output buffer was received, but the output buffer contained data from a previous command (the previous data was not overwritten). The output buffer was cleared when power has been turned off, or after a *RST (reset) command has been executed.</p>
-420	<p>Query UNTERMINATED</p> <p>The N432A was addressed to talk (that is, to send data over the interface) but a command which sends data to the output buffer has not been received. For example, you may have executed a CONFigure command (which does not generate data) and then attempted to read data from the remote interface.</p>
-430	<p>Query DEADLOCKED</p> <p>A command was received which generates too much data to fit in the output buffer and the input buffer is also full. Command execution continues but data is lost.</p>
-440	<p>Query UNTERMINATED after indefinite response</p> <p>The *IDN? query must be the last query within a command string</p>
-101	<p>Invalid character</p> <p>An invalid character was found in the command string. You may have inserted a character such as #, \$, or % in the command header or within a parameter. Example: LIM:LOW O#</p>

## SYSTem:LOCal

This command unlocks the front panel keypad and enables the N432A to be controlled from the front panel. The status reporting line of the N432A display will show “LCL”.

### Syntax



The diagram shows the command structure: a rounded rectangle containing 'SYST' has an arrow pointing to another rounded rectangle containing ':LOC', which in turn has an arrow pointing to the right.

### Example

SYST:LOC

*This command unlocks the N432A front panel keypad and enables local front panel control.*

## SYSTem:PRESet

This command is used to preset the N432A to its default settings. The result from this command execution is similar to the \*RST command. Refer to [Table 15-1](#) for more information on the preset settings.

### NOTE

All settings apply to both \*RST and SYSTem:PRESet unless stated otherwise.

### Syntax

SYST → :PRES →

### Example

SYST:PRESet      *This command presets the N432A.*

**Table 15-1** Preset settings

Command	Preset setting	Remark
CONF[1]   2   3   4?	Command function: POW:AC Expected power level: +20 dBm Resolution: 3	Returns the current configuration
CALC[1]   2   3   4:HOLD:STAT	OFF	Hold disabled
CALC[1]   2   3   4:FEED[1]   2	POW:AVER	Select the average measurement mode
CALC[1]   2   3   4:GAIN[:MAGN]	0.000 dB	Display offset value
CALC[1]   2   3   4:GAIN:STAT	OFF	Display offset disabled
CALC[1]   2   3   4:LIM:CLE:AUTO	ON	Clear limit data at INIT
CALC[1]   2   3   4:LIM:LOW[:DATA]	-90 dBm	Lower limit
CALC[1]   2   3   4:LIM:UPP[:DATA]	+90 dBm	Upper limit

**Table 15-1** Preset settings

Command	Preset setting	Remark
CALC[1] 2 3 4:LIM:STAT	OFF	Window limit checking disabled
CALC[1] 2 3 4:MATH[:EXPR]	Upper – channel A Lower – channel A	Math expression
CALC[1] 2 3 4:REL[:MAGN]:AUTO	OFF	Reference value disabled
CALC[1] 2 3 4:REL:STAT	OFF	Relative mode disabled
CAL[1]:RCF	100%	Reference calibration factor
DISP:ENAB	ON	Display enabled
DISP:SCR:FORM	WIND	Display format set to windowed
DISP[:WIND[1] 2]:ANAL:LOW	–70 dBm	Lower scale limit
DISP[:WIND[1] 2]:ANAL:UPP	20 dBm	Upper scale limit
DISP[:WIND[1] 2]:FORM	Upper – digital Lower – analog	Window format
DISP[:WIND[1] 2]:MET:LOW	–70 dBm	Analog meter lower scale limit
DISP[:WIND[1] 2]:MET:UPP	20 dBm	Analog meter upper scale limit
DISP[:WIND[1] 2]:NUM[1] 2:RES	3	Window resolution
DISP[:WIND[1] 2]:SEL[1] 2	Upper window/upper measurement	Window/measurement line selected
DISP[:WIND[1] 2]:[STAT]	ON	Both windows enabled on the display
FORM[:READ]:BORD	Normal	Byte order of binary data transfer
FORM[:READ][:DATA]	ASCII	Data format
HCOP:SDUM:DATA:FORM	PNG	Screen image file format
LXI:IDEN[:STAT]	OFF	LXI identification disabled
INIT[1]:CONT	ON	The N432A is in the wait-for-trigger state
INIT:CONT:ALL	ON	All trigger sequences are continuously initiated

**Table 15-1** Preset settings

Command	Preset setting	Remark
INIT:CONT:SEQ[1]	ON	The N432A is in the wait-for-trigger state
TRIG[1]:DEL:AUTO	ON	Settling-time delay enabled
TRIG[1]:SOUR	Immediate	Immediate triggering
MEM[:METer][1]:TAB:SEL	Not affected	Active sensor calibration table
OUTP:REC[1]:FEED	Not affected	Previous measurement
OUTP:REC[1]:LIM:AUTO	OFF	Recorder output autoscaling disabled
OUTP:REC[1]:LIM:LOW	-150 dBm	Minimum scaling value
OUTP:REC[1]:LIM:UPP	20 dBm	Maximum scaling value
OUTP:REC[1]:STAT	OFF	Recorder output disabled
OUTP:ROSC[:STAT]	OFF	50 MHz reference disabled
[SENS[1]:]AVER:COUN:VOLT	AVC16	Voltage average set to 16
[SENS[1]:]AVER:SDET	ON	Step detection enabled
[SENS[1]:]BRES	R200	Bridge resistance set to 200 $\Omega$
[SENS[1]:]CORR:CFAC   GAIN[1][:INP][:MAGN]	100.0%	Calibration factor
[SENS[1]:]CORR:CSET[1]   CSET2[:SEL]	Not affected	Selected sensor calibration table or frequency-dependent offset table
[SENS[1]:]CORR:CSET[1]   CSET2:STAT	Not affected	Selected sensor calibration table or frequency-dependent offset table state
[SENS[1]:]CORR:DCYC   GAIN3[:INP][:MAGN]	1.000%	Duty cycle factor
[SENS[1]:]CORR:DCYC   GAIN3:STAT	OFF	Duty cycle correction disabled
[SENS[1]:]CORR:GAIN2:STAT	OFF	Channel offset disabled
[SENS[1]:]CORR:GAIN2[:INP][:MAGN]	0.00 dB	Channel offset value
[SENS[1]:]FREQ[:CW   :FIX]	50 MHz	Frequency setting

**Table 15-1** Preset settings

Command	Preset setting	Remark
SERV:BACK:BRIG	80	Backlight intensity of the front panel
SERV:DISP:BSCR	OFF	Blank screen disabled
SERV:STAT	ON	Warm start enabled
SYST:COMM:GPIB[:SELF]:ADDR?	Not affected	GPIB address
UNIT[1] 2 3 4:POW	dBm	Power measurement unit
UNIT[1] 2 3 4:POW:RAT	dB	Ratio measurement unit

## SYSTem:REMOte

This command locks the N432A front panel keypad except for the **Local** key. The status reporting line of the N432A display will show “RMT”. Local front panel operation of the N432A is inhibited but can be enabled by pressing the **Local** key.

### Syntax

SYST → :REM →

### Example

SYST:REM      *This command locks the N432A front panel keypad except for the **Local** key.*



## SYSTem:RWLock

This command locks the front panel keypad including the **Local** key. The status reporting line of the N432A display will show “RMT”. In this state, the N432A cannot be returned to manual control from the front panel.

### Syntax

SYST → :RWL →

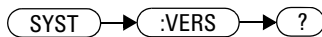
### Example

SYST:RWL      *This command locks the N432A front panel keypad including the **Local** key.*

## SYSTem:VERSion?

This query returns the version of SCPI used in the N432A. The response is in the form of XXXX.Y, where XXXX is the year and Y is the version number.

### Syntax



### Example

SYST:VERS?     *Queries which version of SCPI is used in the N432A.*



## 16 TRIGger Subsystem

TRIGger Subsystem	420
ABORt[1]	421
INITiate Commands	422
INITiate[1]:CONTInuous <boolean>	423
INITiate[1]:IMMEDIATE] 425	
INITiate:CONTInuous:ALL <boolean>	426
INITiate:CONTInuous:SEQuence[1] <boolean>	428
INITiate[:IMMEDIATE]:ALL	430
INITiate[:IMMEDIATE]:SEQuence[1]	431
TRIGger Commands	432
TRIGger[1]:DELay:AUTO <boolean>	433
TRIGger[1]:IMMEDIATE] 435	
TRIGger[1]:SOURce BUS   HOLD   IMMEDIATE	436
TRIGger[:SEQuence[1]]:DELay:AUTO <boolean>	438
TRIGger[:SEQuence[1]]:IMMEDIATE	440
TRIGger[:SEQuence[1]]:SOURce BUS   HOLD   IMMEDIATE	441

This chapter explains how the TRIGger command subsystem is used to synchronize device actions with events.



## TRIGger Subsystem

The TRIGger subsystem provides the commands for synchronizing device actions with events. It includes the ABORt, INITiate, and TRIGger commands. These are all at the root level in the command hierarchy but they are grouped here because of their close functional relationship. These commands have a numeric suffix (for example, TRIGger1) which represents channel A.

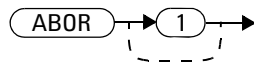
Keyword	Parameter form	Note	Page
ABORt [1]		[no query]	<a href="#">page 421</a>
INITiate[1]			
:CONTInuous	<boolean>		<a href="#">page 423</a>
[:IMMediate]		[no query]	<a href="#">page 425</a>
INITiate			
:CONTInuous			
:ALL	<boolean>		<a href="#">page 426</a>
:SEQuence[1]	<boolean>		<a href="#">page 428</a>
[:IMMediate]			
:ALL		[no query]	<a href="#">page 430</a>
:SEQuence[1]		[no query]	<a href="#">page 431</a>
TRIGger [1]			
:DELay			
:AUTO	<boolean>		<a href="#">page 433</a>
[:IMMediate]		[no query]	<a href="#">page 435</a>
:SOURce	BUS   HOLD   IMMediate		<a href="#">page 436</a>
TRIGger			
[:SEQuence[1]]			
:DELay			
:AUTO	<boolean>		<a href="#">page 438</a>
:IMMediate		[no query]	<a href="#">page 440</a>
:SOURce	BUS   HOLD   IMMediate		<a href="#">page 441</a>

## ABORt[1]

This command removes the N432A from the wait-for-trigger state and places it in the idle state. It does not affect any other settings of the trigger system. When the INITiate command is sent, the trigger system responds as it did before ABORt was executed.

If ABORt is executed while INITiate:CONTinuous is enabled, the measurement immediately goes into the wait-for-trigger state.

### Syntax



### Example

ABOR *This command places the N432A in the idle state.*

## INITiate Commands

Initiate commands allow you to place the N432A in the wait-for-trigger state.

The INITiate commands are overlapped, that is, the N432A can continue parsing and executing subsequent commands while initiated. Note that the pending operation flag is set when the N432A moves out of the idle state, and the flag is cleared when it re-enters the idle state.

The following commands are described in this section:

```
INITiate[1]:CONTInuous <boolean>
```

```
INITiate[1][:IMMEDIATE]
```

```
INITiate:CONTInuous:ALL <boolean>
```

```
INITiate:CONTInuous:SEQuence[1] <boolean>
```

```
INITiate[:IMMEDIATE]:ALL
```

```
INITiate[:IMMEDIATE]:SEQuence[1]
```

## INITiate[1]:CONTinuous <boolean>

This command sets the N432A for either a single trigger cycle or continuous trigger cycles. A trigger cycle means that the N432A exits the wait-for-trigger state and starts a measurement.

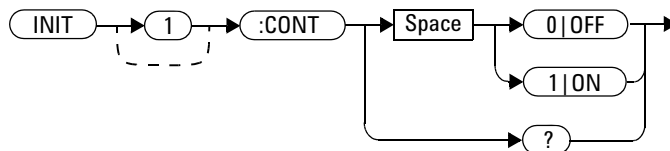
If INITiate:CONTinuous is set to:

- OFF, the trigger system remains in the idle state until it is set to ON, or INITiate:IMMEDIATE is received. Once this trigger cycle has completed, the trigger system returns to the idle state.
- ON, the trigger system is initiated and exits the idle state. On completion of each trigger cycle, the trigger system immediately commences another trigger cycle without entering the idle state.

### NOTE

- This command performs the same function as INITiate:CONTinuous:SEquence[1] <boolean>.
- When entering local mode, INITiate:CONTinuous is set to ON.

### Syntax



### Example

INIT:CONT ON

*This command places the N432A in the wait-for-trigger state.*

## Reset condition

On reset (\*RST), this command is set to OFF.

On preset (SYSTem:PRESet) and instrument power up, INITiate:CONTinuous is set to ON.

## Query

```
INITiate[1]:CONTinuous?
```

This query enters a 1 or 0 into the output buffer indicating which triggering type is used:

- 1 is returned when there is continuous triggering.
- 0 is returned when there is only a single trigger.

## Query example

```
INIT:CONT?    Queries whether the N432A is set for single or continuous triggering.
```



## INITiate[1][:IMMediate]

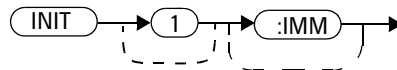
This command sets the N432A in the wait-for-trigger state. When a trigger is received, the measurement is taken and the result is placed in the N432A memory. If TRIGger[1]:SOURce is set to IMMEDIATE, the measurement begins as soon as INITiate:IMMediate is executed.

Use FETCh? to transfer a measurement from memory to the output buffer. Refer to “FETCh[1]|2|3|4 Queries” on page 123 for further details.

### NOTE

This command performs the same function as INITiate[:IMMediate]:SEQuence[1].

### Syntax



### Example

INIT:IMM *This command places the N432A in the wait-for-trigger state.*

### Error message

If the N432A is not in the idle state or INITiate:CONTinuous is set to ON, error -213, “INIT ignored” occurs.

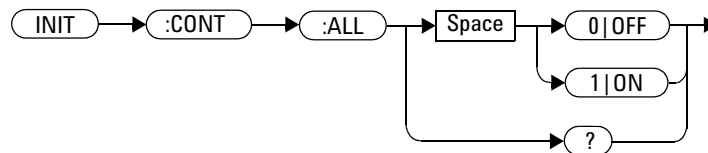
## INITiate:CONTInuous:ALL <boolean>

This command sets all trigger sequences to be continuously initiated.

If INITiate:CONTInuous:ALL is set to:

- ON, trigger sequences are set to be continuously initiated.
- OFF, trigger sequences are not set to be continuously initiated.

### Syntax



### Example

INITiate:CONT:ALL ON

*This command sets all trigger sequences to be continuously initiated.*

### Reset condition

On reset (\*RST), this command is set to OFF.

On preset (SYSTEM:PRESet) and instrument power up, INITiate:CONTInuous:ALL is set to ON.

## Query

INITiate:CONTinuous:ALL?

This query enters a 1 or 0 into the output buffer:

- 1 is returned when trigger sequences are set to be continuously initiated.
- 0 is returned when trigger sequences are not set to be continuously initiated.

## Query example

INIT:CONT:ALL?

*Queries whether the N432A is in a wait-for-trigger state.*

## INITiate:CONTInuous:SEQuence[1] <boolean>

This command sets the N432A for either a single trigger cycle or continuous trigger cycles. A trigger cycle means that the N432A exits the wait-for-trigger state and starts a measurement.

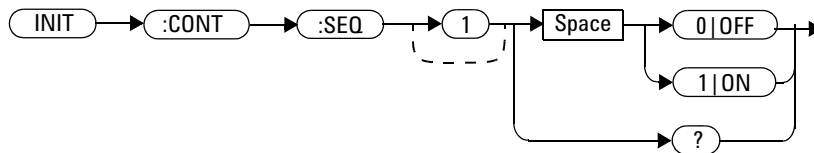
If INITiate:CONTInuous:SEQuence[1] <boolean> is set to:

- OFF, the trigger system remains in the idle state until it is set to ON, or INITiate:IMMediate is received. Once this trigger cycle has completed, the trigger system returns to the idle state.
- ON, the trigger system is initiated and exits the idle state. On completion of each trigger cycle, the trigger system immediately commences another trigger cycle without entering the idle state.

### NOTE

- This command performs the same functions as INITiate[1]:CONTInuous <boolean>.
- When entering local mode, INITiate:CONTInuous is set to ON.

### Syntax



### Example

INIT:CONT:SEQ ON *This command places the N432A in a wait-for-trigger state.*

## Reset condition

On reset (\*RST), this command is disabled.

On preset (SYSTem:PRESet) and instrument power up, this command is enabled.

## Query

```
INITiate[1]:CONTinuous:SEQuence[1]?
```

This query enters a 1 or 0 into the output buffer indicating which triggering type is used:

- 1 is returned when there is continuous triggering.
- 0 is returned when there is only a single trigger.

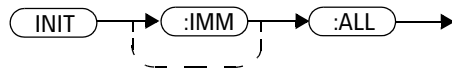
## Query example

```
INIT:CONT:SEQ? Queries whether the N432A is set for single or  
continuous triggering.
```

## INITiate[:IMMediate]:ALL

This command initiates all trigger sequences.

### Syntax



### Example

`INIT:ALL`      *This command initiates all trigger sequences.*

### Error message

If the N432A is not in the idle state or `INITiate:CONTinuous` is set to ON, error -213, “INIT ignored” occurs.

## INITiate[:IMMediate]:SEQUence[1]

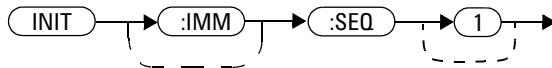
This command sets the N432A in the wait-for-trigger state. When a trigger is received, the measurement is taken and the result is placed in the N432A memory. If TRIGger[1]:SOURce is set to IMMEDIATE, the measurement begins as soon as INITiate:IMMediate is executed.

Use FETCh? to transfer a measurement from memory to the output buffer. Refer to “FETCh[1]|2|3|4 Queries” on page 123 for further information.

### NOTE

This command performs the same function as INITiate[1][:IMMediate].

### Syntax



### Example

INIT:SEQ *This command places the N432A in the wait-for-trigger state.*

### Error message

If the N432A is not in the idle state or INITiate:CONTinuous is set to ON, error -213, “INIT ignored” occurs.

## TRIGger Commands

TRIGger commands control the behavior of the trigger system.

The following commands are described in this section:

```
TRIGger[1]:DELay:AUTO <boolean>
```

```
TRIGger[1][:IMMediate]
```

```
TRIGger[1]:SOURce BUS|HOLD|IMMediate
```

```
TRIGger[:SEQuence[1]]:DELay:AUTO <boolean>
```

```
TRIGger[:SEQuence[1]]:IMMediate
```

```
TRIGger[:SEQuence[1]]:SOURce BUS|HOLD|IMMediate
```



## TRIGger[1]:DElay:AUTO <boolean>

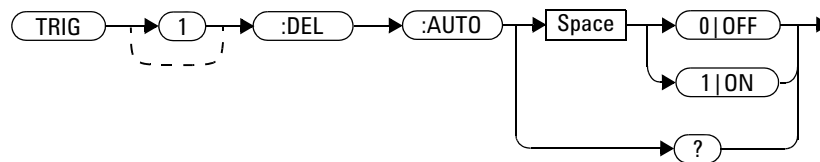
This command is used to determine whether or not there is a settling-time delay before a measurement is made.

When this command is set to:

- ON, the N432A inserts a settling-time delay before taking the requested measurement. This settling time allows the internal digital filter to be updated with new values to produce valid and accurate measurement results. The trigger with delay command allows settling time for the internal amplifiers and filters. It does not allow time for sensor delay. In cases of large power changes, the delay may not be sufficient for a complete settling. Accurate readings can be assured by taking two successive measurements for comparison.
- OFF, the N432A makes the measurement immediately after a trigger is received.

TRIGger[1]:DElay:AUTO is ignored if TRIGger[1][:IMMediate] is set to ON.

### Syntax



### Example

TRIG:DEL:AUTO ON *This command enables a delay on the N432A.*

## Reset condition

On reset, this command is set to ON.

## Query

```
TRIGger[1]:DELay:AUTO?
```

This query enters a 1 or 0 into the output buffer indicating the status of TRIGger[1]:DELay:AUTO.

- 1 is returned when it is enabled.
- 0 is returned when it is disabled.

## Query example

```
TRIG:DEL:AUTO?      Queries whether a settling-time delay is enabled or disabled.
```

## TRIGger[1][:IMMEDIATE]

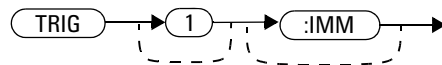
This command causes a trigger to occur immediately, provided the N432A is in the wait-for-trigger state. When this command is executed, the measurement result is stored in the N432A memory. Use `FETCH?` to place the result in the output buffer.

`TRIGger[1]:DELay:AUTO` is ignored if `TRIGger[1][:IMMEDIATE]` is set to ON.

### NOTE

This command performs the same function as `INITiate[1][:IMMEDIATE]`.

### Syntax



### Example

`TRIG` *This command causes a N432A trigger to occur immediately.*

### Error message

If the N432A is not in the wait-for-trigger state, then `TRIGger[1][:IMMEDIATE]` causes error -211, “Trigger ignored”.

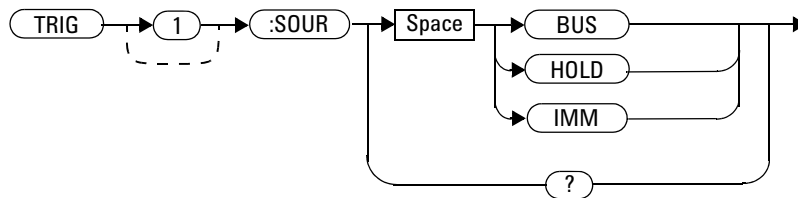
## TRIGger[1]:SOURce BUS | HOLD | IMMEDIATE

This command configures the trigger system to respond to the specified source. This command only selects the trigger source. Use the INITiate command to place the N432A in the wait-for-trigger state.

### NOTE

This command has been included for compatibility purposes. It has the same purpose as TRIGger [ :SEquence [ 1 ] ] :SOURce BUS | HOLD | IMMEDIATE which should be used in preference.

### Syntax



### Parameter

Item	Description/Default	Range of values
source	Available trigger sources: <ul style="list-style-type: none"> <li><b>BUS:</b> The trigger source is the group execute trigger (GET) bus command or the TRIGger : IMMEDIATE SCPI command.</li> <li><b>HOLD:</b> Triggering is suspended. The only way to trigger the N432A is to use TRIGger : IMMEDIATE.</li> <li><b>IMMEDIATE:</b> The trigger system is always true. If INITiate : CONTinuous is set to ON, the N432A is continually triggering free (free run mode). If an INITiate : IMMEDIATE command is sent, a measurement is triggered then the N432A returns to the idle state.</li> </ul>	BUS HOLD IMMEDIATE

**NOTE**

- The trigger source is set to IMMEDIATE on instrument power up and when entering local mode.
  - The MEASure and CONFigure commands automatically set the trigger source to IMMEDIATE.
  - The READ? query or MEASure command should not be used if the trigger source is set to BUS or HOLD.
- 

**Example**

TRIG:SOUR IMM *This command configures the N432A for immediate triggering.*

**Reset condition**

On reset, the trigger source is set to IMMEDIATE.

**Query**

TRIGger[1]:SOURce?

This query returns the current trigger source of either IMM, BUS, or HOLD.

**Query example**

TRIG:SOUR? *Queries the N432A trigger source.*

## TRIGger[:SEQuence[1]]:DELay:AUTO <boolean>

This command is used to determine whether or not there is a settling-time delay before a measurement is made.

When this command is set to:

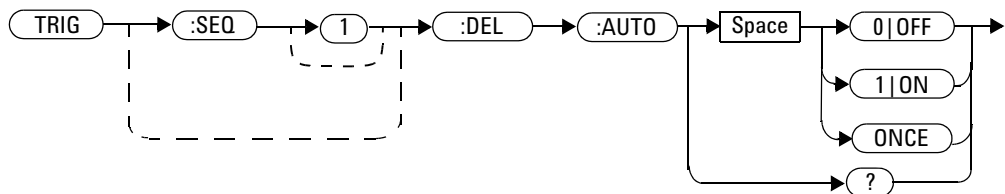
- ON, the N432A inserts a settling-time delay before taking the requested measurement and for subsequent measurements. This settling time allows the internal digital filter to be updated with new values to produce valid and accurate measurement results. The trigger with delay command allows settling time for the internal amplifiers and filters. It does not allow time for sensor delay.

In cases of large power changes, the delay may not be sufficient for a complete settling. Accurate readings can be assured by taking two successive measurements for comparison.

- OFF, no settling-time delay is inserted and the N432A makes the measurement immediately after a trigger is received.
- ONCE, a settling-time delay is inserted before taking the requested measurement, for one measurement only.

TRIGger:DELay:AUTO is ignored if TRIGger:IMMediate is set to ON.

### Syntax



## Example

TRIG:DEL:AUTO ON *This command enables a delay on the N432A.*

## Reset condition

On reset, this command is set to ON.

## Query

TRIGger[:SEquence[1]]:DELay:AUTO?

This query enters a 1 or 0 into the output buffer indicating the status of TRIGger[:SEquence[1]]:DELay:AUTO.

- 1 is returned when it is enabled.
- 0 is returned when it is disabled.

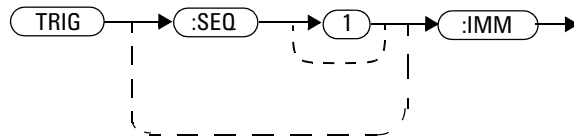
## Query example

TRIG:DEL:AUTO? *Queries the state of the N432A settling-time delay.*

## TRIGger[:SEQuence[1]]:IMMEDIATE

This command provides a one time override of the normal process of the downward path through the wait-for-trigger state. It causes the immediate exit of the event detection layer if the trigger system is in this layer when the command is received. In other words, the N432A stops waiting for a trigger and takes a measurement ignoring any delay set by TRIG:DELAy.

### Syntax



### Example

TRIG:IMM

*This command initiates a measurement on the N432A.*



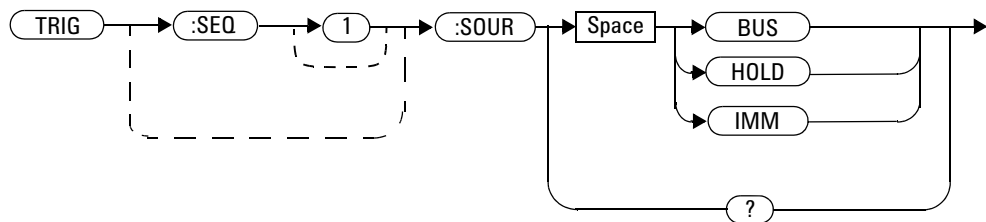
## TRIGger[:SEQuence[1]]:SOURce BUS | HOLD | IMMEDIATE

This command configures the trigger system to respond to the specified source. This command only selects the trigger source. Use the INITiate command to place the N432A in the wait-for-trigger state.

### NOTE

This command has the same purpose as TRIGger [1] :SOURce BUS | HOLD | IMMEDIATE.

### Syntax



### Parameter

Item	Description/Default	Range of values
source	Available trigger sources: <ul style="list-style-type: none"> <li><b>BUS:</b> The trigger source is the group execute trigger (GET) bus command or the TRIGger:IMMEDIATE SCPI command.</li> <li><b>HOLD:</b> Triggering is suspended. The only way to trigger the N432A is to use TRIGger:IMMEDIATE.</li> <li><b>IMMEDIATE:</b> The trigger system is always true. If INITiate:CONTinuous is set to ON, the N432A is continually triggering free (free run mode). If an INITiate:IMMEDIATE command is sent, a measurement is triggered then the N432A returns to the idle state.</li> </ul>	BUS HOLD IMMEDIATE

### NOTE

- The trigger source is set to IMMEDIATE on instrument power up and when entering local mode.
  - The MEASURE and CONFIGure commands automatically set the trigger source to IMMEDIATE.
  - The READ? query or MEASURE command should not be used if the trigger source is set to BUS or HOLD.
- 

### Example

TRIG:SOUR IMM      *This command configures the N432A for immediate triggering.*

### Reset condition

On reset, the trigger source is set to IMMEDIATE.

### Query

TRIGger[:SEquence[1]]:SOURce?

This query returns the current trigger source of either IMM, BUS, or HOLD.

### Query example

TRIG:SOUR?      *Queries the current trigger source for the N432A.*



## 17 UNIT Subsystem

UNIT Subsystem [444](#)

UNIT[1]|2|3|4:POWer <amplitude\_unit> [445](#)

UNIT[1]|2|3|4:POWer:RATio <ratio\_unit> [447](#)

This chapter explains how the `UNIT` command subsystem is used to set the N432A measurement units to Watts and % (linear), or dBm and dB (logarithmic).

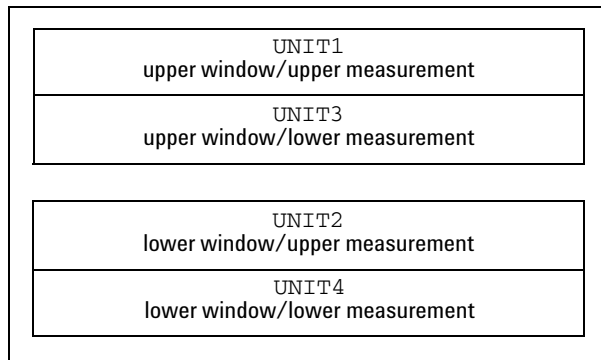


## UNIT Subsystem

The UNIT command subsystem:

- sets the power measurement unit to dBm or Watts.
- sets the measurement ratio unit to dB or %.

The UNIT commands have a numeric suffix which represents the selected window/measurement:



**Figure 17-1** Measurement display UNIT block window

The following commands are described in this section:

Keyword	Parameter form	Note	Page
UNIT[1]   2   3   4			
:POWer	<amplitude_unit>		<a href="#">page 445</a>
:RATio	<ratio_unit>		<a href="#">page 447</a>

The UNIT:POWer and UNIT:POWer:RATio commands are coupled as follows:

- If UNIT:POWer is set to dBm, then UNIT:POWer:RATio is dB.
- If UNIT:POWer is set to W, then UNIT:POWer:RATio is %.

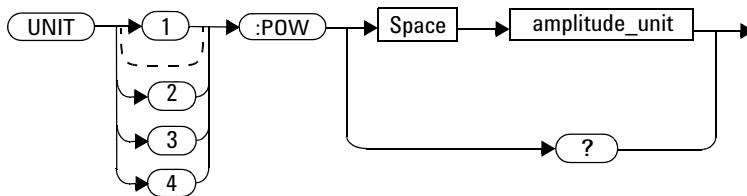
## UNIT[1]|2|3|4:POWer <amplitude\_unit>

This command sets the power measurement unit for a specified window/measurement.

For ratio and relative power measurements:

- If UNIT:POWer is set to W, the measurement unit is percentage.
- If UNIT:POWer is set to DBM, the measurement unit is dB relative.

### Syntax



### Parameter

Item	Description/Default	Range of values
amplitude_unit	Power measurement unit. The default unit is DBM.	DBM W

### Example

UNIT1:POW DBM *This command sets the measurement unit to dBm for the upper window/upper measurement.*

### Reset condition

On reset, the unit is DBM for all windows/measurements.

## Query

```
UNIT[1] | 2 | 3 | 4 : POWer?
```

This query returns the current setting of the power measurement unit.

## Query example

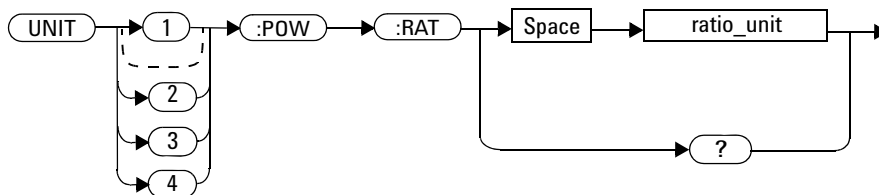
```
UNIT2 : POW?
```

*Queries which measurement unit is being used for the lower window/upper measurement.*

## UNIT[1|2|3|4]:POWer:RATio <ratio\_unit>

This command sets the ratio measurement unit for a particular window/measurement.

### Syntax



### Parameter

Item	Description/Default	Range of values
ratio_unit	Ratio measurement unit. The default unit is DB.	DB PCT

### Example

UNIT1:POW:RAT DB *This command sets the ratio measurement unit to dB for the upper window/upper measurement.*

### Reset condition

On reset, the ratio unit is DB.

### Query

UNIT[1|2|3|4]:POWer:RATio?

This query returns the current setting of the ratio measurement unit.

### Query example

UNIT2:POW:RAT? *Queries which ratio measurement unit is being used for the lower window/upper measurement.*







## 18 Programming Examples

- Example 1: Identifying the N432A In Use [450](#)
- Example 2: FETCh, MEASure, and READ Queries [451](#)
- Example 3: Basic Power Measurement [453](#)
- Example 4: Status of Various Settings [454](#)
- Example 5: Display Offset, Max/Min Hold, and Limit Settings [455](#)
- Example 6: Frequency-Dependent Offsets [456](#)

This chapter displays the programming sequences or examples to remotely control the N432A using SCPI commands.



## Example 1: Identifying the N432A In Use

The following example shows how to verify whether you are communicating with the right N432A. Refer to “[Error list](#)” on page 404 for more information on the error messages.

```
-> *IDN?           //Queries the instrument identification and checks
                   //whether you are communicating with the right
                   //N432A.
-> SYST:ERR?       //Checks the N432A system error queue.
```

**NOTE**

“->” indicates the commands that you send to the N432A.

“<-” indicates the response from the N432A.

---

## Example 2: FETCh, MEASure, and READ Queries

There are three different ways to query the power measurement – using FETCh?, MEASure?, and READ?.

In the Free Run or Continuous mode, you can use either FETCh? or MEASure? to query the power measurement as shown in the following examples:

```
-> INIT:CONT ON      //Sets the N432A to the Free Run mode.
-> FETC?             //Queries the measurement result from the buffer.
```

or

```
-> INIT:CONT ON      //Sets the N432A to the Free Run mode.
-> MEAS?             //Reads the measurement result. MEAS? is
                    //equivalent to CONF followed by a READ?.
```

In the Single Trigger mode, you can use FETCh?, MEASure?, or READ? to query the power measurement as shown in the following examples:

```
-> INIT:CONT OFF     //Sets the N432A to the Single Trigger mode.
-> CONF              //Configures the measurement.
-> INIT              //Initializes the measurement.
-> FETC?             //Queries the measurement result. The above
                    //sequence must be followed.
```

or

```
-> INIT:CONT OFF     //Sets the N432A to the Single Trigger mode.
-> MEAS?             //Reads the measurement result. MEAS? is
                    //equivalent to CONF followed by a READ?.
```

or

```
-> INIT:CONT OFF //Sets the N432A to the Single Trigger mode.
-> CONF //Configures the measurement.
-> READ? //Reads the measurement result. READ? is
equivalent to INIT followed by a FETC?
(Assuming that TRIG:SOUR is set to
IMMediate).
```

You can also use FETCh? to query voltage measurements in the Free Run/Continuous mode or the Single Trigger mode as shown in the following examples:

```
-> INIT:CONT ON //Sets the N432A to the Free Run mode.
-> FETC:V1? //Queries the  $V_1$  measurement result from the
buffer.
```

```
-> INIT:CONT OFF //Sets the N432A to the Single Trigger mode.
-> CONF //Configures the measurement.
-> INIT //Initializes the measurement.
-> FETC:VRF1? //Queries the  $V_{RF1}$  measurement result. The above
sequence must be followed.
```

### NOTE

"->" indicates the commands that you send to the N432A.

"<-" indicates the response from the N432A.

---

## Example 3: Basic Power Measurement

The following command sequences provide the examples to perform a simple power measurement with Free Run and Single Trigger modes.

### Free Run mode

```
-> SYST:PRES //Presets the N432A.
-> INIT:CONT ON //Sets the N432A to the Free Run mode.
-> FREQ 1000MHZ //Sets the frequency to 1000 MHz.
-> CORR:CFAC 80 //Sets the calibration factor to 80%.
-> FETC? //Queries the measurement result. The above
sequence must be followed.
```

### Single Trigger mode

```
-> SYST:PRES //Presets the N432A.
-> INIT:CONT OFF //Sets the N432A to the Single Trigger mode.
-> FREQ 1000MHZ //Sets the frequency to 1000 MHz.
-> CORR:CFAC 80 //Sets the calibration factor to 80%.
-> INIT //Initializes the measurement.
-> FETC? //Queries the measurement result. The above
sequence must be followed.
```

#### NOTE

In the Single Trigger mode:

- INIT must be executed before FETC?.
- MEAS? can be used without executing INIT.
- READ? can be used without executing INIT.

#### NOTE

"->" indicates the commands that you send to the N432A.

"<-" indicates the response from the N432A.

## Example 4: Status of Various Settings

The following example shows a program that polls registers to display the status of various settings.

```

-> SYST:PRES //Presets the N432A.
-> STAT:OPER:CAL:COND? //Queries the zeroing status of the
                        N432A.
<- +0 //Returns a 0.
-> CAL:ZERO:AUTO ONCE //Performs zeroing.
-> STAT:OPER:CAL:COND? //Queries the zeroing status of the
                        N432A.
<- +2 //The N432A is performing zeroing.
-> STAT:OPER:CAL:COND? //Queries the zeroing status of the
                        N432A.
<- +2 //The N432A is performing zeroing.
Upon completion of zeroing
-> STAT:OPER:CAL:COND? //Queries the zeroing status of the
                        N432A.
-> +0 //Returns a 0 at the end of zeroing.

```

### NOTE

"->" indicates the commands that you send to the N432A.

"<-" indicates the response from the N432A.

## Example 5: Display Offset, Max/Min Hold, and Limit Settings

Below shows an example of a command sequence using the display offset, max/min hold, and limit features.

Signal level: -3 dBm

```
-> SYST:PRES //Presets the N432A.
-> CALC:GAIN 3 //Sets the display offset to 3 dB.
-> CALC:GAIN:STAT ON //Turns on display offset.
The measurement now shows 0 dBm.
-> CALC:LIM:UPP -3 //Sets the window upper limit to
-3 dBm.
-> CALC:LIM:STAT ON Turns on the limit-checking function.
-> STAT:OPER:ULF? //Checks the upper limit fail status.
<- +8 //The upper window upper limit fail
status has been set.
```

“Over Limit” is displayed on the measurement window as the measurement exceeds the upper-limit level.

```
-> CALC:HOLD:STAT MAX //Sets the hold value to maximum.
```

Decrease the signal level to -23 dBm.

The measurement still shows 0 dBm as the hold feature is set to MAX and the MAX value is shown, which is 0 dBm.

```
-> CALC:HOLD:STAT OFF //Turns off the hold feature.
```

Measurement now shows -20 dBm.

```
-> CALC:GAIN:STAT OFF //Turns off the display offset.
```

Measurement now shows -23 dBm.

### NOTE

“->” indicates the commands that you send to the N432A.

“<-” indicates the response from the N432A.

## Example 6: Frequency-Dependent Offsets

The frequency-dependent offset (FDO) feature provides you with a convenient way to store the offset values in a handy table. Multiple FDO tables can be created to compensate for different external test setups with different frequency responses. By enabling the FDO table during measurement, the pre-entered offset values will be retrieved to compensate for your external test setup over a range of frequencies.

The following command sequence provides a simple example to create an FDO table, assign an offset for a frequency value, store the table under your preferred name, and turn on the table for measurement use.

```
-> MEM:TABL:SEL "CUSTOM_A" //Selects an FDO table named
                             "CUSTOM_A".
-> MEM:TABL:FREQ 50MHZ //Enters a frequency of 50 MHz into
                        the selected table.
-> MEM:TABL:GAIN 50 //Enters an offset factor of 50%
                    into the selected table.
-> CORR:CSET2 "CUSTOM_A" //Enters the name of the selected
                           table.
-> CORR:CSET2:STAT 1 //Enables the selected table.
-> FREQ 1000MHZ //Sets the frequency to 1000 MHz.
-> FETC? //Queries the measurement result.
          The above sequence must be
          followed.
```

### NOTE

"->" indicates the commands that you send to the N432A.

"<-" indicates the response from the N432A.





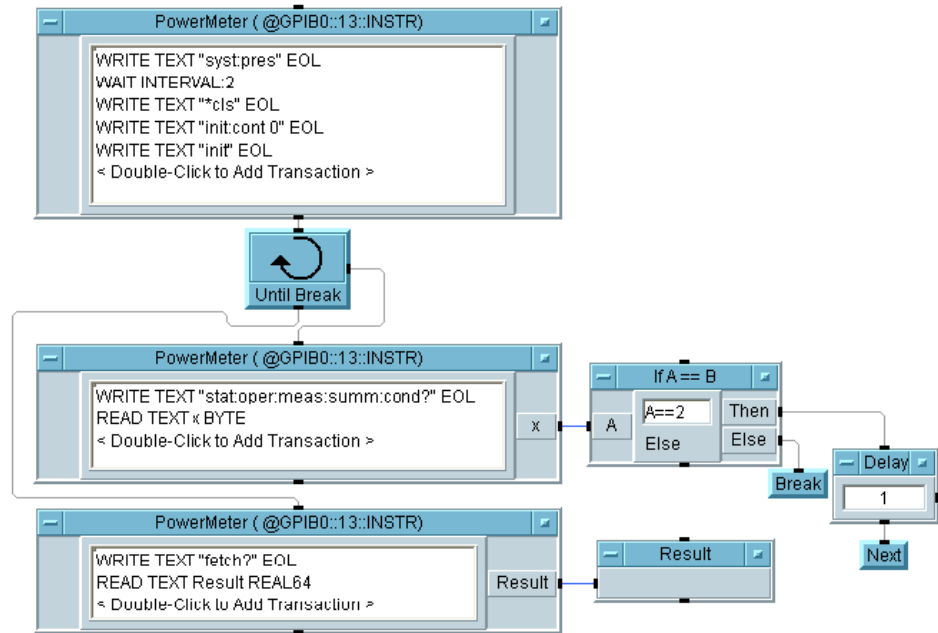
## **A** **Measurement Polling Example Using** **VEE**

This chapter contains a VEE program example in measurement polling.



## A Measurement Polling Example Using VEE

The following figure provides an example on how to perform measurement polling with a VEE program. The information relates to the condition polling method as described in “Condition polling method” on page 57.



**www.agilent.com**

### **Contact us**

To obtain service, warranty, or technical assistance, contact us at the following phone or fax numbers:

United States:

(tel) (800) 829 4444 (fax) 800 829 4433

Canada:

(tel) (877) 894 4414 (fax) 800 746 4866

China:

(tel) 800 810 0189 (fax) 800 820 2816

Europe:

(tel) 31 20 547 2111

Japan:

(tel) 0120 (421) 345 (fax) 0120 421 678

Korea:

(tel) 080 769 0800 (fax) (080) 769 0900

Latin America:

(tel) 305 269 7500

Taiwan:

(tel) 0800 047 866 (fax) 0800 286 331

Other Asia Pacific Countries:

(tel) (65) 6375 8100 (fax) (65) 6755 0042

Or visit Agilent World Wide Web at:

[www.agilent.com/find/assist](http://www.agilent.com/find/assist)

Product specifications and descriptions in this document are subject to change without notice. Always refer to the Agilent web site for the latest revision.

© Agilent Technologies, Inc. 2010

Printed in Malaysia  
First Edition, May 13, 2010

N432A-90004



**Agilent Technologies**