Agilent Technologies 8960 Series 10 E5515T Wireless Communications Test Set
Agilent Technologies E1962A CDMA 2000 Mobile Test Mode Application

GPIB Command Syntax

Test Application Revision: A.01

Printed in U.S.A. October 2000

Agilent Part Number: E1962-90013

http://www.agilent.com/find/8960support/

Agilent Technologies

Notice

Information contained in this document is subject to change without notice.

Agilent Technologies, Inc.
Learning Products Department
24001 E. Mission
Liberty Lake, WA 99019-9599
U.S.A.

Contents

Contents

Contents

Contents

Diagram Conventions

Description

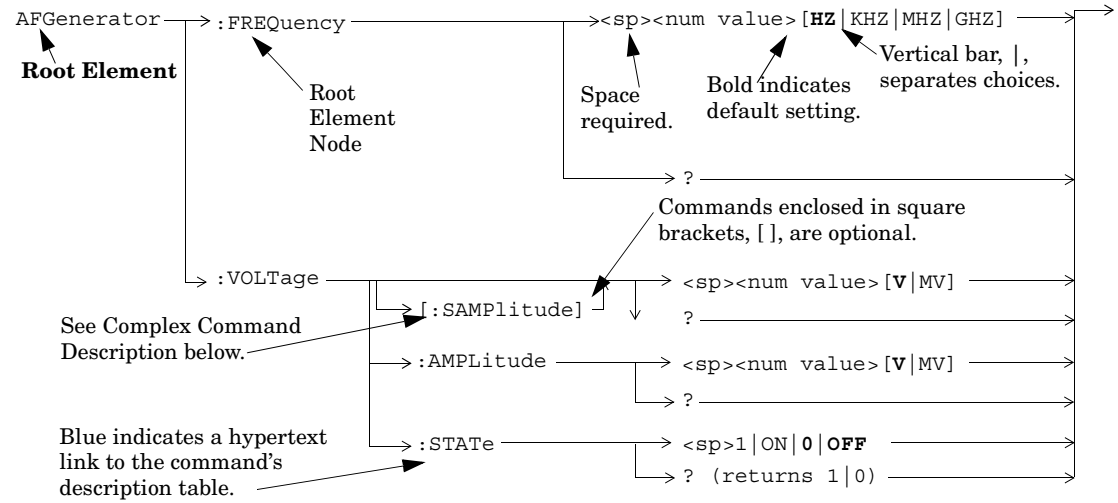


Diagram Description

Statement elements are connected by lines. Each line can be followed in only one direction, as indicated by the arrow at the end of the line. Any combination of statement elements that can be generated by starting at the **Root Element** and following the line the **directionof the arrow** is syntactically correct. The drawings show the proper use of spaces. Where spaces are required they are indicated by **<sp>**, otherwise no spaces are allowed between statement elements.

Complex Command Description A complex command sets the state of the parameter to ON, and is used to set a value for that parameter. These parameters; amplitude, frequency, gain, number, time, and value can be used as a complex command. Refer to the specific command for the parameter that applies.

Developing Code

It is recommended that you set the Test Set's operating environment to debug. To set the Test Set debug mode to "ON" use the following syntax:

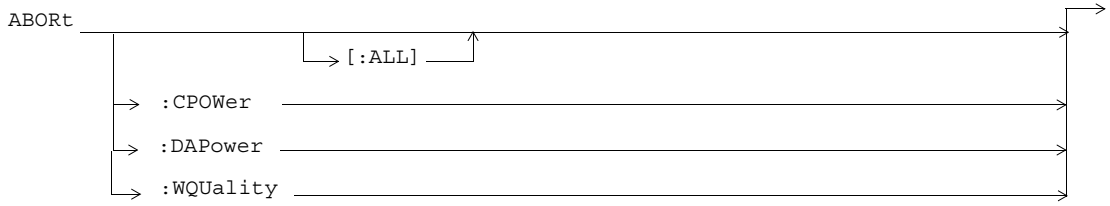
```
SYSTem:COMMunicate:GPIB:DEBug ON
```

Units-of-Measure

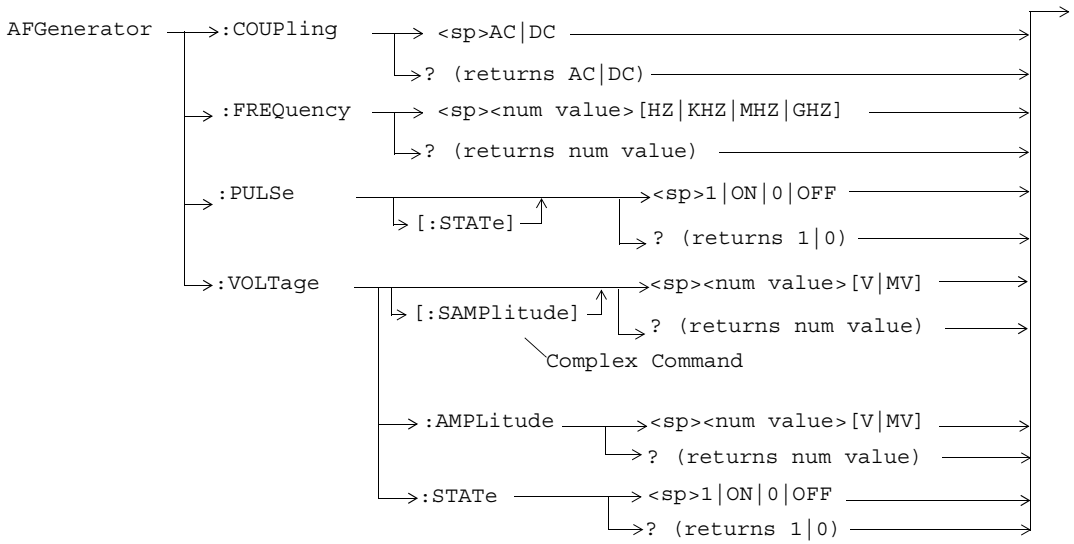
If you do not specify units-of-measure in your code the following table indicates the default units-of-measure that will be assumed.

Amplitude (linear)	V
Frequency	Hz
Power (logarithmic)	dBm
Time	s

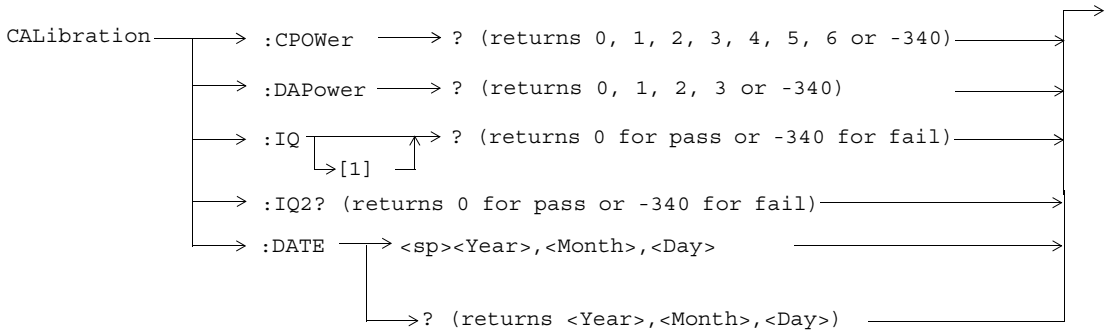
ABORt



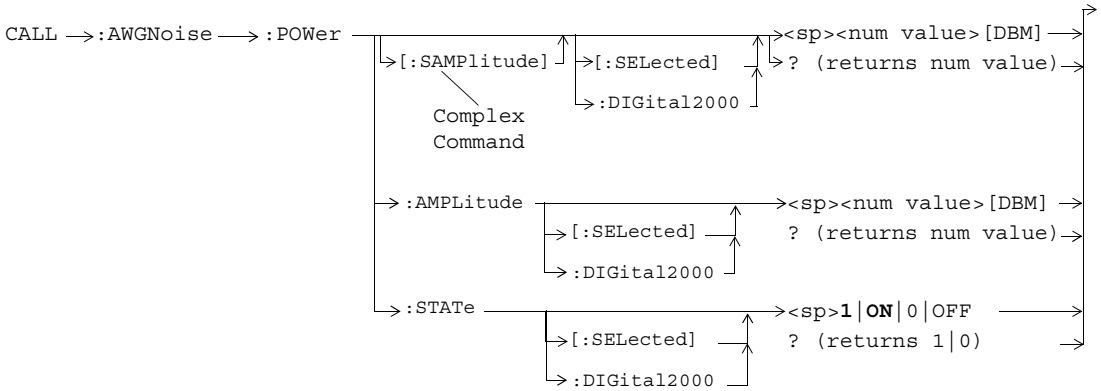
AFGenerator



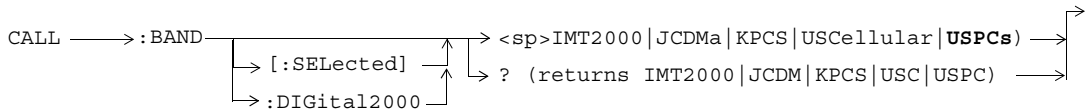
CALibration



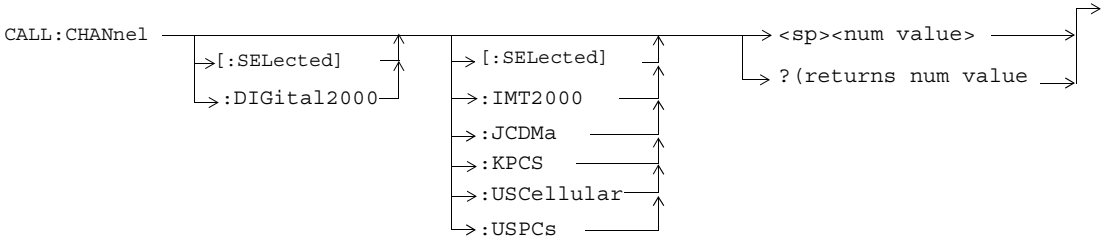
CALL:AWGNoise:POWer



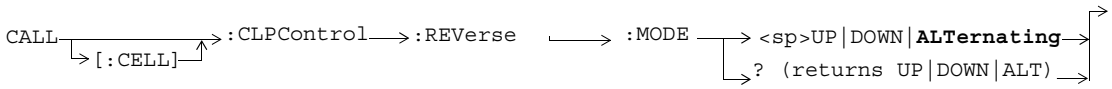
CALL:BAND


CALL
:BAND
[:SELected]
:DIGital2000
<sp>IMT2000|JCDMa|KPCS|USCellular|USPCs)
? (returns IMT2000|JCDM|KPCS|USC|USPC)

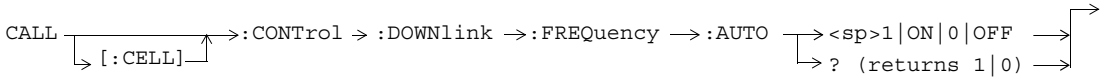
CALL:CHANnel



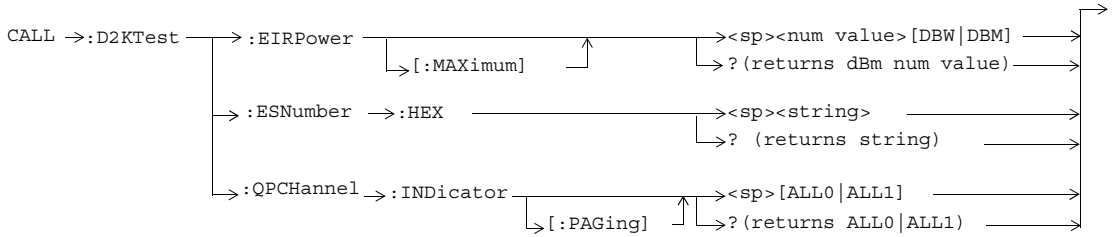
CALL[:CELL]:CLPControl



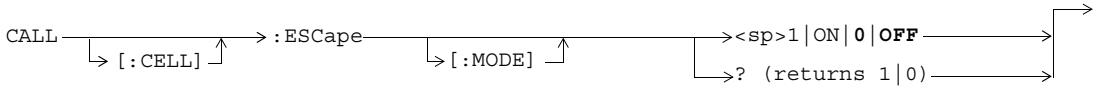
CALL[:CELL]:CONTrol:DOWNlink:FREQuency:AUTO



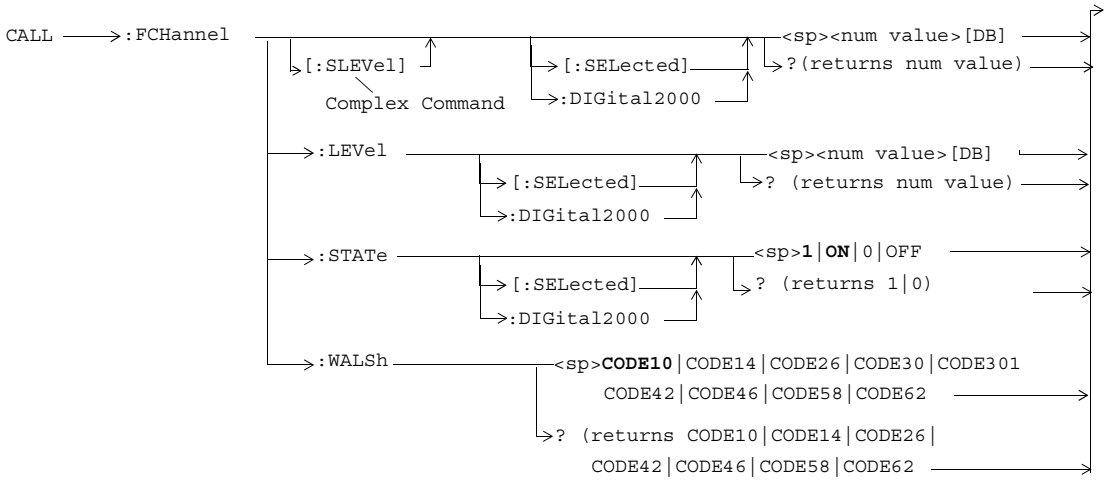
CALL:D2KTest



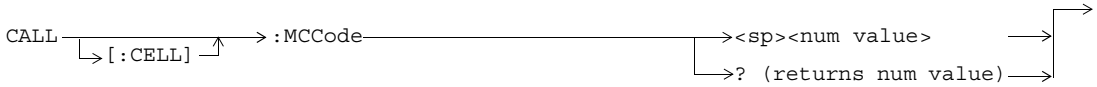
CALL[:CELL]:ESCape



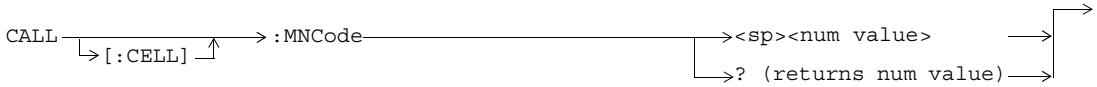
CALL:FCHannel



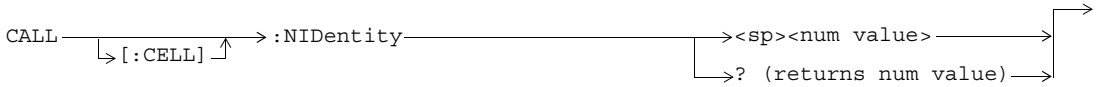
CALL[:CELL]:MCCode



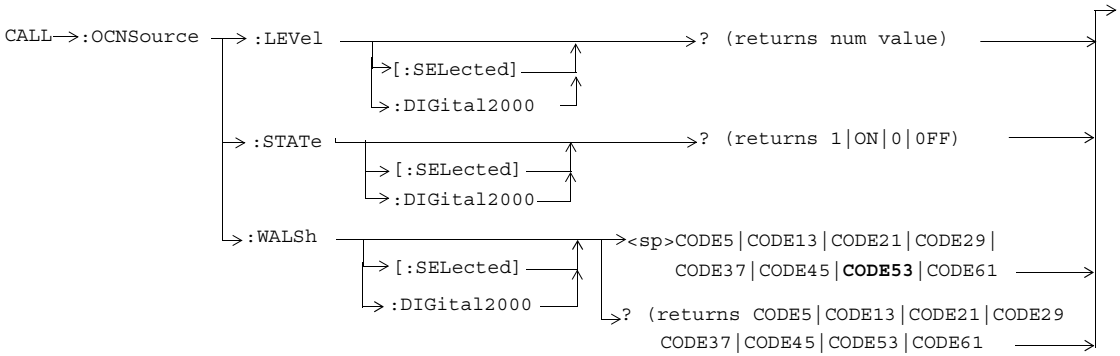
CALL[:CELL]:MNCode



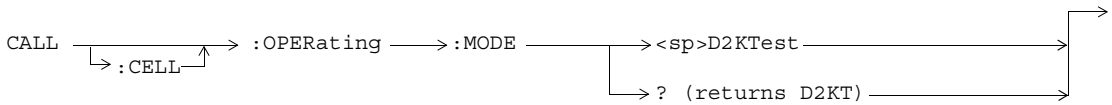
CALL[:CELL]:NIDentity



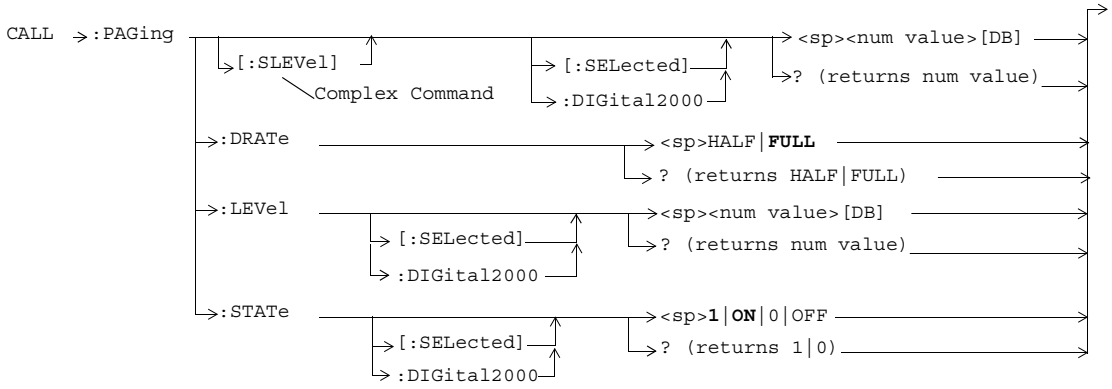
CALL:OCNSource



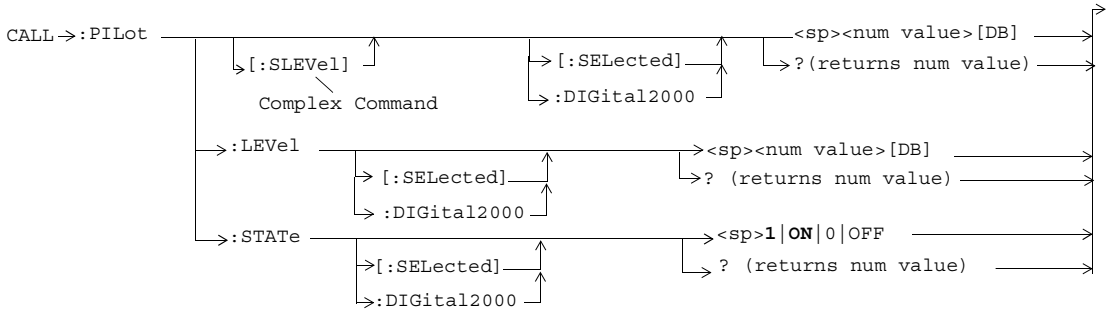
CALL[:CELL]:OPERating



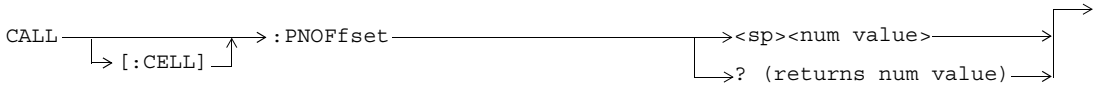
CALL:PAGing


CALL
:PAGing
[:SLEVel]
Complex Command
[:SELected]
:DIGital2000
<sp><num value>[DB]
? (returns num value)
:DRATe
<sp>HALF|FULL
? (returns HALF|FULL)
:LEVel
[:SELected]
:DIGital2000
<sp><num value>[DB]
? (returns num value)
:STATe
[:SELected]
:DIGital2000
<sp>1|ON|0|OFF
? (returns 1|0)

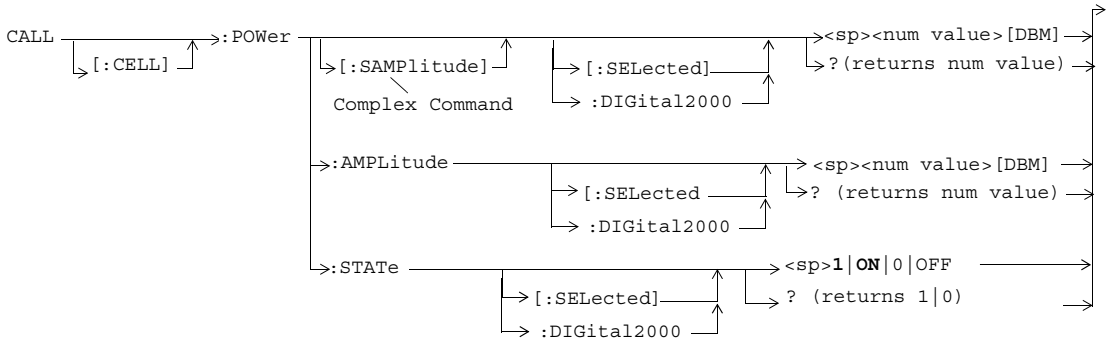
CALL:PILot


CALL
:PILot
[:SLEVel]
Complex Command
[:SELected]
:DIGital2000
<sp><num value>[DB]
?(returns num value)
:LEVel
[:SELected]
:DIGital2000
<sp><num value>[DB]
? (returns num value)
:STATe
[:SELected]
:DIGital2000
<sp>1|ON|0|OFF
? (returns num value)

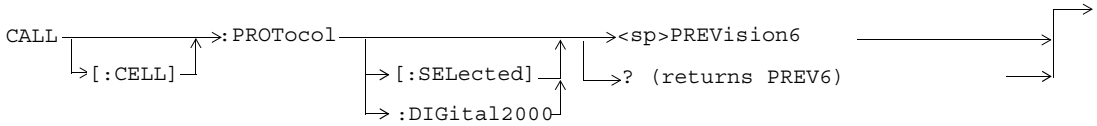
CALL[:CELL]:PNOFfset



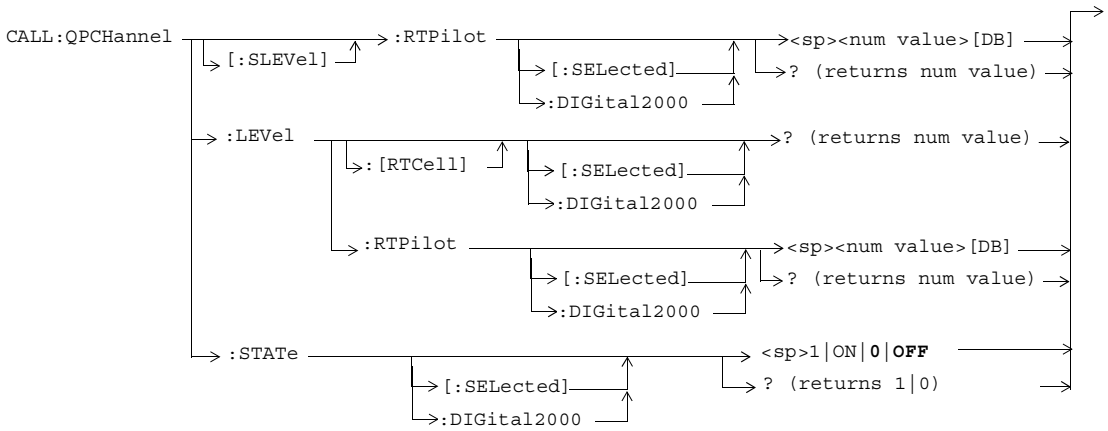
CALL[:CELL]:POWer



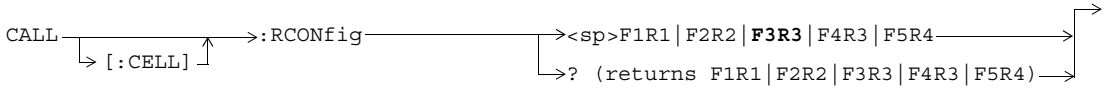
CALL[:CELL]:PROTocol



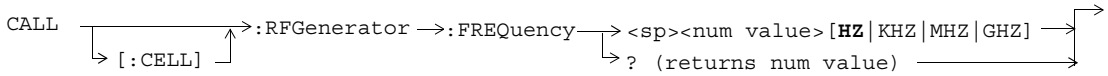
CALL:QPCHannel



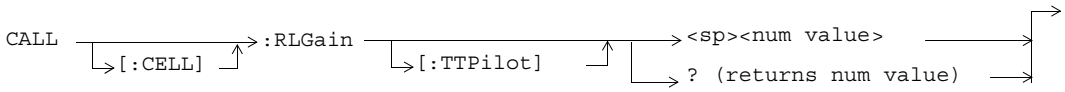
CALL[:CELL]:RCONfig



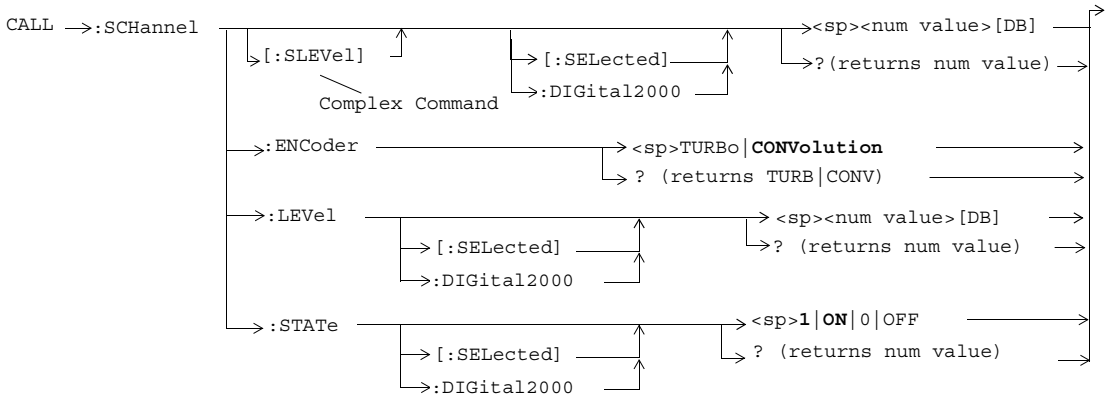
CALL[:CELL]:RFGenerator

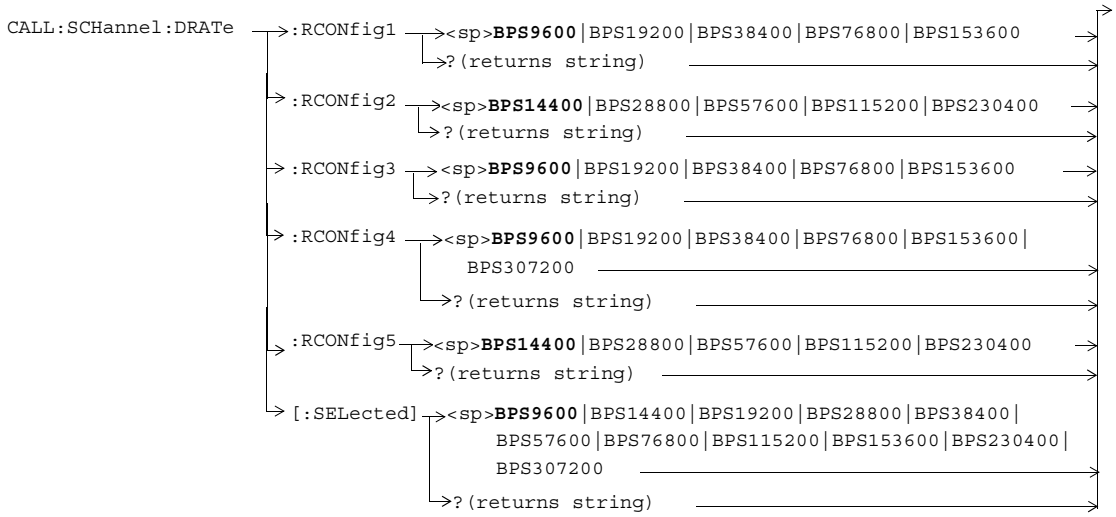


CALL[:CELL]:RLGain



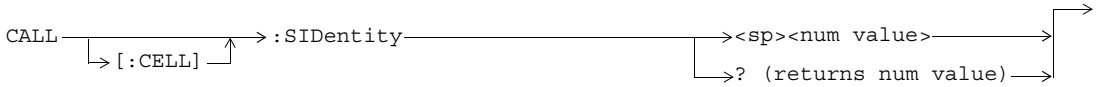
CALL:SCHannel



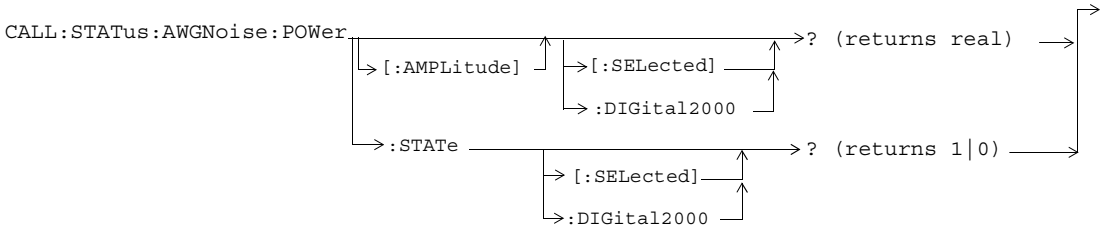


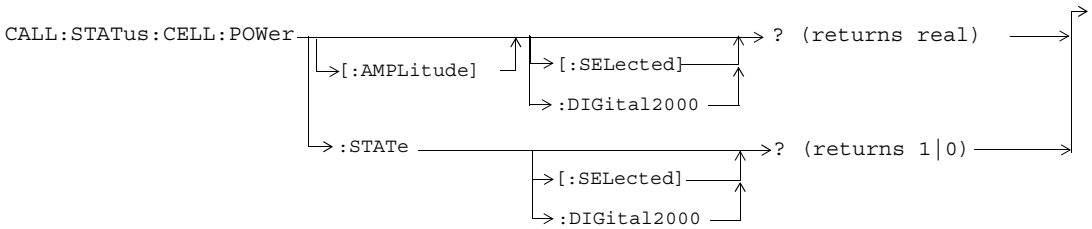
Note: Data rates used with the SELected keyword must be compatible with the currently selected radio configuration. If a data rate that does not belong to the selected radio configuration's rate set is used, an error message will be generated.

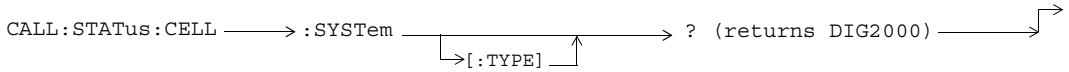
CALL[CELL]:SIDentity

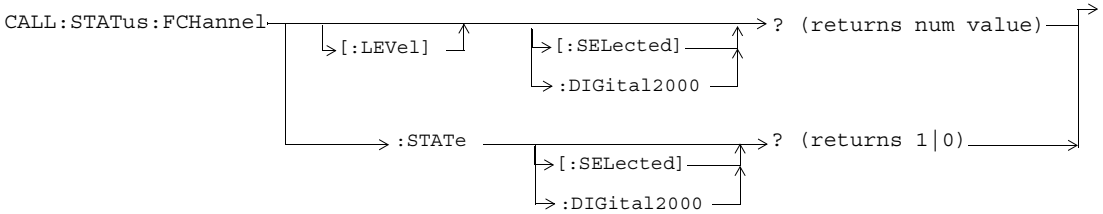


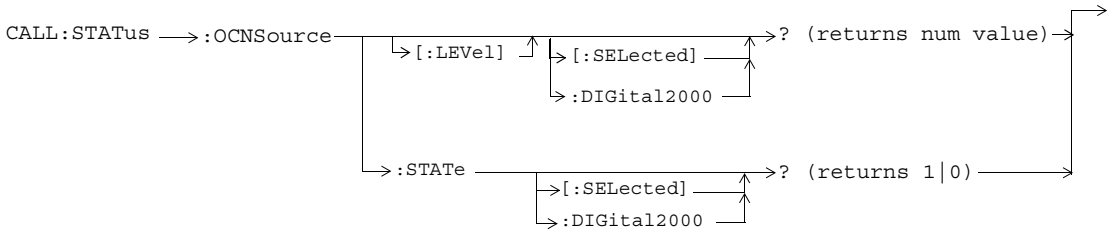
CALL:STATus

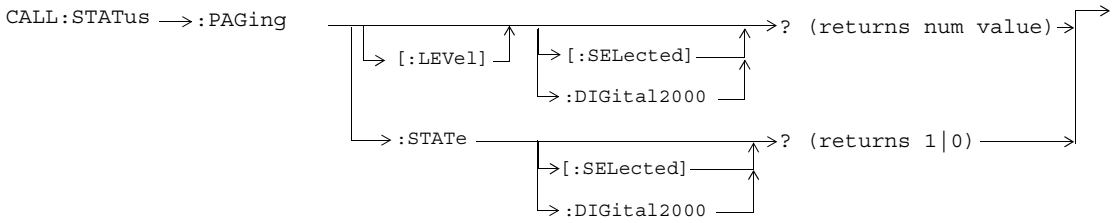

CALL:STATus:AWGNoise:POWer
[:AMPLitude]
[:SELected]
:DIGital2000
? (returns real)
:STATe
[:SELected]
:DIGital2000
? (returns 1|0)

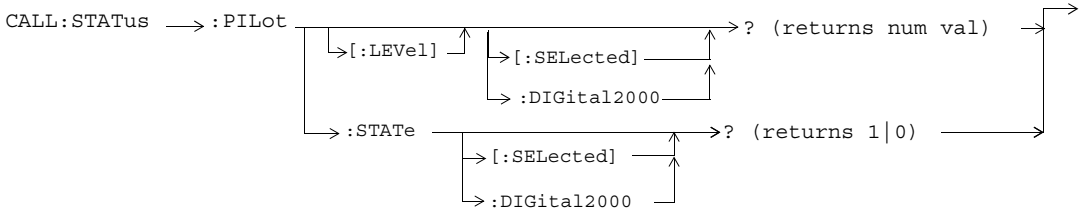

CALL:STATus:CELL:POWer
[:AMPLitude]
[:SELected]
:DIGital2000
? (returns real)
:STATe
[:SELected]
:DIGital2000
? (returns 1|0)

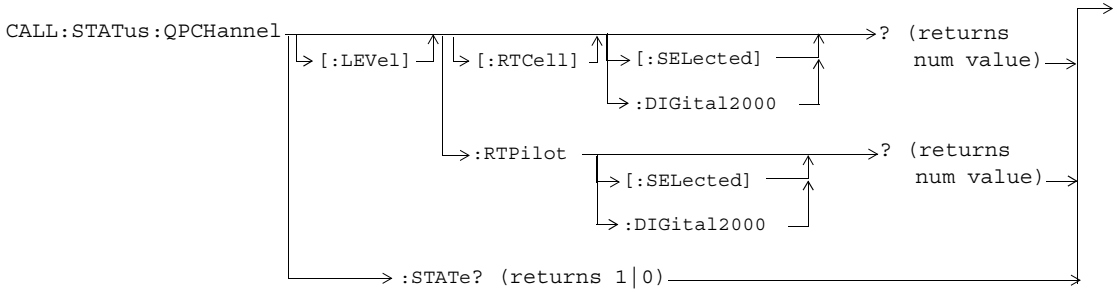

CALL:STATus:CELL
:SYSTem
[:TYPE]
? (returns DIG2000)

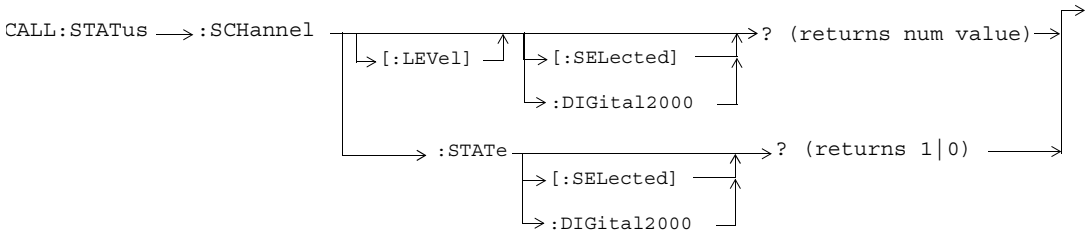

CALL:STATus:FCHannel
[:LEVel]
[:SELected]
:DIGital2000
? (returns num value)
:STATe
[:SELected]
:DIGital2000
? (returns 1|0)

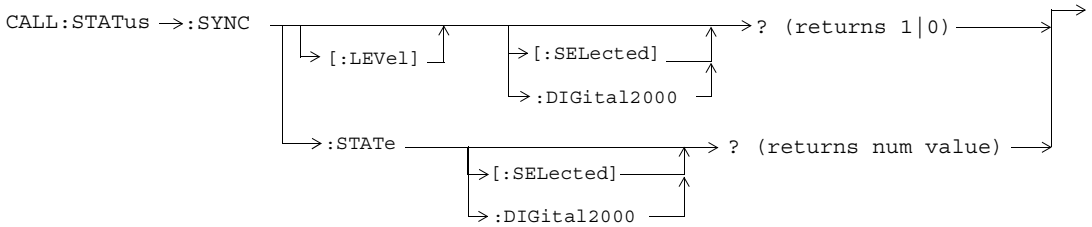

CALL:STATus
:OCNSource
[:LEVel]
[:SELected]
:DIGital2000
? (returns num value)
:STATe
[:SELected]
:DIGital2000
? (returns 1|0)

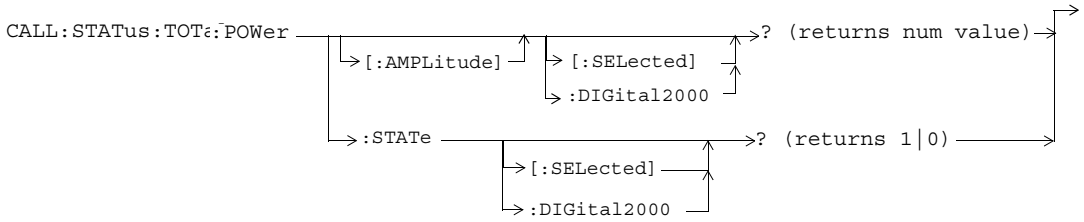

CALL:STATus
:PAGing
[:LEVel]
[:SELected]
:DIGital2000
? (returns num value)
:STATe
[:SELected]
:DIGital2000
? (returns 1|0)


CALL:STATus
:PILot
[:LEVel]
[:SELected]
:DIGital2000
? (returns num val)
:STATe
[:SELected]
:DIGital2000
? (returns 1|0)

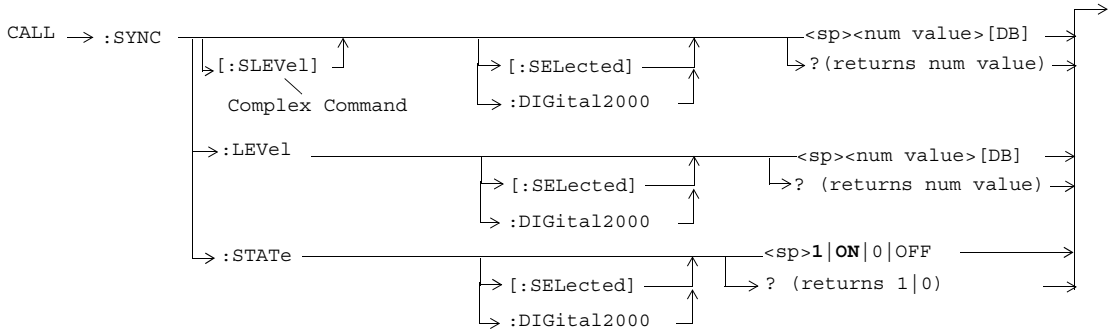

CALL:STATus:QPCHannel
[:LEVel]
[:RTCell]
[:SELected]
:DIGital2000
? (returns
num value)
:RTPilot
[:SELected]
:DIGital2000
? (returns
num value)
:STATe? (returns 1|0)


CALL:STATus
:SCHannel
[:LEVel]
[:SELected]
:DIGital2000
? (returns num value)
:STATe
[:SELected]
:DIGital2000
? (returns 1|0)

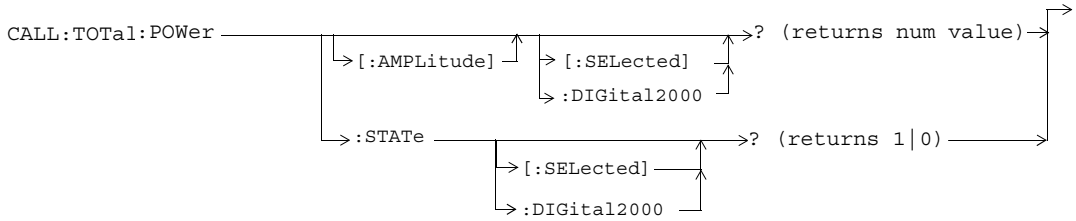

CALL:STATus
:SYNC
[:LEVel]
[:SELected]
:DIGital2000
? (returns 1|0)
:STATe
[:SELected]
:DIGital2000
? (returns num value)


CALL:STATus:TOTal:POWer
[:AMPLitude]
[:SELected]
:DIGital2000
? (returns num value)
:STATe
[:SELected]
:DIGital2000
? (returns 1|0)

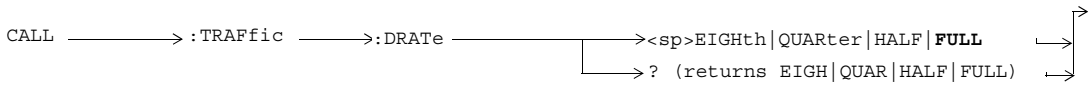
CALL:SYNC


CALL
:SYNC
[:SLEVel]
Complex Command
[:SELected]
:DIGital2000
<sp><num value>[DB]
?(returns num value)
:LEVel
[:SELected]
:DIGital2000
<sp><num value>[DB]
? (returns num value)
:STATe
[:SELected]
:DIGital2000
<sp>1|ON|0|OFF
? (returns 1|0)

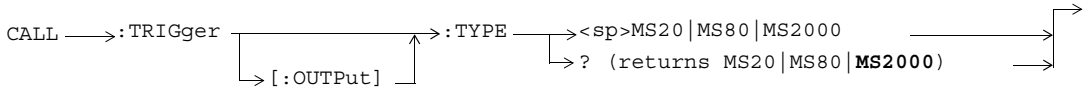
CALL:TOTal:POWer



CALL:TRAFfic



CALL:TRIGger[:OUTPut]:TYPE



IEEE 488.2 Common Commands

Description

***CLS**

The *CLS, clear status command, is defined in “IEEE Std 488.2-1992”, 10.3. This command will also clear and close the error message screen on the test set’s display.

***ESE**

The *ESE, standard event status enable command, is defined in “IEEE Std 488.2-1992”, 10.10.

***ESE?**

The *ESE?, standard event status enable query, is defined in “IEEE Std 488.2-1992”, 10.11.

***ESR?**

The *ESR?, standard event status register query, is defined in “IEEE Std 488.2-1992 “,10.12.

***IDN?**

The *IDN?, identification queryis defined in “IEEE Std 488.2-1992”, 10.14.*IDN? is used to retrieve information about the test set in ASCII format.

*IDN?, returns ASCII codes 32 through 126 excluding comma and semicolon in four comma separated fields. Field 1 returns the manufacturer, field 2 returns the instrument model number, field 3 returns the serial number, field 4 returns 0.

***OPC**

The *OPC, operation complete command, is defined in “IEEE 488.2-1992”, 10.18. *OPC causes the test set to continuously sense the No Operation Pending flag. When the No Operation Pending flag becomes TRUE, the OPC event bit in the standard event status register (ESR) is set to indicate that the state of all pending operations is completed. The *OPC common command is not recommended for use as an overlapped command.

***OPC?**

The *OPC?, operation complete query, is defined in “IEEE Std 488.2-1992”, 10.19. The *OPC? query allows synchronization between the controller and the test set using either the message available (MAV) bit in the status byte, or a read of the output OPC?. The *OPC? query does not effect the OPC event bit in the Standard Event Status Register (ESR). The *OPC? common command is not recommended for use as an overlapped command.

***OPT?**

The *OPT?, option identification query, is defined in “IEEE Std 488.2-1992”, 10.20. Each option will have a unique name, that name will be retuned with the query.

***RST**

The *RST, full preset command, is defined in “IEEE Std 488.2-1992”, 10.32. A full preset is also accomplished

using the SYStem:PRESet2 command. A full preset restores the majority of settings to their default values and sets measurement trigger arm to single.

***SRE**

The *SRE, service request enable command, is defined in “IEEE Std 488.2-1992”, 10.34. The parameter range for this command is 0 through 255.

***SRE?**

The *SRE?, service request enable query, is defined in “IEEE Std 488.2-1992”, 10.35. Values returned by this query range from 0 through 255.

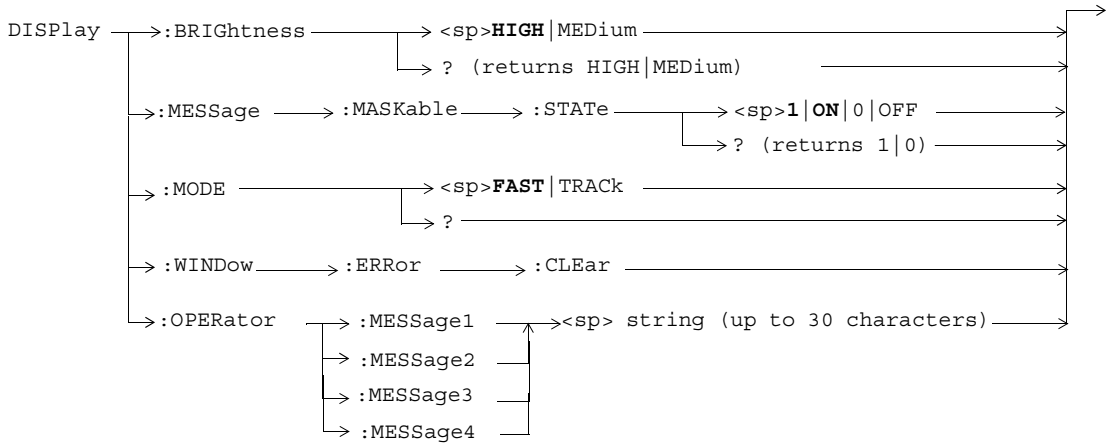
***STB?**

The *STB?, read status byte query, is defined in “IEEE Std 488.2-1992”, 10.36. Values returned by this query range from 0 through 255.

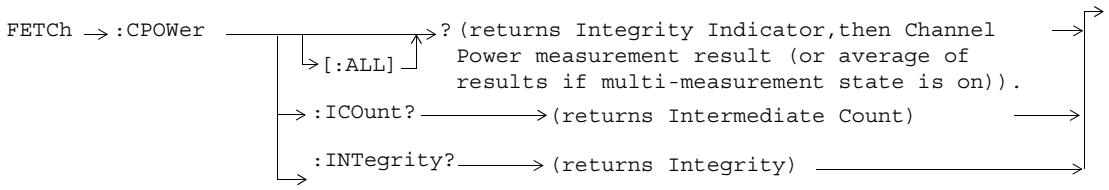
***WAI**

The *WAI, wait-to-continue command, is defined in “IEEE Std 488.2-1992”, 10.39. The *WAI command prevents the test set from executing any further commands or queries until all pending operation flags are false. The *WAI common command is not recommended for use as an overlapped command.

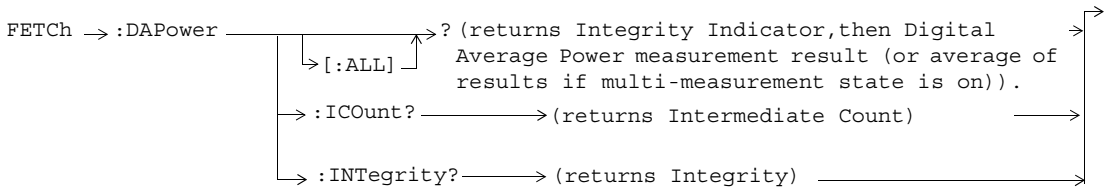
DISPlay


DISPlay
:BRIGhtness
<sp>HIGH|MEDium
? (returns HIGH|MEDium)
:MESSage
:MASKable
:STATe
<sp>1|ON|0|OFF
? (returns 1|0)
:MODE
<sp>FAST|TRACk
?
:WINDow
:ERRor
:CLEar
:OPERator
:MESSage1
:MESSage2
:MESSage3
:MESSage4
<sp> string (up to 30 characters)

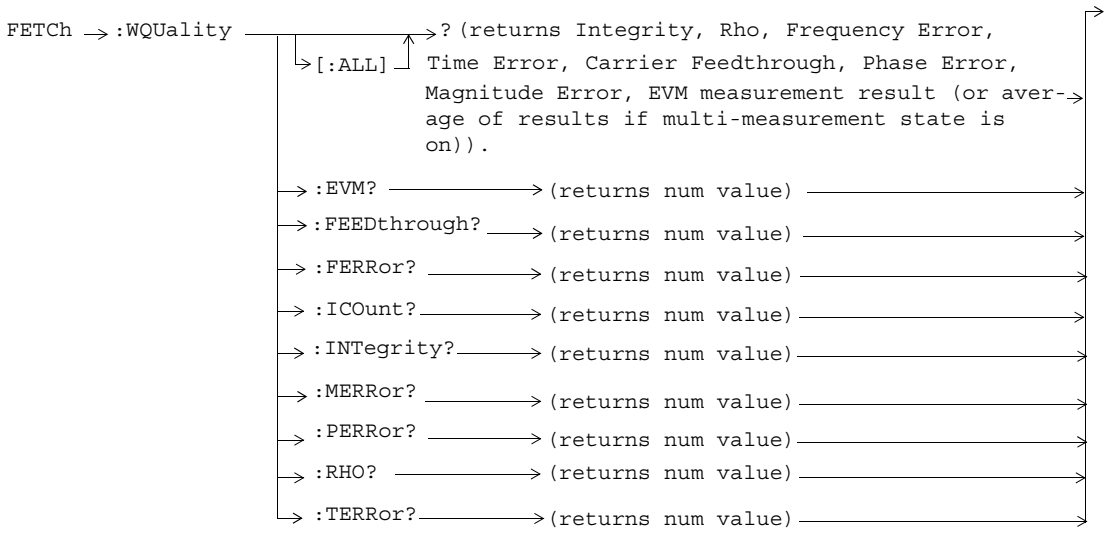
FETCh:CPOWer

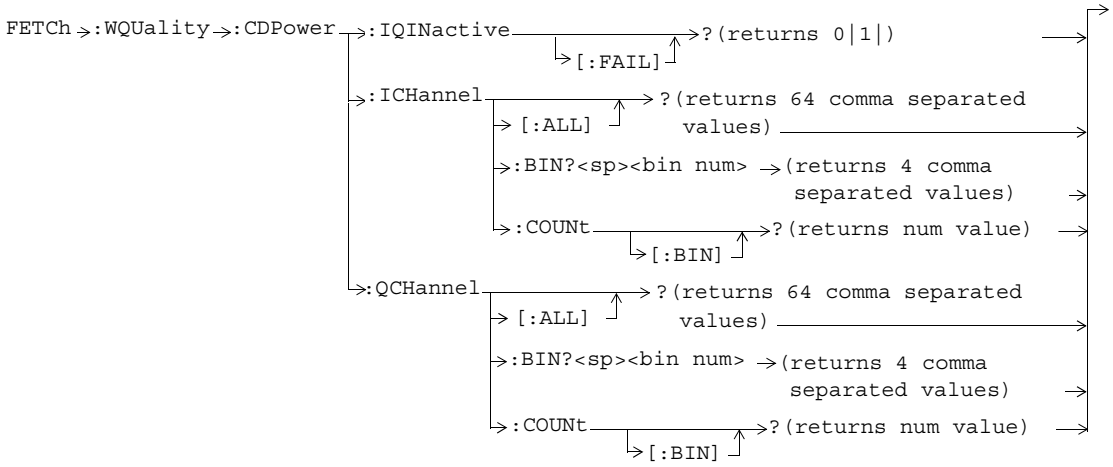


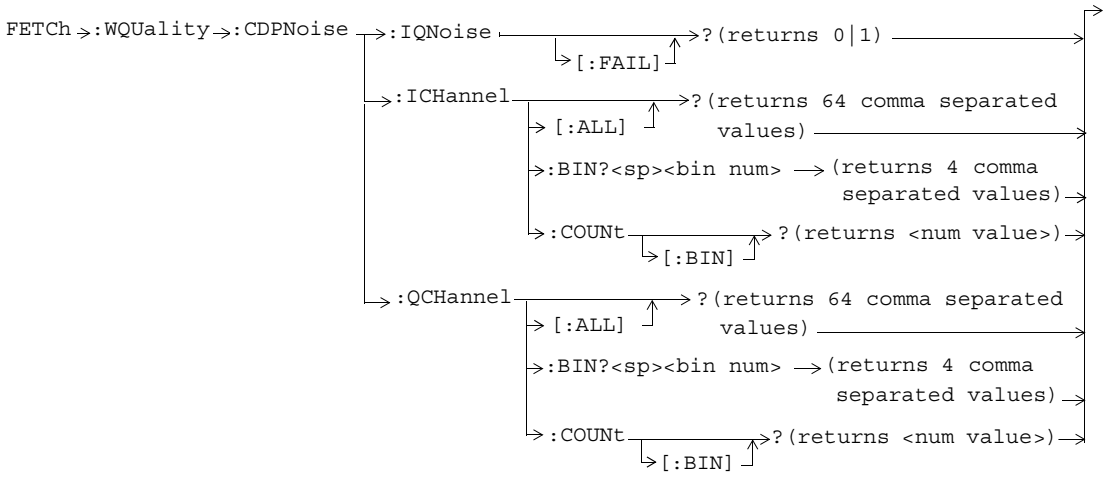
FETCh:DAPower



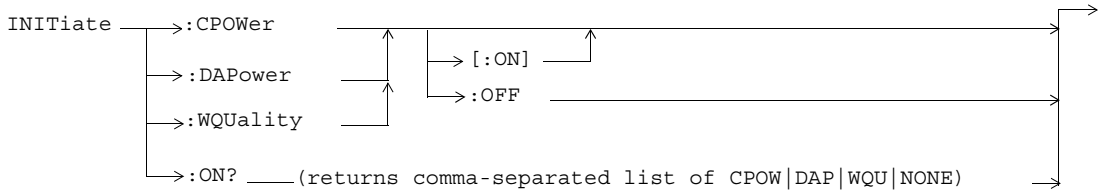
FETCh:WQUality

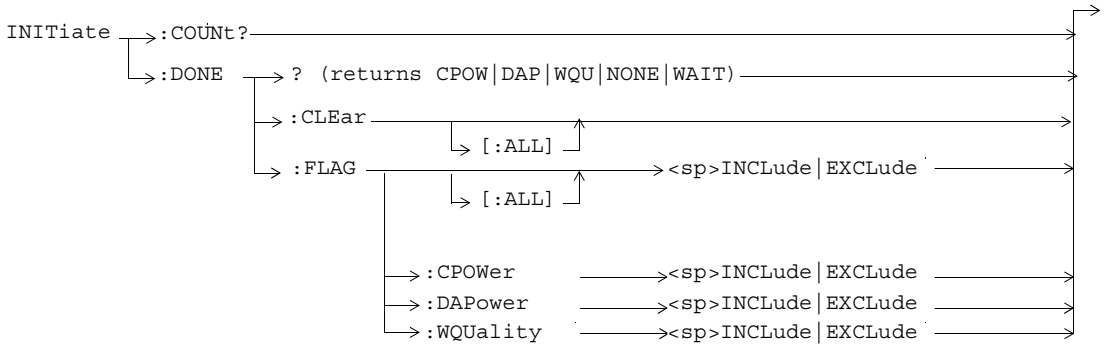



FETCh
:WQUality
:CDPower
:IQINactive
?(returns 0|1|)
[:FAIL]
:ICHannel
?(returns 64 comma separated values)
[:ALL]
:BIN?<sp><bin num>
(returns 4 comma separated values)
:COUNt
?(returns num value)
[:BIN]
:QCHannel
?(returns 64 comma separated values)
[:ALL]
:BIN?<sp><bin num>
(returns 4 comma separated values)
:COUNt
?(returns num value)
[:BIN]

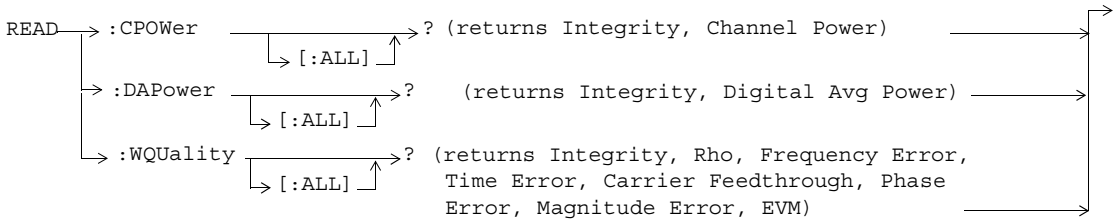

FETCh :WQUality :CDPNoise
:IQNoise
[:FAIL]
?(returns 0|1)
:ICHannel
[:ALL]
?(returns 64 comma separated values)
:BIN?<sp><bin num>
(returns 4 comma separated values)
:COUNt
[:BIN]
?(returns <num value>)
:QCHannel
[:ALL]
?(returns 64 comma separated values)
:BIN?<sp><bin num>
(returns 4 comma separated values)
:COUNt
[:BIN]
?(returns <num value>)

INITiate

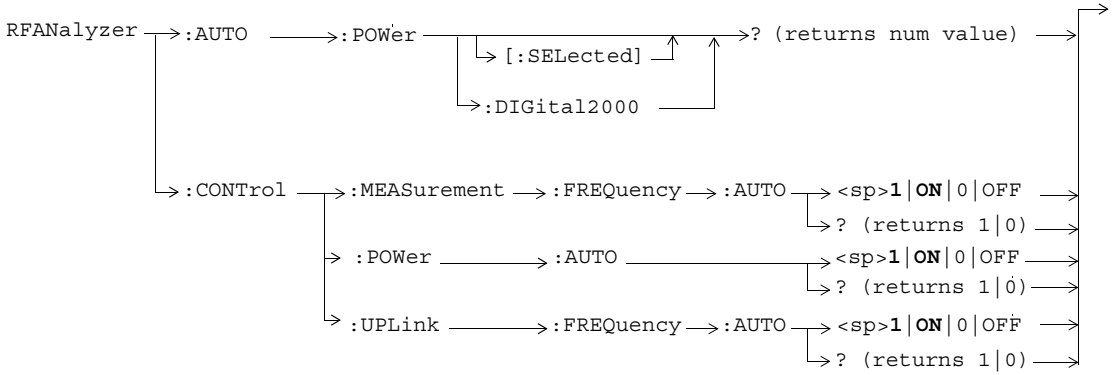

INITiate
:CPOWer
:DAPower
:WQUality
:ON?
[:ON]
:OFF
(returns comma-separated list of CPOW|DAP|WQU|NONE)

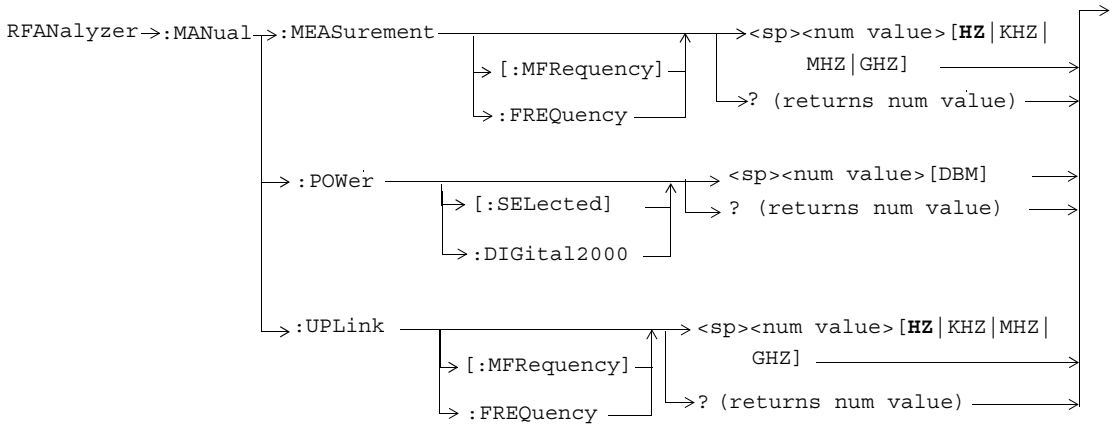

INITiate
:COUNt?
:DONE
? (returns CPOW|DAP|WQU|NONE|WAIT)
:CLEar
[:ALL]
:FLAG
[:ALL]
<sp>INCLude|EXCLude
:CPOWer
:DAPower
:WQUality
<sp>INCLude|EXCLude
<sp>INCLude|EXCLude
<sp>INCLude|EXCLude

READ

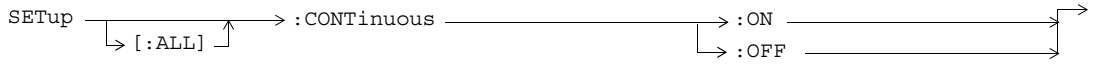

READ
:CPOWer
[:ALL]
? (returns Integrity, Channel Power)
:DAPower
[:ALL]
? (returns Integrity, Digital Avg Power)
:WQUality
[:ALL]
? (returns Integrity, Rho, Frequency Error, Time Error, Carrier Feedthrough, Phase Error, Magnitude Error, EVM)

RFANalyzer

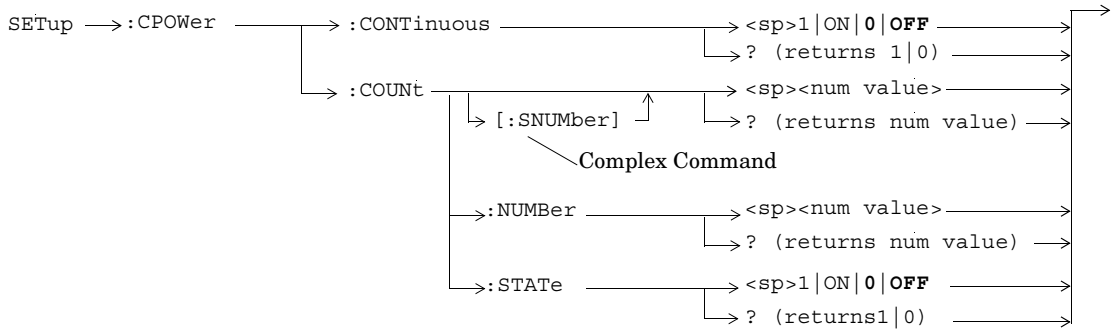

RFANalyzer
:AUTO
:POWer
[:SELected]
:DIGital2000
? (returns num value)
:CONTrol
:MEASurement
:FREQuency
:AUTO
<sp>1|ON|0|OFF
? (returns 1|0)
:POWer
:AUTO
<sp>1|ON|0|OFF
? (returns 1|0)
:UPLink
:FREQuency
:AUTO
<sp>1|ON|0|OFF
? (returns 1|0)

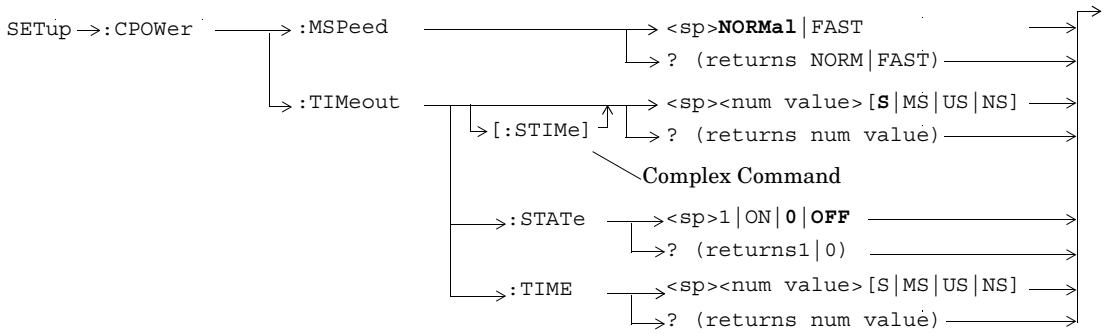

RFANalyzer
:MANual
:MEASurement
[:MFRequency]
:FREQuency
<sp><num value>[HZ|KHZ|MHZ|GHZ]
? (returns num value)
:POWer
[:SELected]
:DIGital2000
<sp><num value>[DBM]
? (returns num value)
:UPLink
[:MFRequency]
:FREQuency
<sp><num value>[HZ|KHZ|MHZ|GHZ]
? (returns num value)

SETup[:ALL]:CONTinuous

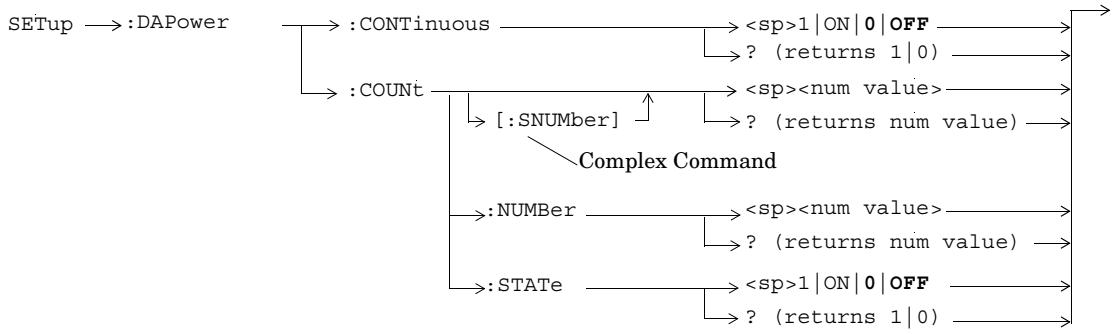


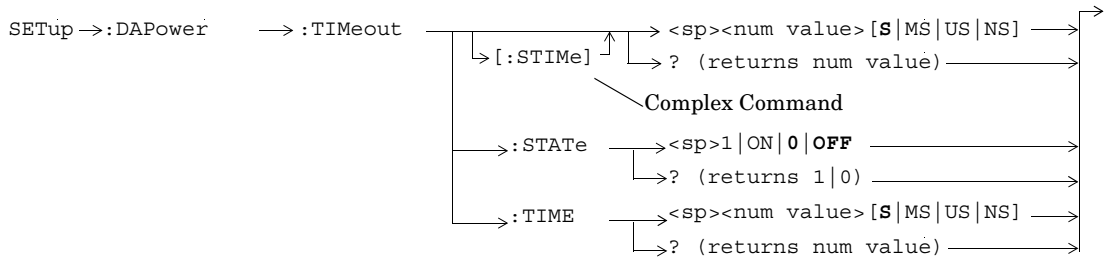
SETup:CPOWer



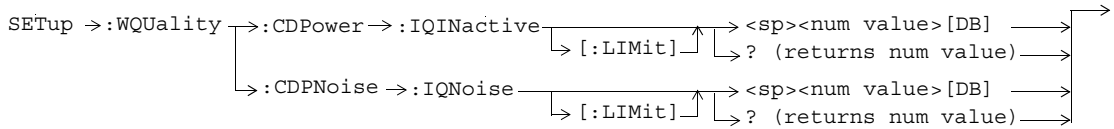

SETup
:CPOWer
:MSPeed
<sp>NORMal|FAST
? (returns NORM|FAST)
:TIMeout
<sp><num value>[S|MS|US|NS]
[:STIMe]
? (returns num value)
Complex Command
:STATe
<sp>1|ON|0|OFF
? (returns1|0)
:TIME
<sp><num value>[S|MS|US|NS]
? (returns num value)

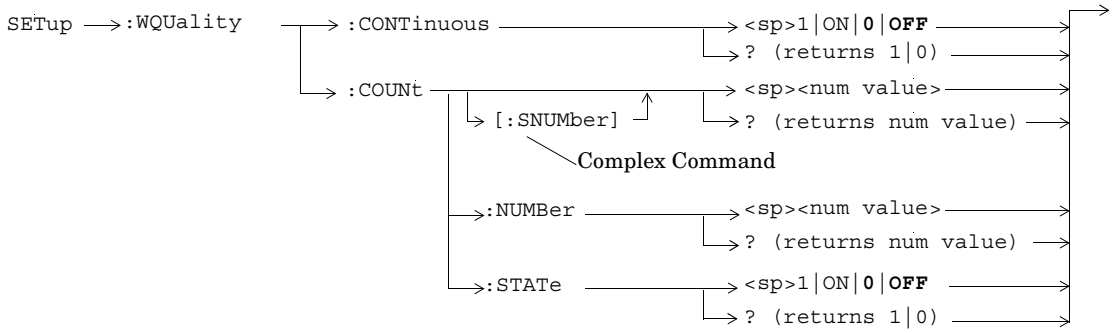
SETup:DAPower

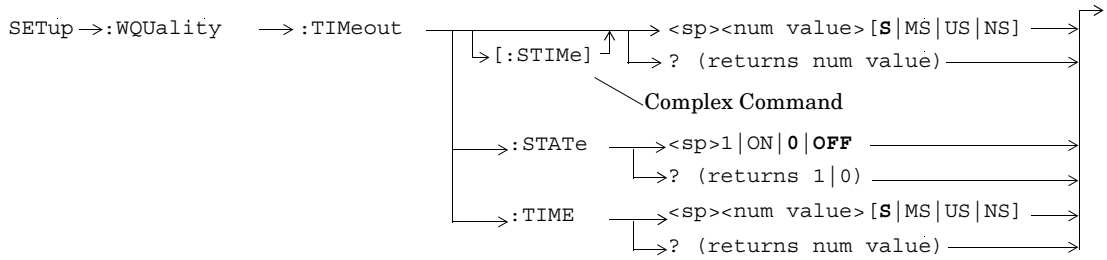



SETup
:DAPower
:TIMeout
[:STIMe]
<sp><num value>[S|MS|US|NS]
? (returns num value)
Complex Command
:STATe
<sp>1|ON|0|OFF
? (returns 1|0)
:TIME
<sp><num value>[S|MS|US|NS]
? (returns num value)

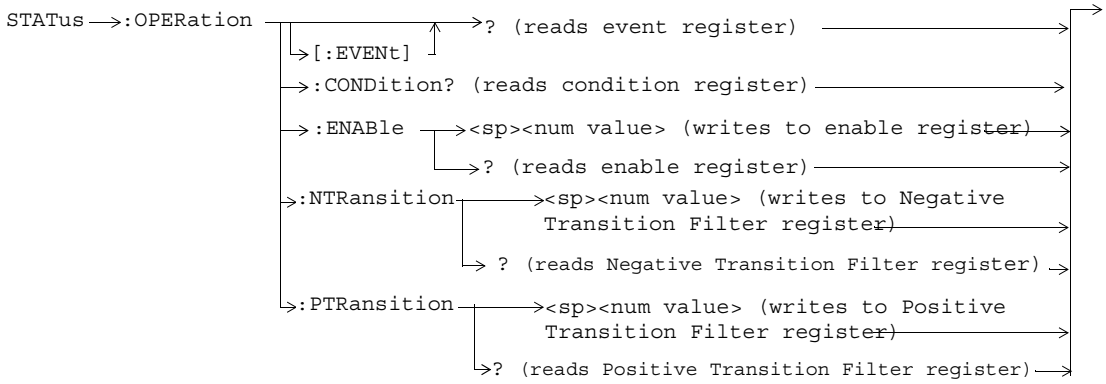
SETup:WQUality

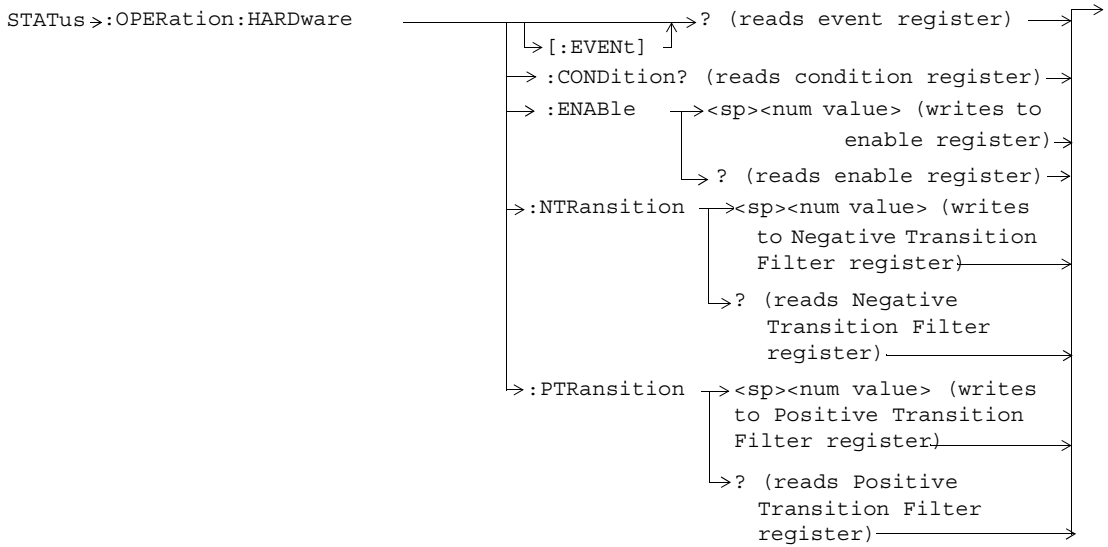


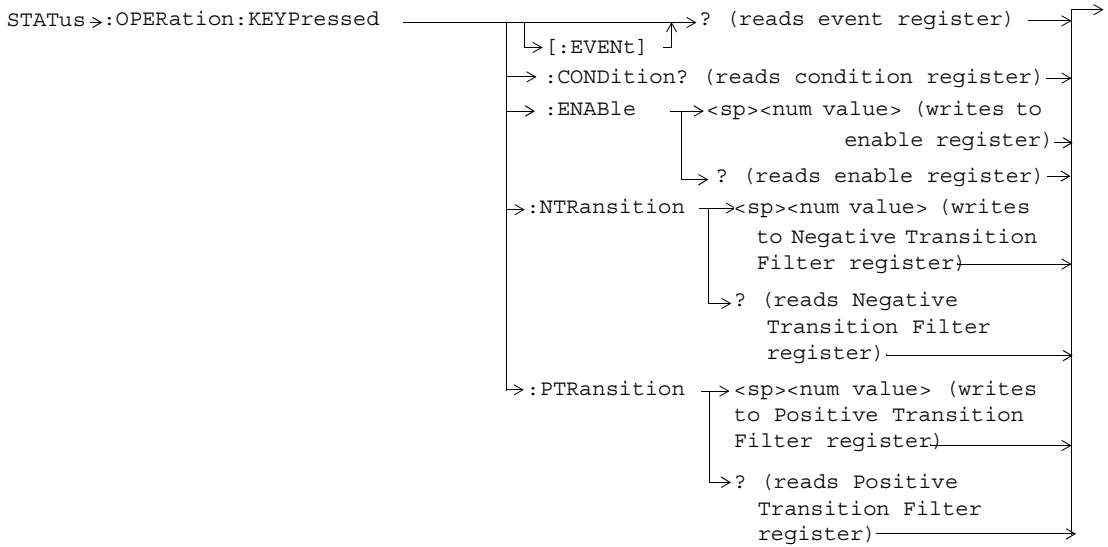

SETup
:WQUality
:CONTinuous
<sp>1|ON|0|OFF
? (returns 1|0)
:COUNt
<sp><num value>
[:SNUMber]
? (returns num value)
Complex Command
:NUMBer
<sp><num value>
? (returns num value)
:STATe
<sp>1|ON|0|OFF
? (returns 1|0)

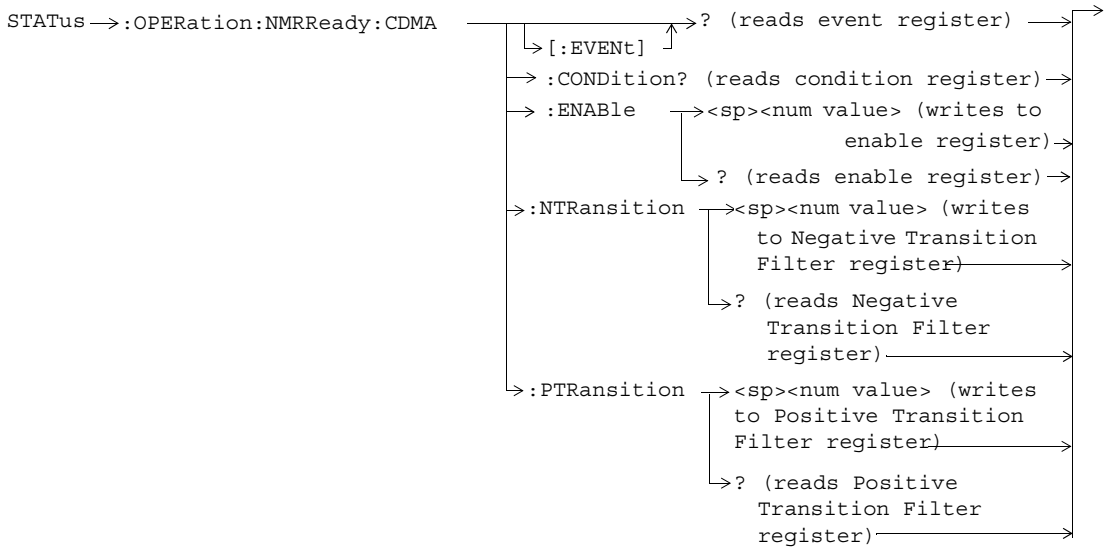

SETup
:WQUality
:TIMeout
[:STIMe]
<sp><num value>[S|MS|US|NS]
? (returns num value)
Complex Command
:STATe
<sp>1|ON|0|OFF
? (returns 1|0)
:TIME
<sp><num value>[S|MS|US|NS]
? (returns num value)

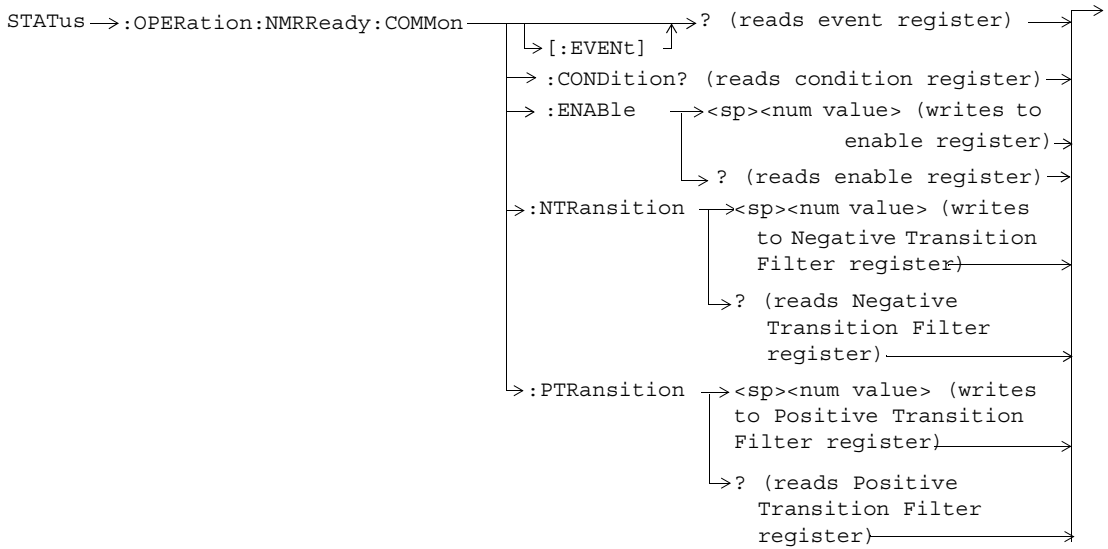
STATus:OPERation:

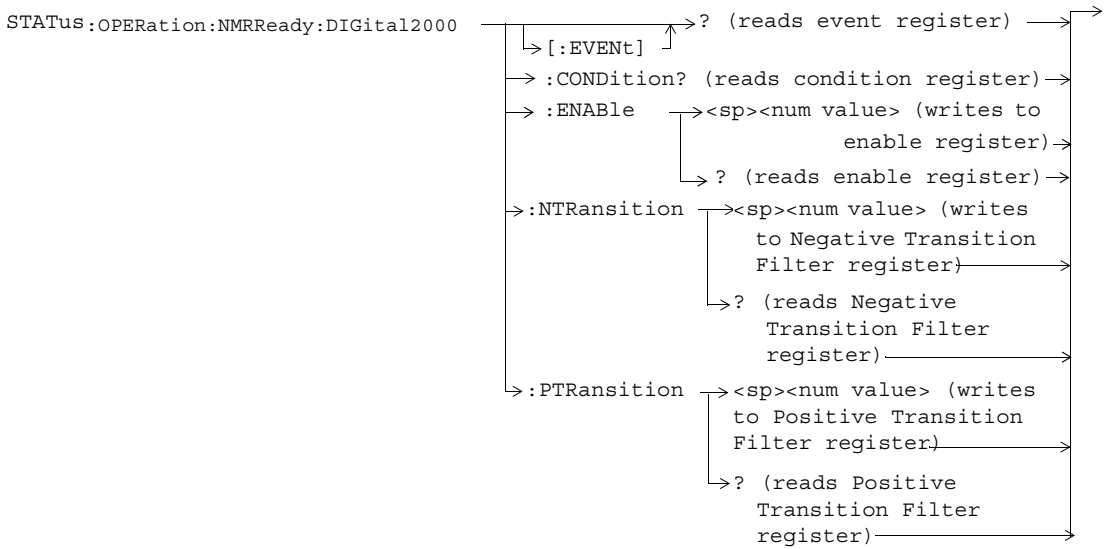


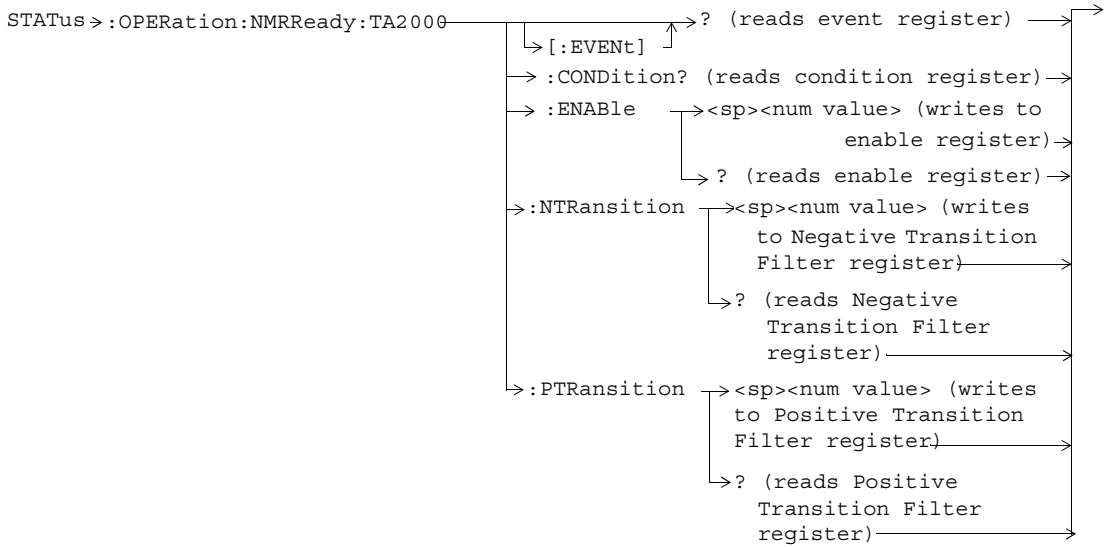

STATus:OPERation:HARDware
? (reads event register)
[:EVENt]
:CONDition? (reads condition register)
:ENABle
<sp><num value> (writes to enable register)
? (reads enable register)
:NTRansition
<sp><num value> (writes to Negative Transition Filter register)
? (reads Negative Transition Filter register)
:PTRansition
<sp><num value> (writes to Positive Transition Filter register)
? (reads Positive Transition Filter register)


STATus:OPERation:KEYPressed
? (reads event register)
[:EVENt]
:CONDition? (reads condition register)
:ENABle
<sp><num value> (writes to enable register)
? (reads enable register)
:NTRansition
<sp><num value> (writes to Negative Transition Filter register)
? (reads Negative Transition Filter register)
:PTRansition
<sp><num value> (writes to Positive Transition Filter register)
? (reads Positive Transition Filter register)

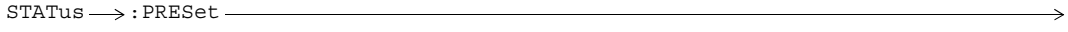

STATus
:OPERation:NMRReady:CDMA
? (reads event register)
[:EVENt]
:CONDition? (reads condition register)
:ENABle
<sp><num value> (writes to enable register)
? (reads enable register)
:NTRansition
<sp><num value> (writes to Negative Transition Filter register)
? (reads Negative Transition Filter register)
:PTRansition
<sp><num value> (writes to Positive Transition Filter register)
? (reads Positive Transition Filter register)


STATus
:OPERation:NMRReady:COMMon
[:EVENt]
? (reads event register)
:CONDition? (reads condition register)
:ENABle
<sp><num value> (writes to enable register)
? (reads enable register)
:NTRansition
<sp><num value> (writes to Negative Transition Filter register)
? (reads Negative Transition Filter register)
:PTRansition
<sp><num value> (writes to Positive Transition Filter register)
? (reads Positive Transition Filter register)

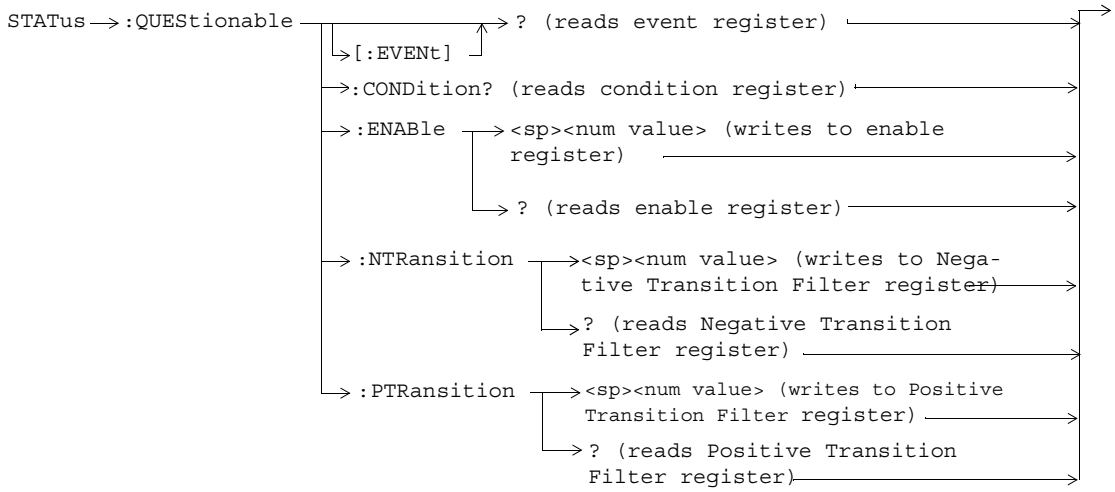

STATus:OPERation:NMRReady:DIGital2000
? (reads event register)
[:EVENt]
:CONDition? (reads condition register)
:ENABle
<sp><num value> (writes to enable register)
? (reads enable register)
:NTRansition
<sp><num value> (writes to Negative Transition Filter register)
? (reads Negative Transition Filter register)
:PTRansition
<sp><num value> (writes to Positive Transition Filter register)
? (reads Positive Transition Filter register)

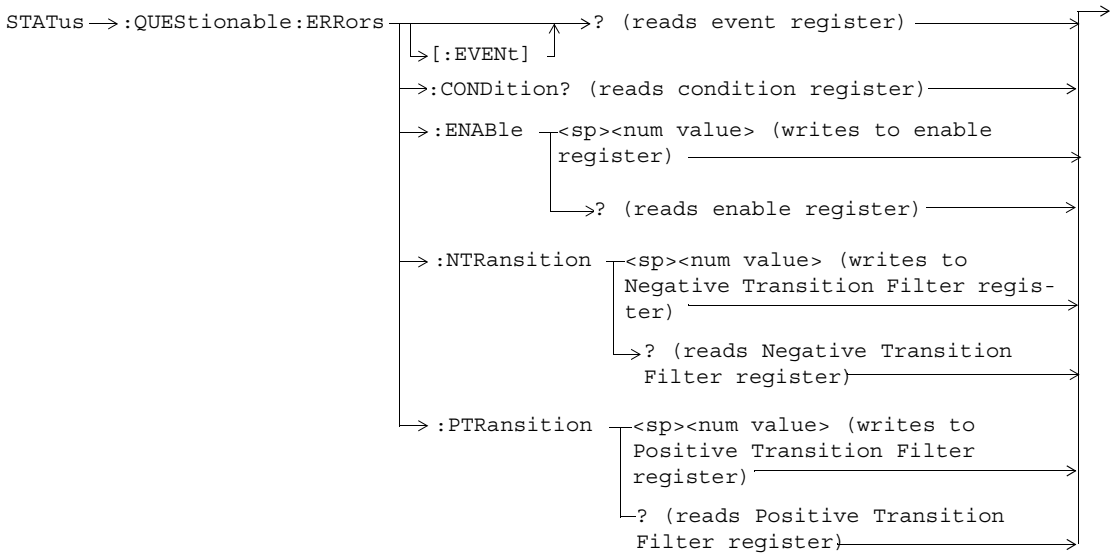

STATus
:OPERation:NMRReady:TA2000
? (reads event register)
[:EVENt]
:CONDition? (reads condition register)
:ENABle
<sp><num value> (writes to enable register)
? (reads enable register)
:NTRansition
<sp><num value> (writes to Negative Transition Filter register)
? (reads Negative Transition Filter register)
:PTRansition
<sp><num value> (writes to Positive Transition Filter register)
? (reads Positive Transition Filter register)

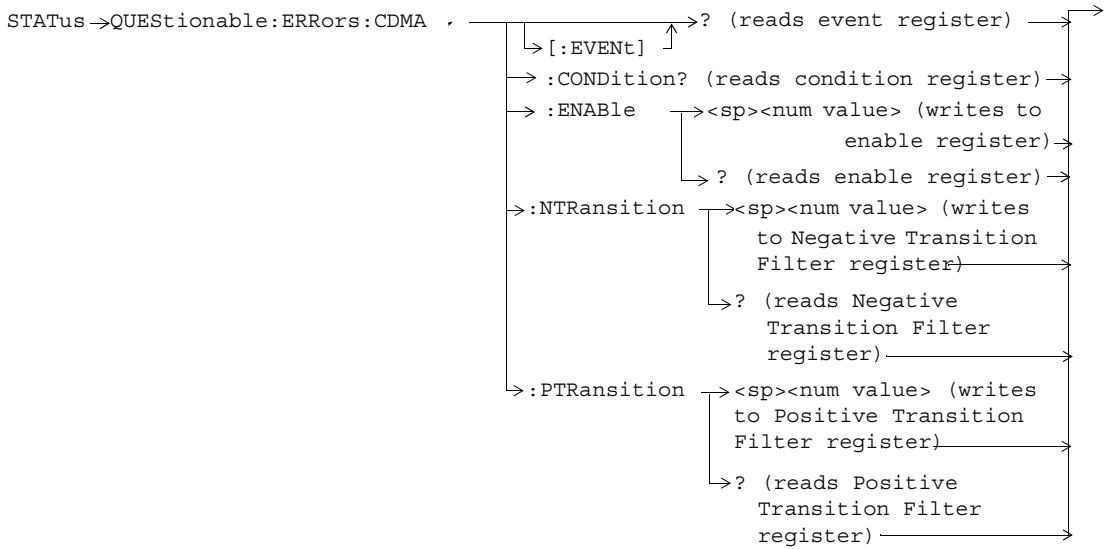
STATus:PRESet

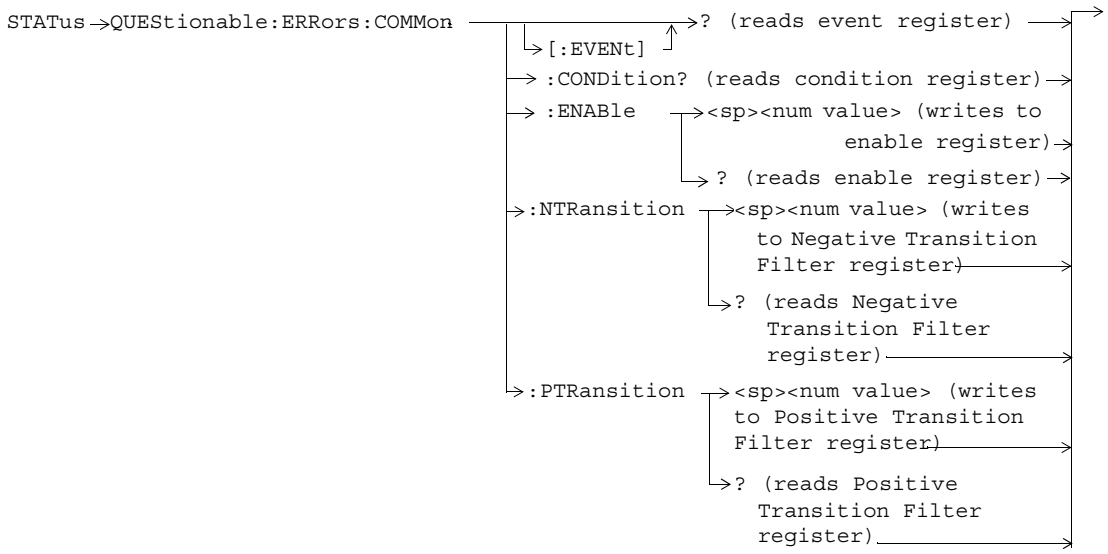


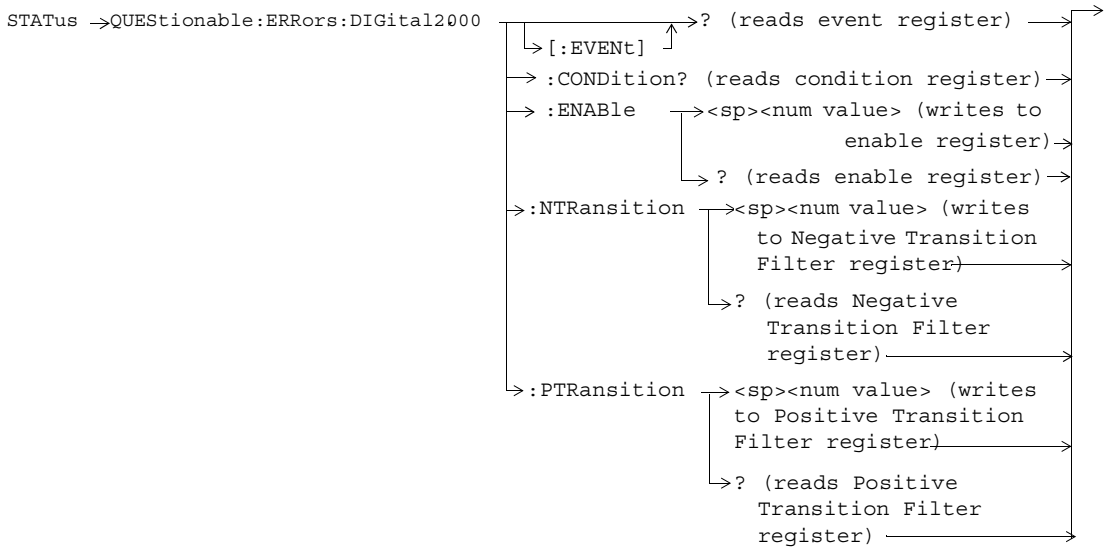
STATus:QUEStionable

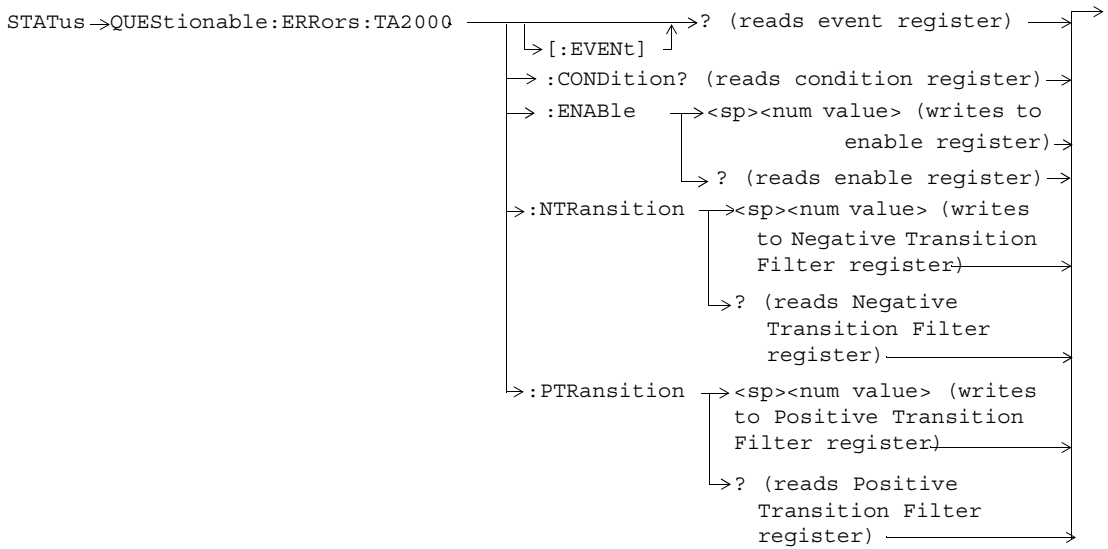


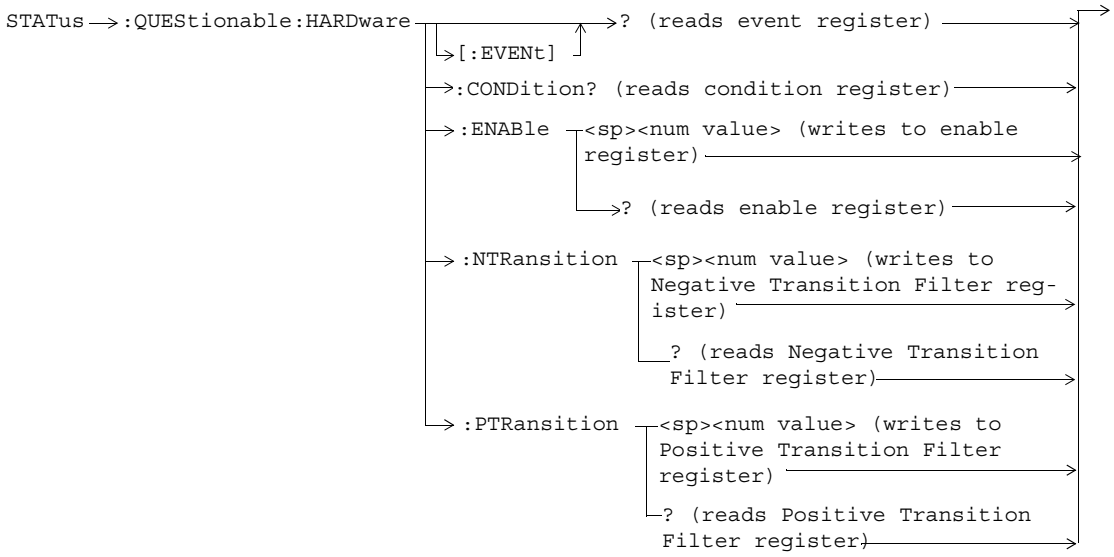

STATus
:QUEStionable:ERRors
? (reads event register)
[:EVENt]
:CONDition? (reads condition register)
:ENABle
<sp><num value> (writes to enable register)
? (reads enable register)
:NTRansition
<sp><num value> (writes to Negative Transition Filter register)
? (reads Negative Transition Filter register)
:PTRansition
<sp><num value> (writes to Positive Transition Filter register)
? (reads Positive Transition Filter register)


STATus
QUEStionable:ERRors:CDMA
,
? (reads event register)
[:EVENt]
:CONDition? (reads condition register)
:ENABle
<sp><num value> (writes to enable register)
? (reads enable register)
:NTRansition
<sp><num value> (writes to Negative Transition Filter register)
? (reads Negative Transition Filter register)
:PTRansition
<sp><num value> (writes to Positive Transition Filter register)
? (reads Positive Transition Filter register)


STATus:QUEStionable:ERRors:COMMon
[:EVENt]
? (reads event register)
:CONDition? (reads condition register)
:ENABle
<sp><num value> (writes to enable register)
? (reads enable register)
:NTRansition
<sp><num value> (writes to Negative Transition Filter register)
? (reads Negative Transition Filter register)
:PTRansition
<sp><num value> (writes to Positive Transition Filter register)
? (reads Positive Transition Filter register)


STATus
QUEStionable:ERRors:DIGital2000
[:EVENt]
? (reads event register)
:CONDition? (reads condition register)
:ENABle
<sp><num value> (writes to enable register)
? (reads enable register)
:NTRansition
<sp><num value> (writes to Negative Transition Filter register)
? (reads Negative Transition Filter register)
:PTRansition
<sp><num value> (writes to Positive Transition Filter register)
? (reads Positive Transition Filter register)


STATus
QUEStionable:ERRors:TA2000
[:EVENt]
? (reads event register)
:CONDition? (reads condition register)
:ENABle
<sp><num value> (writes to enable register)
? (reads enable register)
:NTRansition
<sp><num value> (writes to Negative Transition Filter register)
? (reads Negative Transition Filter register)
:PTRansition
<sp><num value> (writes to Positive Transition Filter register)
? (reads Positive Transition Filter register)


STATus
:QUEStionable:HARDware
[:EVENt]
? (reads event register)
:CONDition? (reads condition register)
:ENABle
<sp><num value> (writes to enable register)
? (reads enable register)
:NTRansition
<sp><num value> (writes to Negative Transition Filter register)
? (reads Negative Transition Filter register)
:PTRansition
<sp><num value> (writes to Positive Transition Filter register)
? (reads Positive Transition Filter register)

```
OUTPUT 714;"STATUS:QUESTIONABLE:EVENT?" !Queries and clears the Questionable Event
                                        !Register
OUTPUT 714;"STATUS:QUESTIONABLE:CONDITION?" !Queries and clears the Questionable Condition
                                           !Register
OUTPUT 714;"STATUS:QUESTIONABLE:ENABLE 1024" !Sets the Questionable Enable Register
                                             !for bit 10
OUTPUT 714;"STATUS:QUESTIONABLE:NTRANSITION 2" !Sets the Questionable Negative
                                               !Transition Filter Register for bit 1
OUTPUT 714;"STATUS:QUESTIONABLE:PTRANSITION 2" !Sets the Questionable Positive
                                               !Transition Filter Register for bit 1


OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:EVENT?" !Queries and clears the Questionable
                                               !Errors Event Register
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:CONDITION?" !Queries and clears the Questionable
                                                  !Errors Condition Register
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:ENABLE 1024" !Sets the Questionable
                                                    !Errors Enable
                                                    !Register for bit 10
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:NTRANSITION 2" !Sets the Questionable Errors
                                                      !Negative Transition Filter Register
                                                      !for bit 1
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:PTRANSITION 2" !Sets the Questionable Errors
                                                      !Positive Transition Filter Register
                                                      !for bit 1
```

```
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:CDMA:EVENT?" !Queries and clears the Questionable
                                                    !Errors CDMA Event Register
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:CDMA:CONDITION?" !Queries and clears the Questionable
                                                        !Errors CDMA Condition Register
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:CDMA:ENABLE 1024" !Sets the Questionable
                                                         !Errors CDMA Enable
                                                         !Register for bit 10
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:CDMA:NTRANSITION 2" !Sets the Questionable Errors
                                                           !CDMA Negative Transition
                                                           !Filter Register for bit 1
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:CDMA:PTRANSITION 2" !Sets the Questionable Errors
                                                           !CDMA Positive Tranasition
                                                           !Filter Register for bit 1

OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:COMMON:EVENT?" !Queries and clears the Questionable
                                                      !Errors Common Event Register
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:COMMON:CONDITION?" !Queries the and clears the
                                                          !Questionable Errors
                                                          !Common Condition Register
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:COMMON:ENABLE 1024" !Sets the Questionable
                                                           !Errors Common Enable
                                                           !Register for bit 10
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:COMMON:NTRANSITION 2" !Sets the Questionable Errors
                                                             !Common Negative Transition
                                                             !Register for bit 1
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:COMMON:PTRANSITION 2" !Sets the Questionable Errors
```

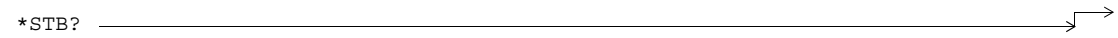
```
                                                              !Common Positive Tranasition
                                                              !Register for bit 1
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:DIGITAL2000:EVENT?" !Queries and clears the
                                                         !Questionable Errors
                                                         !DIGITAL2000 Event Register
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:DIGITAL2000:CONDITION?" !Queries and clears the
                                                             !Qestionable Errors
                                                             !DIGITAL2000 Condition Register
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:DIGITAL2000:ENABLE 1024" !Sets the Questionable
                                                              !Errors DIGITAL2000 Enable
                                                              !Register for bit 10
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:DIGITAL2000:NTRANSITION 2" !Sets the Questionable
                                                                !Errors DIGITAL2000 Negative
                                                                !Transition Filter
                                                                !Register for bit 1
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:DIGITAL2000:PTRANSITION 2" !Sets the Questionable
                                                                !Errors DIGITAL2000 Positive
                                                                !Transition Filter
                                                                !Register for bit 1
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:TA2000:EVENT?" !Queries and clears the
                                                             !Questionable Errors
                                                             !TA2000 Event Register
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:TA2000:CONDITION?" !Queries and clears the
                                                                 !Qestionable Errors
```

```
                                                                !TA2000 Condition Register
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:TA2000:ENABLE 1024" !Sets the Questionable
                                                                 !Errors TA2000 Enable
                                                                 !Register for bit 10
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:TA2000:NTRANSITION 2" !Sets the Questionable
                                                                    !Errors TA2000 Negative
                                                                    !Transition Filter
                                                                    !Register for bit 1
OUTPUT 714;"STATUS:QUESTIONABLE:ERRORS:TA2000:PTRANSITION 2" !Sets the Questionable
                                                                    !Errors TA2000 Positive
                                                                    !Transition Filter
                                                                    !Register for bit 1


OUTPUT 714;"STATUS:QUESTIONABLE:HARDWARE:EVENT?" !Queries and clears the Questionable
                                                 !Hardware Event Register
OUTPUT 714;"STATUS:QUESTIONABLE:HARDWARE:CONDITION?" !Queries and clears the Questionable
                                                     !Hardware Condition Register
OUTPUT 714;"STATUS:QUESTIONABLE:HARDWARE:ENABLE 1024" !Sets the Questionable
                                                      !Hardware Enable
                                                      !Register for bit 10
OUTPUT 714;"STATUS:QUESTIONABLE:HARDWARE:NTRANSITION 2" !Sets the Questionable
                                                        !Hardware Negative Transition Filter
                                                        !Register for bit 1
OUTPUT 714;"STATUS:QUESTIONABLE:HARDWARE:PTRANSITION 2" !Sets the Questionable
                                                        !Hardware Positive Transition Filter
                                                        !Register for bit 1
```

Status Byte Register

*STB?



NOTE The Status Byte Register can also be read with a serial poll. For example, the command “Status_byte = SPOLL(714)” would perform a serial poll of the Status Byte Register, returning and releasing RQS (bit 6).

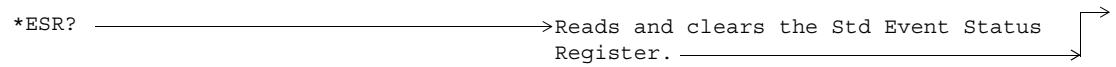
Status Byte Register Bit Assignments

Bit Number	Binary Weighting	Label	Description
7	128	STATus: OPERation	Summarizes the STATus: OPERation Status Register, which fans out to the NMRReady and CALL Status Registers.

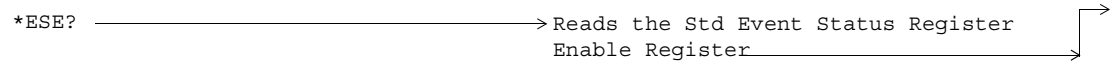
Bit Number	Binary Weighting	Label	Description
6	64	RQS (SRQ TRUE?)/Master Summary Status	RQS is read by a serial poll (SPOLL) Master Summary Status is read by a *STB? query - defined by IEEE 488.2
5	32	Standard Event Status Register	Summarizes the Standard Event Status Register
4	16	Message Available	SCPI - Defined
3	8	STATus: QUEStionable Status Register	Summary Message comes from the STATus: QUEStionable Status Register, which fans out to the CALL and HARDware Status Registers
2	4	Error/ Event Queue	SCPI - Defined
1	2	Reserved	
0	1	Reserved	

Standard Event Status Register

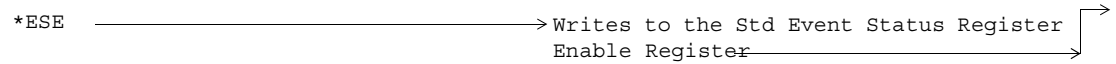
***ESR?**



***ESE?**



***ESE**



“Diagram Conventions” on page 7

Standard Event Status Register Bit Assignment

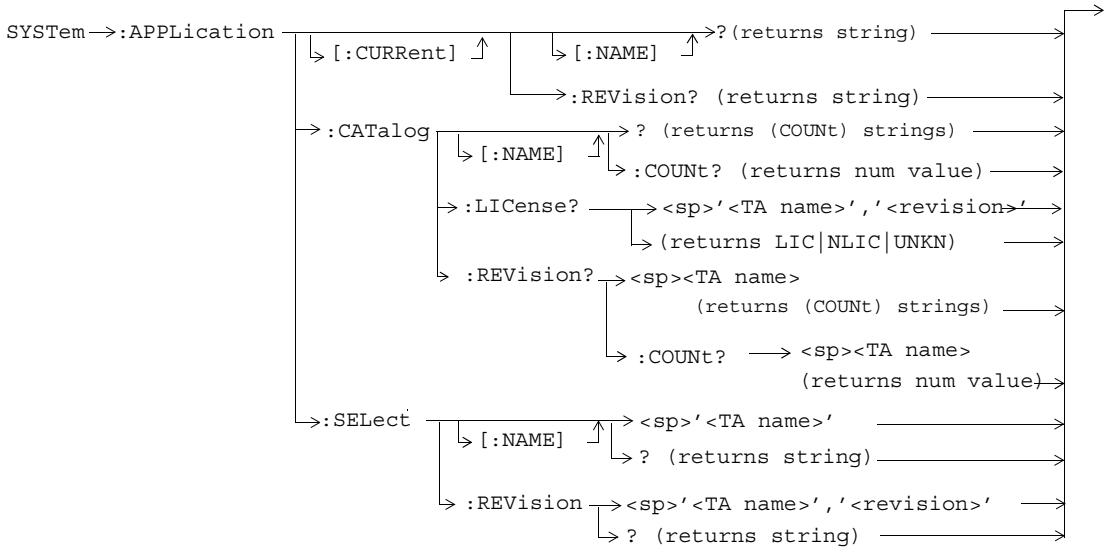
Bit Number	Binary Weighting	Condition	Description
15	32768	Reserved by IEEE.	This bit will always be 0.
14	16384	Reserved by IEEE.	This bit will always be 0.
13	8192	Reserved by IEEE.	This bit will always be 0.
12	4096	Reserved by IEEE.	This bit will always be 0.
11	2048	Reserved by IEEE.	This bit will always be 0.
10	1024	Reserved by IEEE.	This bit will always be 0.
9	512	Reserved by IEEE.	This bit will always be 0.
8	256	Reserved by IEEE.	This bit will always be 0.
7	128	Power On	This bit is set to 1 if the power supply has been turned off and on since the last time this register was read or otherwise cleared. Defined in "IEEE Std. 488.2-1992",11.5.1.1.2
6	64	Reserved for future use.	This bit will always be 0.

Bit Number	Binary Weighting	Condition	Description
5	32	Command Error	This bit is set to 1 if the test set detects an error while trying to process a command. The following events cause a command error: • An IEEE 488.2 syntax error. The test set received a message that did not follow the syntax defined by the standard. • A semantic error. For example the test set received an incorrectly spelled command. • The test set received a group execution trigger (GET) inside a program message

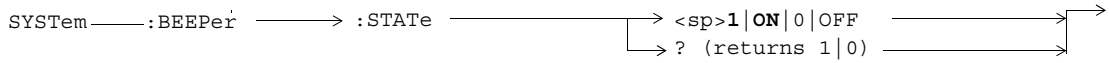
Bit Number	Binary Weighting	Condition	Description
4	16	Execution Error	This bit is set to 1 if the test set detects an error while trying to execute a command. The following events cause a execution error: • A <PROGRAM DATA> element received in a command is outside the legal range for the test set, or it is inconsistent with the operation of the test set. • The test set could not execute a valid command due to some test set hardware/firmware condition.
3	8	Device Dependent Error	This bit is set to 1 if a test set operation does not execute properly due to an internal condition (such as, overrange). This bit indicates that the error was not a command, query, or execution error.

Bit Number	Binary Weighting	Condition	Description
2	4	Query Error	This bit is set to 1 if an error has occurred while trying to read the test set's output queue. The following events cause a query error: • An attempt is made to read data from the output queue when no data is present or is pending. • Data in the output queue has been lost. An example of this would be an output queue overflow.
1	2	Reserved for future use.	This bit will always be 0.
0	1	Operation Complete	This bit is set to 1 when the test set has completed all pending operations and is ready to accept new commands. This bit is only generated in response to the *OPC IEEE 488.2 common command.

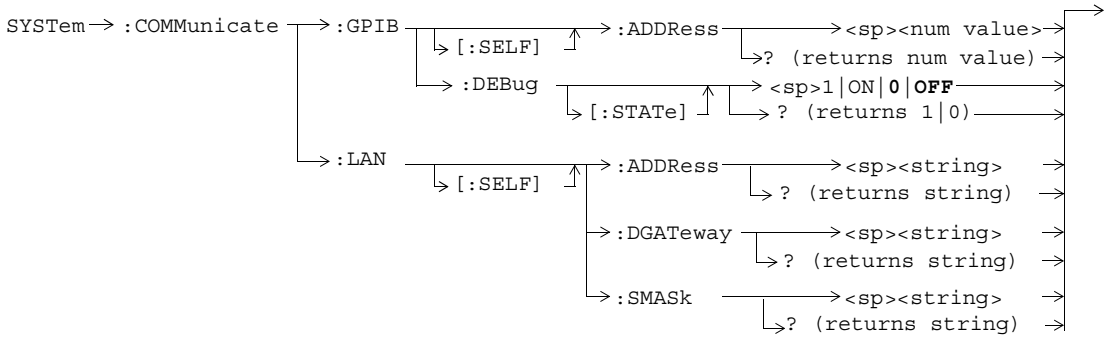
SYSTem:APPLication



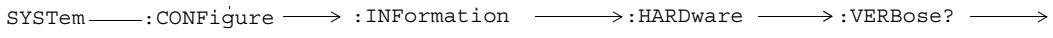
SYSTem:BEEPer



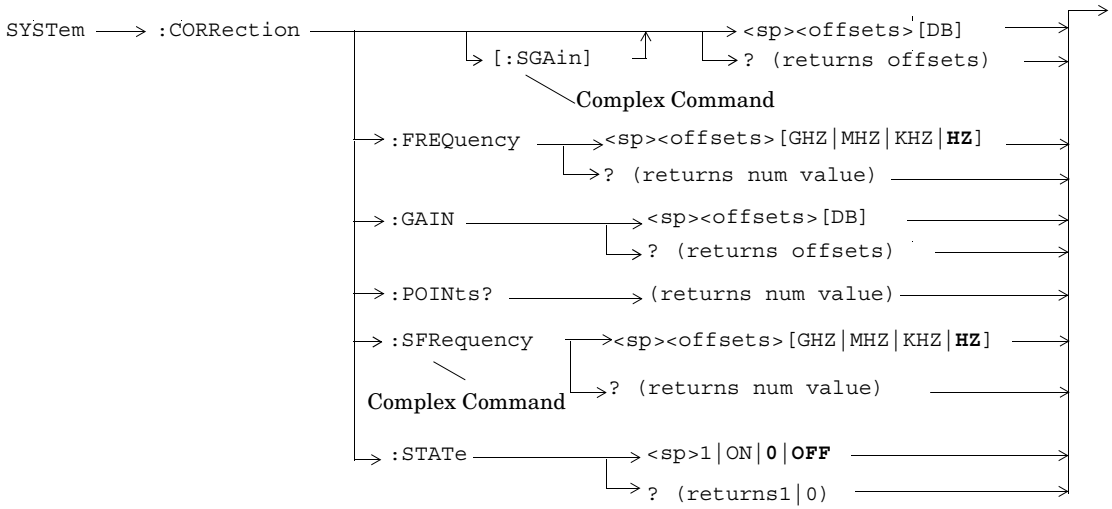
SYSTem:COMMunicate



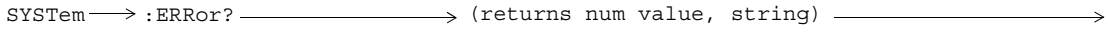
SYSTem:CONFigure:INFormation:HARDware:VERBose?



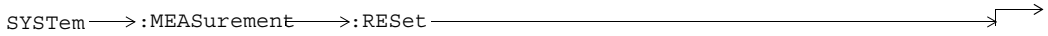
SYSTem:CORRection



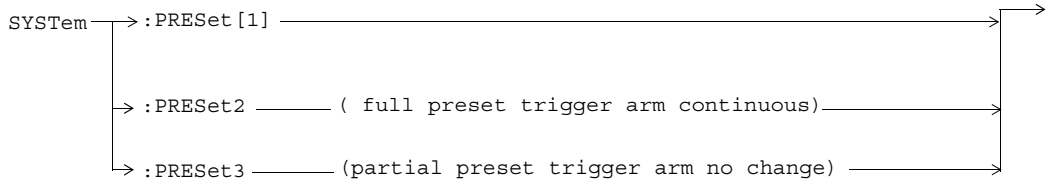
SYSTem:ERRor?



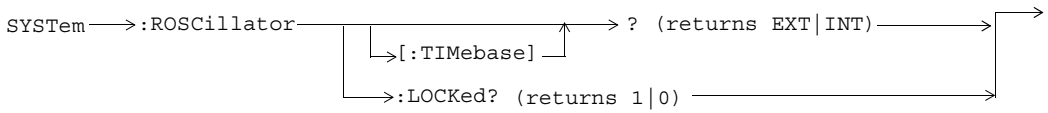
SYSTem:MEASurement:RESet



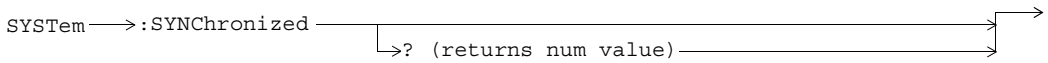
SYSTem:PRESet



SYSTem:ROSCillator



SYSTem:SYNChronized



Syntax Equivalents

To find a syntax equivalent for a field on the Test Set's display.

1. Find the field name on the Test Set's display.

2. Look up the name in the alphabetical listing.

3. Turn to the page indicated.

Syntax Equivalents