ADVANTEST®
ADVANTEST CORPORATION

R3681 Series
Programming Guide

MANUAL NUMBER FFE-8440092H00

Applicable Models
R3681
R3671

First printing April 25, 2003
Printed in Japan

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

This chapter describes the outline of this manual for efficient utilization.

1.1 Outline of This Manual

This manual is a program creation guide for the R3681 Series Signal Analyzer. This manual is organized so that the user can learn the method of creating programs that perform remote control of the R3681 Series Signal Analyzer by using SCPI commands.

This manual assumes that the user is familiar with the manual operation of the R3681 Series Signal Analyzer. Therefore, this manual does not include a detailed explanation about the operation method and functions of the R3681 Series Signal Analyzer. For more information on the operation method and functions, refer to the User's Guide.

The outline of each chapter is shown below:

Chapter 1 "INTRODUCTION"	This chapter describes the outline of this manual for efficient utilization.
Chapter 2 "OVERVIEW OF REMOTE CONTROL"	This chapter describes the overview of the remote control system and SCPI commands.
Chapter 3 "MEASUREMENT PROCEDURES"	This chapter describes the methods of setting the measurement conditions, performing measurement, and reading measurement data.
Chapter 4 "PROGRAM EXAMPLES"	The following measurement program examples are explained: • Marker setting and marker reading • Frequency measurement using the frequency counter • Channel Power measurement
Chapter 5 "SCPI COMMAND REFERENCE"	SCPI command reference. The command reference explains the commands in order of function. In explanation, the following items are explained: • Command format • Function explanation • Parameters • Query response • Use example • Related commands

1.2 Other Manuals for This Instrument

The following manuals are provided for this instrument:

- User's Guide (Product code: **{ER3681SERIES/U}**, English)

  This manual describes the information required to use the R3681 Series. It includes setup, basic operation, application measurement, function explanation, specifications, and maintenance.

- Programming Guide (Product code: **{ER3681SERIES/P}**, English, This manual}

  This manual describes information related to programming for automatic measurement of the R3681 Series. It includes an overview of remote control, SCPI command reference, and application program examples.

- Performance Verification Guide (Product code: **{ER3681SERIES/T}**, English)

  This manual describes the information required to verify the performance of the R3681 Series. It includes the performance test procedure and specifications.

2. OVERVIEW OF REMOTE CONTROL

This chapter describes the overview of the remote control system and SCPI commands.

2.1 Types of Remote Control Systems

The following two types of remote control systems can be configured, depending on the interface:

Interface	Overview
GPIB (Talker/Listener mode)	This system controls the R3681 Series and other devices connected from the external controller through GPIB. For more information, refer to "2.2 GPIB Remote Control System" (on page 2-1).
LAN	This system controls the R3681 Series and other devices connected from the external controller through LAN. For more information, refer to "2.3 LAN Remote Control System" (on page 2-6).

2.2 GPIB Remote Control System

The GPIB (General Purpose Interface Bus) that is compliant with IEEE standards 488.1-1978 and 488.2-1987 comes standard with this instrument so that remote control can be performed from the external controller.

The controlling method using the GPIB remote control function is explained below.

2.2.1 What is the GPIB?

The GPIB (General Purpose Interface Bus) is a high performance bus that integrates computers and measuring instruments.

Operation of the GPIB is defined by IEEE standard 488.1-1978. Since the GPIB has bus structure interfaces, a specific device can be specified by assigning a unique device address to each device. Up to 15 devices can be connected to one bus in parallel. A GPIB device is equipped with at least one of the following functions:

- Talker

  The device that is set to send data to the bus is called the "talker". On the GPIB bus, only one device acts as an active talker.

- Listener

  The device that is set to receive data on the bus is called the "listener". Two or more active listener devices can exist on a GPIB bus.

- Controller

  The device that specifies talkers and listeners is called the "controller". On a GPIB bus, only one device acts as an active controller. Of these controllers, the device that can control IFC and REN messages is expressly called the "system controller".

  Only one system controller is permitted on a GPIB bus. If there are two or more controllers on a bus, the system controller becomes the active controller at the time of system startup and the other devices with controller capability act as addressable devices.

  To set another controller to an active controller, use Take Control (TCT) interface messages. At this time, this controller becomes a non-active controller.

  The controller controls the entire system by sending interface or device messages to each measuring instrument. The roles of these messages are shown below.

  - Interface message: Controls the GPIB bus.
  - Device message: Controls the measuring instruments.

2.2.2 Setting up the GPIB

1. GPIB connection

   The standard GPIB connection is shown below. Fix the GPIB connector firmly with two screws such that they do not loosen during use.

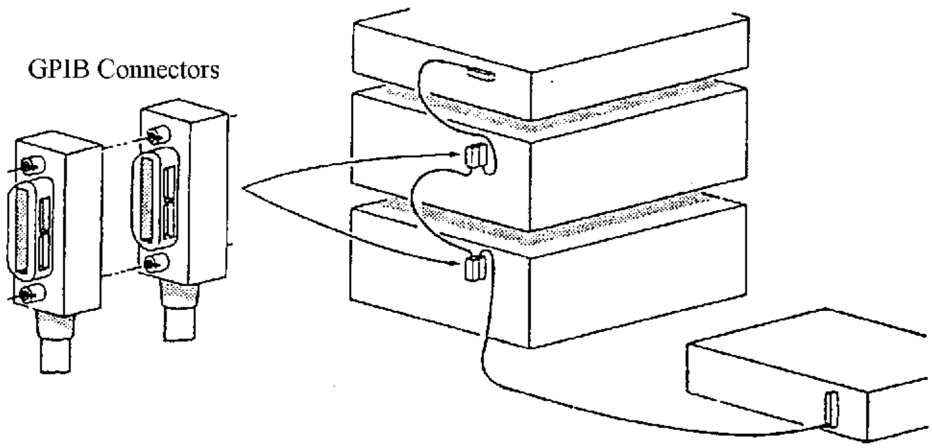


Figure 2-1 GPIB Connection

Note the following when using the GPIB interface:

- Connect the GPIB cable to the GP-IB 1 connector on the rear panel of this instrument.
- The total cable length of the GPIB cable used in one bus system is not longer than 2 m × {the number of connected devices (the GPIB controller is counted as one device)}.
  The total cable length should be 20 m or less.
- Up to 15 devices can be connected to one bus system.

- There is no restriction in the method of connecting cables. However, four or more GPIB connectors should not be stacked on one device. If four or more GPIB connectors are stacked, the joints of the connectors may be broken because excessive force is applied to them.

  For example, a system consisting of five devices can use cables of up to 10 m (2 m/device × 5 devices = 10 m) in total. Cable lengths can be allocated freely unless the total cable length exceeds the permitted maximum length. When 10 or more devices are to be connected, however, some devices should be connected with cables of 2 m or less so that the total cable length does not exceed 20 m.

2. GPIB address setting

   GPIB addresses should be set from the GPIB dialog box in the System menu.

2.2.3 GPIB Bus Functions

2.2.3.1 GPIB Interface Functions

Table 2-1 GPIB Interface Functions

Code	Description
SH1	Has the source handshake function
AH1	Has the accepter handshake function
T6	Basic talker function, serial polling function, listener-specified talker cancel function
TE0	No extended talker function
L4	Basic listener function, talker-specified listener cancel function
LE0	No extended listener function
SR1	Has service request function
RL1	Remote function, local function, local lockout function
PP0	No parallel polling function
DC1	Device clear function
DT1	Device trigger function
C1	System controller function
C2	IFC transmission, controller-in-charge function
C3	REN transmission function
C4	SRQ response function
C12	Interface message transmission function, pass control back function
E1	Using the open-collector bus driver

2.2.3.2 Responses to Interface Messages

The responses of this instrument to interface messages explained in this section are defined in IEEE standards 488.1-1987 and 488.2-1987.

For information on the method of sending interface messages to this instrument, refer to the operation manual of the controller used.

1. Interface clear (IFC)

   This message is directly sent to this instrument through a signal line.

   With this message, this instrument stops the operation of the GPIB bus. Though all input/output is stopped, the I/O buffer is not cleared (it is cleared by DCL). When this instrument is defined as an active controller, the control right of the GPIB bus is canceled and the system controller gets the control right.

2. Remote enable (REN)

   This message is directly sent to this instrument through a signal line.

   If this instrument is specified as a listener when this message is TRUE, it enters the remote state.

   This state continues until this instrument receives GTL, REN is changed to FALSE, or the LOCAL key is pressed.

   This instrument ignores all the received data when it is in the local state.

   When it is in the remote state, this instrument ignores all key entry except the LOCAL key.

   When it is in local lockout state (refer to "Local lockout (LLO)"), this instrument ignores all key entry.

3. Serial port enable (SPE)

   When receiving this message from outside, this instrument enters the serial polling mode.

   When this instrument specified as a talker in this mode, it sends status bytes instead of ordinary messages. This mode continues until this instrument receives a serial polling disable (SPD) message or an IFC message.

   When this instrument is sending a service request (SRQ) message to the controller, bit6 (RQS bit) of response data is set to 1 (TRUE). After transmission is completed, RQS bit is set to 0 (FALSE). A service request (SRQ) message is directly sent through a signal line.

4. Group execute trigger (GET)

   This message triggers this instrument. If the following conditions are met, this instrument starts measurement.

   - The trigger source is set to the GPIB bus. (TRIG: SOUR BUS)
   - This instrument is in trigger waiting state. (Refer to "Trigger system")

   GET performs the same operation as *TRG, but not the same operation as TRIG:IMM and TRIG:SIG.

   GET, *TRG, TRIG:IMM, and TRIG:SIG are executed in the order they are stacked and received on the input buffer.

5. Device clear (DCL)

   When receiving DCL, this instrument performs the following operations:

   - Clearing the input and output buffers
   - Resetting the syntax analysis, execution control, and response data generation units
   - Canceling all the commands that impede the remote command to be executed next
   - Canceling the command that is temporarily stopped to wait for other parameters

- Canceling *OPC and *OPC?

The following operations are not executed:

- Changing data set or stored in this instrument
- Interrupting the front panel operation
- Affecting or interrupting the operation of this instrument during execution
- Changing the status byte except MAV (MAV is set to 0 as the result of clearing the output buffer)

6. Selected device clear (SDC)

   Performs the same operation as DCL. However, SDC is executed only when this instrument is a listener.
   In other cases, it is ignored.

7. Go to local (GTL)

   This message sets this instrument to the local state. In the local state, all the front panel operations are enabled.

8. Local lockout (LLO)

   This message sets this instrument to the local lockout state. When this instrument enters the remote state in this state, all front panel operation is prohibited (In the ordinary remote state, the front panel operation can be performed with the LOCAL key).

   In this case, this instrument can be set to the local state by any of the following three methods:

   - Sending a GTL message to this instrument
   - Setting the REN message to FALSE (At this time, the local lockout state is also canceled)
   - Turning on the power again

   If this instrument receives this message when it is set to a talker, it is set to an active controller by path control. When receiving an IFC message, this instrument enters the addressable mode again.

2.3 LAN Remote Control System

The LAN (Local Area Network) interface that is compliant with IEEE standard 802.3 comes standard with this instrument so that remote control by socket communication between the external controller and this instrument can be performed.

The controlling method using the LAN remote control function is explained below.

2.3.1 Setting up LAN

1. LAN connection

   The standard LAN connection is shown below. To perform communication through LAN between an external controller and this instrument or other devices, connect them with the 10BASE-T LAN cable of the RJ45 connector. To directly connect this instrument and an external controller with a LAN cable, use a LAN cable (cross over cable) having connection as shown in Table 2-2. To connect this instrument and devices other than an external controller with a LAN, use an external device designed to connect devices having two or more LAN interfaces such as an Ethernet hub. The LAN cable used in this case is a LAN cable (straight cable) having connection as shown in Table 2-3.

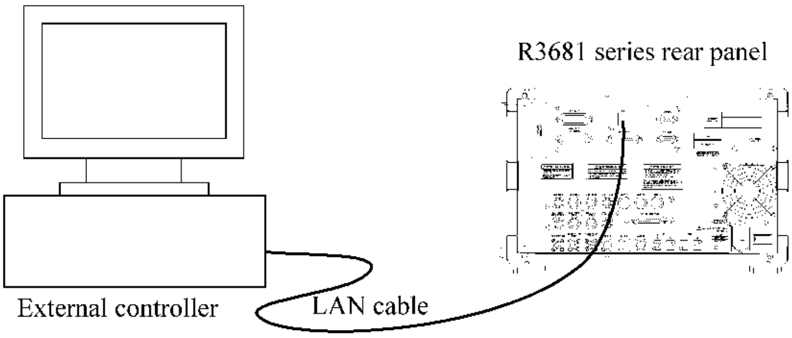


Figure 2-2 LAN Connection

Table 2-2 Connection of Cross-over Cables

Connector A side		Connector B side	
Signal name	RJ45 Pin number	RJ45 Pin number	Signal name
RX+	1	3	TX+
RX-	2	6	TX-
TX+	3	1	RX+
TX-	6	2	RX-
Not Used	4	4	Not Used
	5	5	
	6	6	
	7	7	
	8	8	

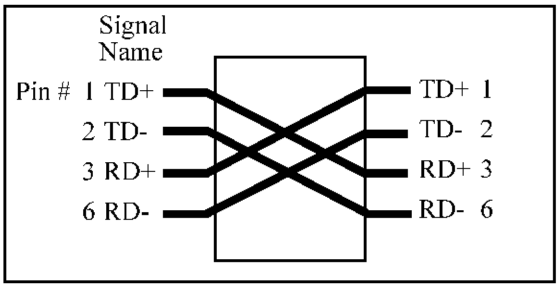


Figure 2-3 Connection of Cross-over Cables

Table 2-3 Connection of 10BASE-T Straight Cables

Signal name	RJ45 Pin number	Line color	Pair number
RX+	1	White/Orange	2
RX-	2	Orange	
TX+	3	White/Green	3
TX-	6	Green	
Not Used	4	Blue	1
	5	White/Blue	
	7	White/Brown	4
	8	Brown	

2.3.2 IP Address Setting

The IP address should be set from the network dialog box in the System menu.

2.3.3 Control from a Program

To control this instrument from a program of an external controller, a port number for socket communication is required. The port number "5025" is provided for socket communication for the remote control on the side of this instrument. To write a program for socket communication, a library for network connection with the TCP/IP protocol is required. The library differs depending on the environment, such as the OS of the external controller. In the Windows OS environment, for example, WinSock is provided.

Of the functions available in the GPIB remote control system, some functions specific to the GPIB bus, such as service request, cannot be used in the LAN remote control system.

2.4 Message Exchanging Protocol

This instrument receives program messages from the controller or other devices through the GPIB bus or LAN and generates response data. Program messages include commands, queries (which are commands that ask for response data) and data.

2.4.1 Buffers

This instrument has three buffers.

1. Input buffer

   A buffer for storing data temporarily to analyze commands.

   (1024-byte length)

   The input buffer can be cleared by the following two methods:

   - Power on
   - Executing DCL or SDC

2. Output buffer

   A buffer for storing data until data is read by the controller.

   (1024-byte length)

   The output buffer can be cleared by the following two methods:

   - Power on
   - Executing DCL or SDC

3. Error queue

   The error queue exists only in IEEE488.2-1987 command mode.

   This is a queue that stores error messages of remote commands, and its depth is 10.

   Each time an error occurs during analysis or execution of remote commands, messages are stacked in the queue.

   Messages can be read with the SYST:ERR command. Each time a message is read, it is deleted from the queue.

   The error queue can be cleared by the following two methods:

   - Power on
   - Executing *CLS

2.4.2 IEEE488.2-1987 Command Mode

The IEEE488.2-1987 command mode sends and receives messages in accordance with the message exchange protocol that is compliant with IEEE standard 488.2-1987.

When other controllers or devices receive messages from this instrument in this mode, the following items are especially important:

- Generating response data by receiving a query (Refer to “Parser”).
- Generating data in the order queries are executed (Refer to “Generating response data”).

1. Parser

   The parser receives command messages in the order they are received from the input buffer, executes syntax analysis, and determines what operations are to be executed by the commands received.

   It also traces the tree structure of commands when performing the syntax analysis of commands.

   For the next command analysis, it remembers the part of the tree structure from which analysis should be performed.

   This information is returned to the head of the tree structure when the parser is cleared.

   The parser can be cleared by the following four methods:

   - Power on
   - Receiving DCL or SDC
   - Receiving ":" next to ";"
   - Receiving the terminator or EOI

2. Generating response data

   When the parser executes a query, this instrument generates data on the output buffer as its response (that is, a query must be sent immediately before outputting data).

   It means that data is not cleared until the controller reads data generated by a query.

   Besides the controller reading data, there are two conditions in which data is cleared. In these conditions, a Query Error is generated.

   - Unterminated condition : When the controller reads response data without terminating the query (with ASCII LF code or GPIB END message) or when the controller reads response data without sending a query.
   - Interrupted condition : When the controller receives the next program message before reading response data.

2.5 Command Syntax

This chapter describes the command syntax.

2.5.1 IEEE488.2-1987 Command Mode

The command syntax is defined in the following format:

Header	→	Space (space character)	⇒	Data

MEMO: ***⇒ means repetition.***

1. Header

   The header has a layered structure consisting of two or more mnemonics delimited by colons (:). A mnemonic consisting of four or more characters has a "short form" of four (or three) characters (a non-abbreviated mnemonic is called a "long form"). Any combination of these forms is allowed.

   If a question mark (?) is attached immediately after a header, it becomes a query command.

2. Space (space character)

   A space of one or more characters is required. If any character except a space is used, an error will result.

3. Data

   When the command requires more than one data item, list these data items by delimiting them with commas (,).

   A space (space character) may be inserted before or after the comma (,).

   For more information on data type, refer to "2.5.2 Data Format."

4. Writing more than one command

   In IEEE488.2-1987 command mode, more than one command may be written on one line by delimiting them with semicolons (;).

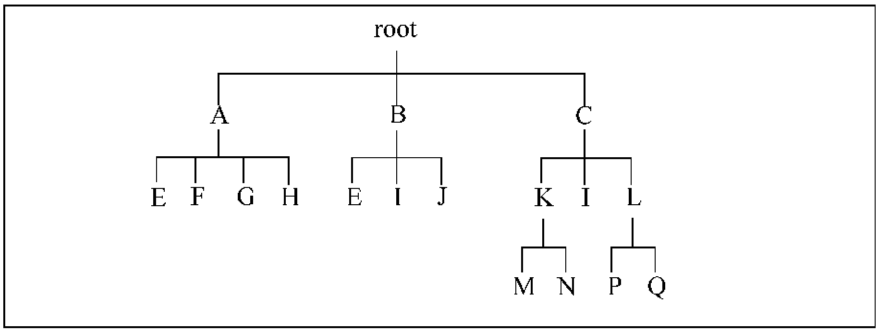
   When commands are written in this way, the system executes commands while moving the current path in the layered structure of the header.

5. Moving the current path

   The current path moves in accordance with the following rule:

   - When power is turned on: The current path is set to root.
   - Terminator: The current path is set to root.
   - Colon (:): If the colon (:) that moves the current path to one layer below in the command tree is the initial character, the colon (:) sets the current path to root.
   - Semicolon (;): The current path is not changed.
   - Common command: This command can be executed regardless of the current path. When an *RST command is executed, the current path is set to root (* See the following example).

Example: Assume the following header structure:



In the above structure, the current path moves in the following way:

1. :A:E;:B:E

   Since the colon (:) in the second command moves the current path to root, both A:E and B:E are correct commands.

2. :A:E<END>B:E

   Since <END> (terminator) moves the current path to root, both A:E and B:E are correct commands.

3. :A:E;F;G;H

   Since a semicolon (;) does not move the current path, :A:E;F;G;H results in the four commands A:E, A:F, A:G, and A:H.

4. :C:I;K:N;M

   Since a colon (:) moves the current path, K:N is seen from the layer of :C:.

   Therefore, K:N is equal to C:K:N. At the same time, the current path is changed to :C:K: because K:N contains a colon (:) and the last M is treated as C:K:M.

5. :A:E;*ESR 16

   Since the common command is independent of the current path, *ESR 16 is executed correctly.

6. :A:E;*ESR 16;F;G;H

   Since the common command does not change the current path, the third F is searched for from the current path :A: that is set in the first :A:E.

   Therefore, F, G, and H are equal to A:F, A:G, and A:H, respectively.

In the following examples, a syntax error will result.

1. :A:E;B:E

   A:E changes the current path to :A:.

   Therefore B:E is searched for from the layer of :A: but the mnemonic B cannot be found. As a result, an error will result.

2. :C:K:M;L:P

   :C:K:M changes the current path to :C:K:.

   Therefore L:P is searched for from :C:K: but the mnemonic L cannot be found. As a result, an error will result.

2.5.2 Data Format

In IEEE488.2-1987 command mode, the data types shown in this section are used in data input/output.

1. Numeric data

   There are three formats for numeric data as shown below. When entering numeric values for this instrument, any format may be used (the value is rounded depending on the data type entered).

   Depending on the command, a unit may be attached to the entered numeric value. For units, see Section (5), which is described later.

   - Integer type: NR1 format

     [Sign] → Number ⇒

   - Fixed point type: NR2 format

     [Sign] → Number ⇒ . → Number ⇒

   - Floating point type: NR3 format

     [Sign] → Number ⇒ . → Number ⇒ E or e → Sign → Number ⇒

MEMO: ***⇒ means repetition.***
The sign at the head may be omitted.

2. Character data

   The format of character data is shown below:

   English → Alphanumerics or ... ⇒

MEMO: ***⇒ means repetition.***

3. String data

   There are two formats for string data.

   " → Character/ " " ⇒ "

   ' → Character/ ' ' ⇒ '

   In string data, ASCII 7bit code characters can be used.

MEMO: ***In string data starting with ", " should be represented as " ".***
In string data starting with ', ' should be represented as ''.
⇒ means repetition.

If response data is string data, string data starting with " is always output.

4. Block data

   There are two formats for block data. When entering data into this instrument, either format may be used.

   Fixed length format: | # | → | Number | → | Number | ⇒ | DATA | ⇒

   Undefined length format: | # | → | 0 | → | DATA | ⇒ | LF^EOI |

   MEMO: ***⇒ means repetition.***

   In fixed length format, one character number following # indicates the number of digits for the subsequent bytes. 0 cannot be used (because it becomes the undefined length format).

   Example: Block data #3128<data byte>

   The number 3 following # indicates the number of digits of the subsequent string (128), and the number 128 represents the number of bytes of the subsequent <data byte>.

5. Unit

   The unit is a suffix following the numeric value. For units, a suffix can be used as a prefix.
   Available suffixes and units are listed below:

Table 2-4 Available Units

Unit	Description
Hz*	Frequency unit
DB	Level unit (relative value)
DBM	Level unit (absolute value)
S	Time unit

Table 2-5 Available Suffixes

Suffixes	
1E18	EX
1E15	PE
1E12	T
1E9	G
1E6	MA
1E3	K
1E-3	M *
1E-6	U
1E-9	N
1E-12	P
1E-15	F
1E-18	A

*: If the unit is HZ, the suffix is 1E6 (equivalent to MA).

2.6 Status Bytes

This instrument has a layered status register structure that is compliant with IEEE standard 488.2-1987, and can send various statuses of the devices to the controller. This section describes the behavioral model of status bytes and allocation of events.

1. Status registers

   This instrument adopts the model of the status registers defined in IEEE standard 488.2-1987. The status registers consist of the condition register, event register, and enable register.

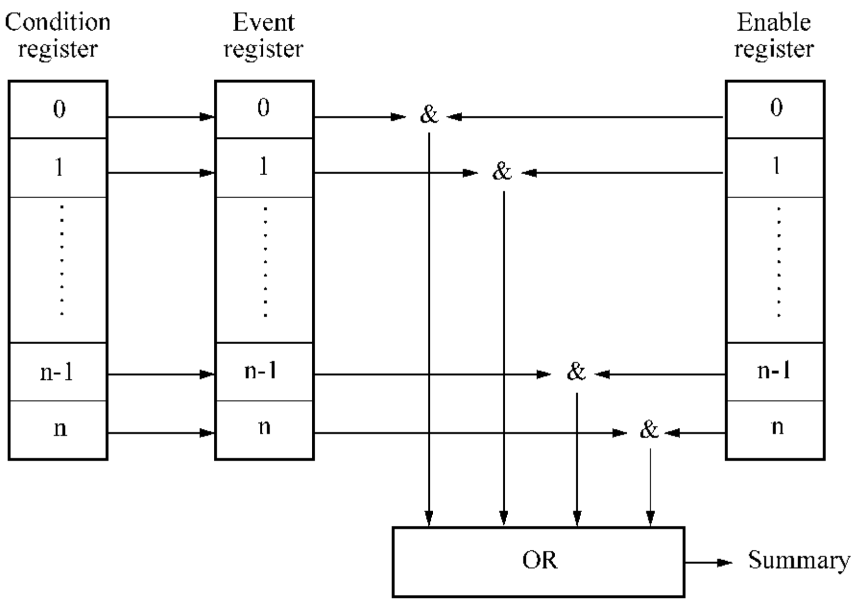


Figure 2-4 Status Register Configuration

a. Condition register

   The condition register is always monitoring the status of the device. That is, this register always retains the latest status of the device.

   However, the condition register cannot read and write data because it retains data as internal information.

b. Event register

   The event register latches and retains the status from the condition register (or retains change).

   Once this register is set, the setting value is kept until it is read by a query or cleared by *CLS.

   Data cannot be written to the event register.

c. Enable register

   The enable register specifies which bit in the event register is set as an effective status to generate a summary. The enable register is ANDed with the event register and the OR of the result is generated as a summary. The summary is written to the status byte register.

   Data can be written to the enable register.

This instrument has the following five types of status registers:

- Status byte register
- Standard event status register
- Standard operation status register
- Questionable status register
- Measuring status register

The layout of the status registers in this instrument is shown in Figure 2-5.

The detail of the status registers is shown in Figure 2-6.

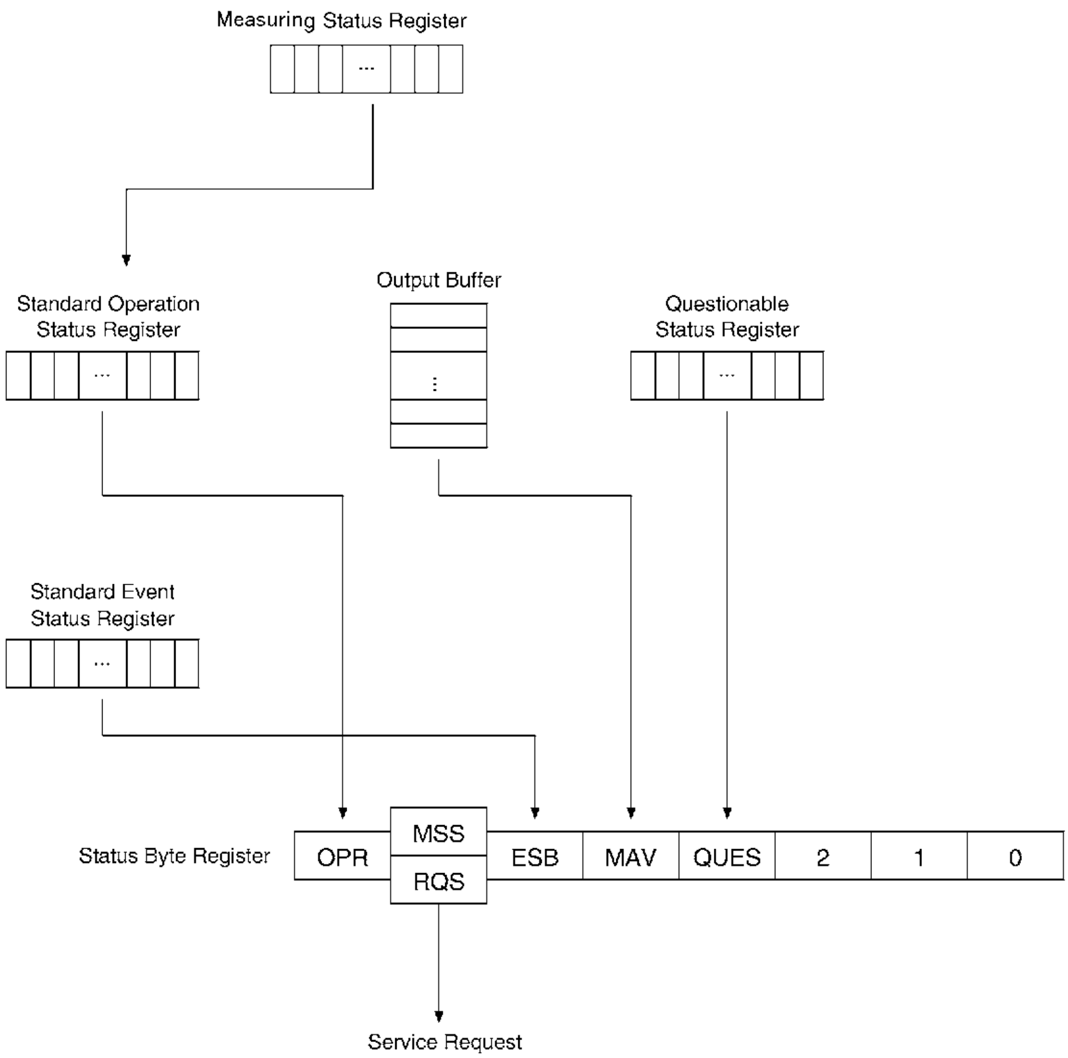


Figure 2-5 Status Register Layout

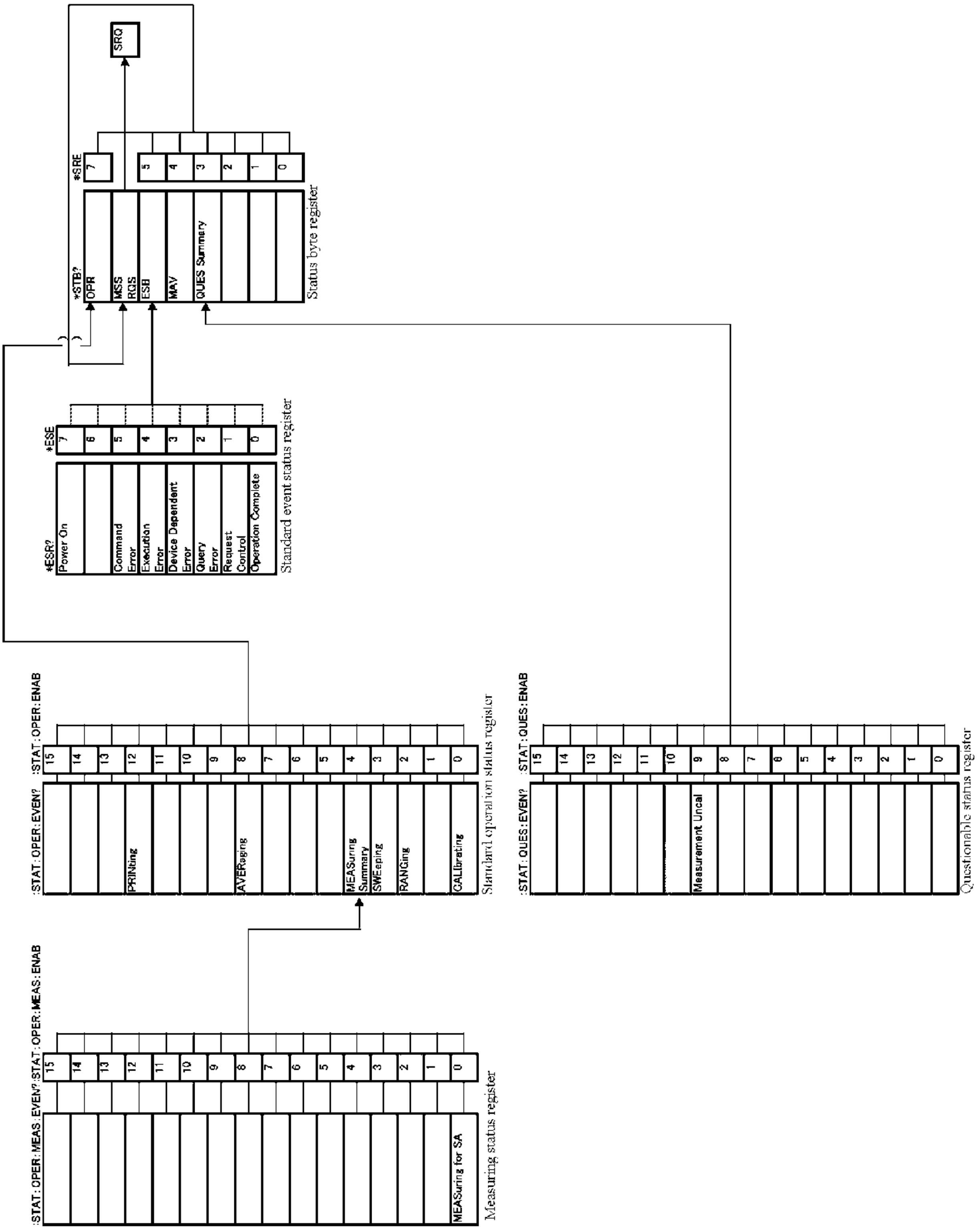


Figure 2-6 Status Register Detail

2. Event enable register
   Each event register has an enable register that determines which bit is to be enabled.

Service request enable register set	*SRE
Standard event status enable register set	*ESE
Standard operation status enable register set	:STAT:OPER:ENAB
Measuring status enable register set	:STAT:OPER:MEAS:ENAB
Questionable status enable register set	:STAT:QUES:ENAB

3. Standard operation status register
   Allocation in the standard operation status register is listed below:

Table 2-6 Allocation in the Standard Operation Status Register

bit	Function definition	Description
15	-	Always 0
14	-	Reserved
13	-	Always 0
12	-	Always 0
11 to 9	-	Always 0
8	AVERaging	Set to 1 when averaging is completed.
7 to 5	-	Always 0
4	MEASuring Summary	Set to 1 depending on the status of the measuring status register.
3	SWEeping	Set to 1 when sweep is completed.
2	RANGing	Set to 1 when Auto Level is completed.
1	-	Always 0
0	CALIbrating	Set to 1 when correction data acquisition is completed.

4. Status byte register

   The status byte register summarizes the information from the status register.

   The summary of this status byte register is sent to the controller as a service request. Therefore, the status byte register operates slightly differently than the status register structure.

   This section describes the status byte register.

   The structure of the status byte register is shown in Figure 2-7.

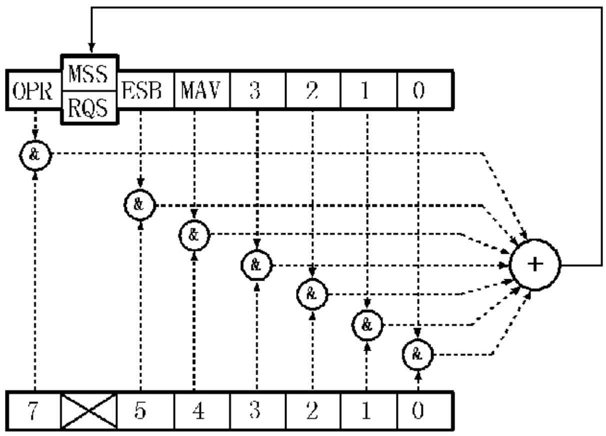


Figure 2-7 Structure of Status Byte Register

This status byte register follows the status register except for the following three points:

- The summary of the status byte register is written to bit6 of the status byte register.
- The bit6 of the enable register is always enabled and it cannot be changed.
- The bit6 (MSS) of the status byte register writes RQS of the service request.

This register responds to the serial polling from the controller. When responding to the serial polling, bit0 to bit5, bit7, and RQS of the status byte register are read, and then RQS is reset to 0. The other bits are not cleared until each factor is set to 0.

The status byte register, RQS, and MSS can be cleared by executing "*CLS". When they are cleared, the SRQ line is also set to FALSE.

The meaning of each bit in the status byte register is shown below:

Table 2-7 Meaning of Status Byte Register

bit	Function definition	Description
7	OPR	The OPR is the summary of the standard operation status register.
6	MSS	The RQS is set to TRUE when the MSS of the status byte register is set to 1, and the MSS is the summary bit of all of the status data structure. The MSS cannot be read in serial polling (but it is known that the MSS is 1 when RQS is 1). To read the MSS, use the common command *STB?. With *STB?, bit0 to bit5, bit7, and the MSS of the status byte register are read. In this case, the status byte register and the MSS are not cleared. The MSS is not set to 0 until all the unmasked factors in the status register structure are cleared.
5	ESB	The ESB is the summary of the standard event register.
4	MAV	The summary bit of the output buffer. It is set to 1 when there is output data in the output buffer and set to 0 after data is read.
3 to 0		Always 0

5. Standard event status register
   Allocation in the standard event status register is listed below:

Table 2-8 Allocation in the Standard Event Status Register

bit	Function definition	Description
7	Power on	Set to 1 when power is turned on.
6	-	Always 0
5	Command Error	Set to 1 when the parser detects a syntax error.
4	Execution Error	Set to 1 when execution of an instruction received as a GPIB command fails for some reason (e.g., the parameter is out of range).
3	Device Dependent Error	Set to 1 when an error except Command Error, Execution Error, and Query Error occurs.
2	Query Error	Set to 1 when no data exists or data is lost when the controller tries to read data from this instrument.
1	Request Control	Set to 1 when this instrument is required to be an active controller.
0	Operation Complete	Set to 1 when there is no command that is being executed for this instrument after the *OPC command is received.

6. Measuring status register
   Allocation in the measuring status register is listed below:

Table 2-9 Allocation in the Measuring Status Register

bit	Function definition	Description
15 to 13		Always 0
12		Reserved (SG)
11 to 2		Always 0
1		Reserved (Modulation)
0	MEASuring for SA	Set to 1 when measurement having a sequence in SA mode is completed.

7. Questionable status register
   Allocation in the questionable status register is listed below:

Table 2-10 Allocation in the Questionable Status Register

bit	Function definition	Description
15 to 13		Always 0
12		Reserved (SG)
11		Always 0
10		Reserved (Modulation)
9	Measurement Uncal	This bit is set to 1 if a signal level error occurs because the sweep is too fast.
8		Always 0
7		Reserved (Modulation)
6 to 0		Always 0

3. MEASURING PROCEDURE

This chapter describes the procedures for executing measurements with this instrument by remote control. The description is divided into the following sections and actual examples are shown:

3.1 Setting the Measuring Conditions

3.2 Executing a Measurement

3.3 Reading Measurement Data

3.1 Setting the Measuring Conditions

This section describes the setting of the measuring conditions.

3.1.1 Selecting the Measurement Mode

This instrument has two modes: the mode of analyzing the spectrum and the mode of analyzing the signal in the Base Band. Therefore, you must select the mode to be used in accordance with the purpose of measurement.

In this example, the spectrum analysis mode is set.

- Setting the spectrum analysis mode:
  :SYST:SEL SAN

3.1.2 Setting the Frequency

Set the center frequency, span frequency, and resolution bandwidth, etc. depending on the frequency of the signal to be measured in spectrum analysis mode. The following commands are used for the above settings:

- Setting the center frequency:
  :SENS:FREQ:CENT
- Setting the span frequency:
  :SENS:FREQ:SPAN
- Setting the resolution bandwidth (RBW):
  :SENS:BAND:RES
- Setting the video bandwidth (VBW):
  :SENS:BAND:VID

3.1.3 Setting the Level

Set the reference level and attenuator of this instrument, depending on the output level of the signal to be measured. The following commands are used for the above settings:

- Setting the reference level:
  :DISP:TRAC:Y:RLEV
- Setting the attenuator:
  :INP:ATT

3.1.4 Setting the Sweep Time

Set the sweep time during measurement. The following command is used for the above setting:

- Setting the sweep time:
  :SENS:SWE:TIME

3.2 Executing a Measurement

This section describes the execution of measurement.

After the basic setting in spectrum analysis mode is completed, set the parameters associated with each measurement. After that, execute a measurement.

Measurement is usually performed by the following procedure:

- Ordinary measurement procedure:

Initializing the measurement status (ABORt)

Moving to the measurement mode (CONFigure)

Starting a measurement (INITiate)

Waiting for the completion of measurement (*OPC, *WAI, etc.)

Reading the measured value (FETCh)

By using the convenient command "MEASure," all of the above steps are executed automatically and the measurement result is returned, although a command is provided for each step.

3.2.1 Setting the Measurement Parameters, Selecting the Measuring Items, and Starting a Measurement

The explanation in this section assumes that the Power will be measured.
First, set the target frequency width of the Power measurement.

- Initializing the measurement status:
  :ABORt
- Setting the target frequency width of the power measurement:
  :SENS:CPOW:WIND:WIDT

Next, set the number of times of averaging the Power measurement operation.

- Setting the number of times of averaging operation in measurement:
  :SENS:CPOW:AVER:COUN

Finally, move to the measurement mode with the next commands and start measurement.

- Command for moving to the measurement mode:
  :CONF:CPOW
- Command for starting measurement:
  :INIT:IMM

3.2.2 Waiting for the Completion of Measurement

To wait for the completion of measurement, the following methods can be used:

- Polling the status register
- Using a Service Request (SRQ)
- Using the common commands *WAI, *OPC, and *OPC?
- Using READ of the SCPI command or the MEASure command

3.2.3 Polling the Status Register

In the method of using the polling of the status register, the external controller checks for the change of the status of this instrument by using a command of an appropriate status register.

This method is useful in the following cases:

- When the programming environment of the external controller to be used does not support the SRQ interrupt mechanism
- When a remote control through a LAN is used
- When you do not want to make a complicated setting for SRQ processing because you will write a program for a simple measurement

3.2.4 Using a Service Request (SRQ)

In the method of using an SRQ, an SRQ signal is sent from this instrument to the external controller in accordance with the detection condition preset by the external controller, and the external controller checks the state of this instrument in accordance with the SRQ signal.

This method is useful in the following cases:

- When the measurement time is restricted as a system
- When multiple measuring instruments must be monitored in addition to this instrument
- When the external controller must execute other processing during measurement wait time

3.2.5 Using the Common Commands

Of the common commands, the following commands can be used for measurement synchronization:

- *OPC

  When measurement is completed, the "Operation Complete" bit in the standard event status register is set.
- OPC?

  As an ordinary query response instead of the bit information of the standard event status register, the number "1" is returned when measurement is completed.
- WAI

  All the commands that were sent before sending the *WAI command are executed and the commands after the *WAI command are made to wait.

By using the characteristics of these commands, the completion of this instrument is detected from the external controller.

For more information on each common command, refer to "Command Reference."

3.2.6 Using the READ/MEASure Command

Besides the above methods, the measurement completion of this instrument can be detected by this instrument's response to a query for the READ/MEASure command sent by the external controller.

In this method, without accessing the status register, the point at which the response of the measurement result is returned from this instrument can be regarded as the completion of measurement.

- When the external controller to be used does not support the SRQ interrupt mechanism
- When a remote control through a LAN is used
- When there is relatively little restriction for the measurement time in the entire system and you want to execute a measurement and read the measurement result easily

3.3 Reading Measurement Data

This section describes the method of reading measurement data.

3.3.1 Types of Commands for Reading Measurement Data

After measurement in spectrum analysis mode is completed and the completion of measurement is detected, the external controller reads the measurement result data.
To read data from the external controller, a query command for reading the measurement result data is provided for each measurement function.

This instrument provides the following three commands for reading the measurement result:

- FETCh command
- READ command
- MEASure command

Since these three types of commands have the following features, an appropriate command can be used according to the purpose of use.

- FETCh command

  This command only reads the measurement result of the target.
- READ command

  The ABORt command and the measurement mode are not moved and a measurement is started by the INITiate command. After measurement is completed, internal operation by the above FETCh command is executed and the measurement result is read.
- MEASure command

  After being moved to the measurement mode by the ABORt and CONFigure commands, a measurement is started by the INITiate command. After measurement is completed, internal operation by the above FETCh command is executed and the measurement result is read.

3.3.2 Reading Measurement Data

This section shows the FETCh command for reading the result data of Power measurement as an example.

- Query command for reading the Channel Power measured value:

  :FETCh:CPOW?

In response to this reading query command, this instrument outputs the measurement result data in question to the output buffer. In this example, the Power value as the measurement result is set to the output buffer. The external controller can read the data in the output buffer through the GPIB or LAN interface by using a program.

4. PROGRAM EXAMPLES FOR REMOTE CONTROL

This chapter describes program examples for remote control.

In the program examples in this chapter, Microsoft's Visual Basic is used. When writing a program in another language, change the description to the language used.

In the explanation of these programs, a GPIB board manufactured by National Instruments (NI) is assumed as the GPIB bus controller.

4.1 Basic Steps for GPIB Bus Control

This section describes step-by-step the operations required to control the GPIB bus from Visual Basic. For the initialization of variables which depend on Visual Basic and the definition of function routines, follow the notational conventions of Visual Basic programs.

4.1.1 Reading the GPIB Control Library for Visual Basic

To control an NI's GPIB board from a program written in Visual Basic, you must integrate two files - a VBIB-32.BAS file containing a GPIB communications interface for Visual Basic provided by NI, and a NIGLOBAL.BAS file that defines errors and timeout values - into the Project of Visual Basic.

4.1.1.1 Initializing the Controller

To communicate with this instrument through GPIB, you must first of all initialize the GPIB controller. An example of GPIB initialization is shown below:

```
Rem ----------------- Initialize GPIB Controller ----------------------------
Public Sub InitGPIB()

saaddress% = 8

Call ibfind ("GPIB0", boardID%)              ' Open GPIB board
Call ibfind ("DEV1", analyzer%)              ' Open SA analyzer port
Call ibpad( analyzer%, saaddress%)           ' Set the SA's GPIB address

Call ibtmo( analyzer%, 12)                   ' Set timeout value to 3 sec

End Sub
```

4.1.1.2 Initializing this Instrument

The following program initializes this instrument before controlling the GPIB:

```
Rem ----------------- Initialize Spectrum Analyzer ----------------------------
Public Sub InitSA( )

Call ibwrt ( analyzer%, "*CLS" )                   ' Reset status register
Call ibwrt ( analyzer%, "*RST" )                   ' Reset this instrument

End Sub
```

4.1.1.3 Simple Setting Commands

The following program makes simple settings of this instrument:

```
Rem ----------------- Brief setting of Spectrum Analyzer ----------------------------
Public SUB SASetting( )

Call ibwrt( analyzer%, ":FREQ:CENT 1.9984GHZ" )     ' Set Center Freq. to 1.9984MHz
Call ibwrt( analyzer%, ":FREQ:SPAN 10MHZ" )         ' Set Freq. Span to 10MHz
Call ibwrt( analyzer%, ":DISP:TRAC:Y:RLEV  10DBM" ) ' Set Reference level 10dBm

End Sub
```

4.1.1.4 Reading the Setting Values

The following program reads the setting values of this instrument:

```
Rem ----------------- Read the setting value of Spectrum Analyzer ---------------
Public Sub ReadSASetting( )

CF$= Space$(32)                                    ' Prepare the text variable for read
Call ibwrt( analyzer%, ":FREQ:CENT?" )             ' Read request of center freq.
Call ibrd( analyzer%, CF$ )                        ' Read setting value
SP$= Space$(32)                                    ' Prepare the text variable for read
Call ibwrt( analyzer%, ":FREQ:SPAN?" )             ' Read request of span freq.
Call ibrd( analyzer%, SP$ )                        ' Read setting value

Rem -------------------- Display setting value ---------------------------------------
Call MsgBox ( "Center freq.: " & CF$ )
Call MsgBox ( "Span freq.: " & SP$ )

End Sub
```

4.1.1.5 Setting the Marker and Reading the Marker Value

The following program searches for the maximum level of the signal by using the marker and reads the level of the signal with the marker.

```
Rem ----------------- Read signal level using the marker function-------------------
Public Sub ReadMkrSignal( )

MKFreq$= Space$(32)                                  ' Prepare the text variable for read
MKLevel$= Space$(32)                                 ' Prepare the text variable for read
Call ibwrt( analyzer%, ":CALC:MARK:FUNC ON" )        ' Turn on the marker
Call ibwrt( analyzer%, ":CALC:MARK:MAX" )            ' Search peak point of signal
Call ibwrt( analyzer%, ":CALC:MARK:X?" )             ' Query the marker frequency
Call ibrd( analyzer%, MKFreq$ )                      ' Read it

Call ibwrt( analyzer%, ":CALC:MARK:Y?" )             ' Query the marker level
Call ibrd( analyzer%, MKLevel$ )                     ' Read it

Rem -------------------- Display the freq. and level of signal------------------------
Call MsgBox ( "Marker freq.: " & MKFreq$ & " Level: " & MKLevel$ )

End Sub
```

4.1.1.6 Executing a Measurement and Reading the Measurement Result

This section shows some measurement execution examples and presents some examples of how to synchronize with this instrument for the execution of a measurement and reading of the measurement result after execution.

- Using the common commands for synchronization

  Of common commands, there are commands defined for synchronization during command execution (*WAI, *OPC?, *OPC). Examples using these commands are shown below:

Example 1: This example performs peak search of the marker after sweep, reads the result data, and displays it (using the *WAI command).

```
Rem -----------------Do search the peak point and get level data after sweeping---------
Public Sub GetPeakPoint1( )

Call ibwrt( analyzer%, ":INIT:CONT OFF" )            ' Set sweep mode to single sweep
Call ibwrt( analyzer%, ":INIT:ABOR" )                ' Stop sweeping
Call ibwrt( analyzer%, ":INIT:IMM" )                 ' Start sweeping
Call ibwrt( analyzer%, "*WAI" )                      ' Wait for end of sweep

MKLevel$= Space$(32)
Call ibwrt( analyzer%, ":CALC:MARK:FUNC ON" )        ' Turn on the marker
Call ibwrt( analyzer%, ":CALC:MARK:MAX" )            ' Search peak point of signal
Call ibwrt( analyzer%, ":CALC:MARK:Y?" )             ' Query the marker level
Call ibrd( analyzer%, MKLevel$ )                     ' Read it
Rem -------------------- Display setting value-----------------------------------------
Call MsgBox ( "Get Peak level after sweeping := " & MKLevel$ & "dBm" )

End Sub
```

Example 2: This example performs peak search of the marker after sweep, reads the result data, and displays it (using the *OPC? command).

```
Rem -----------------Do search the peak point and get level data after sweeping---------
Public Sub GetPeakPoint2( )

Call ibwrt( analyzer%, ":INIT:CONT OFF" )          ' Set sweep mode to single sweep
Call ibwrt( analyzer%, ":INIT:ABOR" )              ' Stop sweeping
Call ibwrt( analyzer%, "*CLS" )                    ' Clear status
Call ibwrt( analyzer%, ":INIT:IMM" )               ' Start sweeping

OPEND$ = Space$(3)
Do
  Call ibwrt( analyzer%, "*OPC?")                  ' Request Operation complete status
                                                   ' as sweep end info.
  Call ibrd( analyzer%, OPEND$)                    ' Read status
Loop until ( Int(Val(OPEND$) ) ) And 1 = 1

MKLevel$= Space$(32)
Call ibwrt( analyzer%, ":CALC:MARK:FUNC ON")       ' Turn on the marker
Call ibwrt( analyzer%, ":CALC:MARK:MAX" )          ' Search peak point of signal
Call ibwrt( analyzer%, ":CALC:MARK:Y?" )           ' Query the marker level
Call ibrd( analyzer%, MKLevel$ )                   ' Read it
Rem -------------------- Display setting value ----------------------------------------
Call MsgBox ( "Get Peak level after sweeping := " & MKLevel$ & "dBm" )

End Sub
```

Example 3: This example performs peak search of the marker after sweep, reads the result data, and displays it (using the *OPC command and taking the timing with SRQ).

```
Rem -----------------Do search the peak point and get level data after sweeping---------
Public Sub GetPeakPoint3( )

Call ibwrt( analyzer%, "*SRE 32" )                 ' Set SRQ for ESR to enable
Call ibwrt( analyzer%, "*ESE 1" )                  ' Set enable bit for OPC

Call ibwrt( analyzer%, ":INIT:CONT OFF" )          ' Set sweep mode to single sweep
Call ibwrt( analyzer%, ":INIT:ABOR")               ' Stop sweeping
Call ibwrt( analyzer%, "*CLS" )                    ' Clear status
Call ibwrt( analyzer%, ":INIT:IMM" )               ' Start sweeping
Call ibwrt( analyzer%, "*OPC" )                    ' Send OPC for synchronization

Call WaitSRQ( boardID%, res% )                     ' Wait for SRQ using driver's func.
Call ibrsp( analyzer%, stb% )                      ' Execute serial poll

MKLevel$= Space$(32)
Call ibwrt( analyzer%, ":CALC:MARK:FUNC ON")       ' Turn on the marker
Call ibwrt( analyzer%, ":CALC:MARK:MAX" )          ' Search peak point of signal
Call ibwrt( analyzer%, ":CALC:MARK:Y?" )           ' Query the marker level
Call ibrd( analyzer%, MKLevel$ )                   ' Read it
Rem -------------------- Display setting value ----------------------------------------
Call MsgBox ( "Get Peak level after sweeping := " & MKLevel$ & "dBm" )

End Sub
```

- Using the Measure command

  The Measure command contains the functions of command execution, synchronization, and reading so that the synchronization of measurement can be performed without detailed control. Since the time from the execution of measurement to synchronization and reading is treated as the timeout time on the GPIB driver in this case, it may be necessary to extend the timeout value of the GPIB bus. Note that the Measure command is not defined in all measurements.

After the parameters of a Carrier Power measurement are set, a measurement is executed and the result is read.

```
Rem -----------------Do search the peak point and get level data after sweeping---------
Public Sub GetPeakPoint1( )

Call ibtmo( analyzer%, 13)                              ' Set timeout value to 10sec

Call ibwrt( analyzer%, ":INIT:CONT OFF" )               ' Set sweep mode to single sweep
Call ibwrt( analyzer%, ":INIT:ABOR")                    ' Stop sweeping

ResCarPow$ = Space$(32)
Call ibwrt( analyzer%, ":MEAS:CPOW?" )                  ' Start carrier power measurement
Call ibrd ( analyzer%, ResCarPow$)                      ' Wait for receiving of meas. result

MKLevel$= Space$(32)
Call ibwrt( analyzer%, ":CALC:MARK:FUNC ON")            ' Turn on the marker
Call ibwrt( analyzer%, ":CALC:MARK:MAX" )               ' Search peak point of signal
Call ibwrt( analyzer%, ":CALC:MARK:Y?" )                ' Query the marker level
Call ibrd( analyzer%, MKLevel$ )                        ' Read it
Rem -------------------- Display setting value ----------------------------------------
Call MsgBox ( "Get Peak level after sweeping := " & MKLevel$ & "dBm" )

End Sub
```

4.1.1.7 Setting and Reading the Status Registers

For access to the status registers of this instrument, there are two types of commands.

One type consists of the commands defined in IEEE488.2 and the other consists of the commands for the registers extended by the SCPI.

IEEE488.2 Register Commands

Command	Function
*SRE	Sets the enable bit of the status byte register
*STB?	Reads the status byte register
*ESE	Sets the enable bit of the standard event register
*ESR?	Reads the standard event register

SCPI Extended Register Commands

Command	Function
:STATus:OPERation:ENABle	Sets the enable bit of the standard operation status register
:STATus:OPERation:EVENt?	Reads the standard operation status register
:STATus:QUEStionable:ENABle	Sets the enable bit of the questionable status register
:STATus:QUEStionable:EVENt?	Reads the questionable status register
:STATus:OPERation:MEASure:ENABle	Sets the enable bit of the measuring status register
:STATus:OPERation:MEASure:EVENt?	Reads the measuring status register

The following program clears the status registers and prepares for generating a service request, depending on the change of the status byte register.

```
Rem -----------------Prepare status registers condition for getting SQR signal----------
Public Sub PrepStatusReg( )

Call ibwrt( analyzer%, "*CLS" )                       ' Clear status registers
Call ibwrt( analyzer%, "*SRE 160")                    ' Enable service req. for ESB and
                                                      ' OPR bit
Call ibwrt( analyzer%, "*ESE 1")                      ' Set event enable for Operation
                                                      ' Complete of the ESR
Call ibwrt( analyzer%, ":STAT:OPER:ENAB 272" )        ' Set event enable for averaging end
                                                      ' and measurement end
Call ibwrt ( analyzer%, ":STAT:OPER:MEAS:ENAB 1")     ' Enable SA measurement
                                                      ' end event

End Sub
```

The following program checks the cause of the generation of a service request after it is generated.

```
Rem -----------------Read GPIB status register ----------------------
Public Sub ReadStatusReg( )

Stb$ = Space$(5)
Call ibwrt( analyzer%, "*STB?" )                    ' Read standard event reg.
Call ibrd( analyzer%, Stb$)

NumStb% = Int(Val(Stb$))
If (NumStb% And 32) > 0  then Call StanEventProcess ' Call standard event process
If ( NumStb% And 128) > 0 then Call OprEventProcess ' Call operation event process

End Sub

Rem ------------------ Check standard event bit ----------------------------------
Public Sub StanEventProcess( )

Ste$ = Space$(5)
Call ibwrt ( analyzer%, "*ESR?" )                   ' Read Standard event reg.
Call ibrd ( analyzer%, Ste$ )

NumSte% = Int(Val(Ste$))
If (NumSte% And 1) > 0  then Call MsgBox( "Operation complete" )

End Sub

Rem ------------------ Check standard event bit ----------------------------------
Public Sub OprEventProcess( )

Ope$ = Space$(7)
Call ibwrt ( analyzer%, ":STAT:OPER:EVEN?")         ' Read operation event reg.
Call ibrd ( analyzer%, OPE$)

NumOpe% = Int(Val(Ope$))
If (NumOpe% And 256) > 0 then Call MsgBox( "Averaging done" )
If (NumOpe% And 16) > 0 then Call MsgBox( "Some measurement has done" )

End Sub
```

4.1.1.8 Frequency Measurement Using the Frequency Counter

This section shows an example of making a high precision measurement of the signal frequency by using the marker counter function.

Measuring the frequency by using the marker counter function

```
Rem -----------------Read signal frequency using marker counter function -------------
Public Sub ReadPrecisionFreq( )

CounterFreq$ = Space(100)
Call ibwrt( analyzer%, ":INIT:CONT OFF" )          ' Set to single sweep mode
Call ibwrt( analyzer%, ":INIT:ABOR")               ' Stop sweeping
Call ibwrt( analyzer%, ":INIT:IMM" )               ' Start sweeping
Call ibwrt( analyzer%, "*WAI" )                    ' Wait for sweep end
Call ibwrt( analyzer%, ":CALC:MARK:FUNC ON")       ' Turn on the marker
Call ibwrt( analyzer%, ":CALC:MARK:MAX" )          ' Search peak point of signal

Call ibwrt( analyzer%, ":FCO:AVER:COUN 2" )        ' Set counter average times
Call ibwrt( analyzer%, ":FCO:AVER ON" )            ' Set counter average func. to ON
Call ibwrt( analyzer%, ":CALC:MARK:FCO ON" )       ' Freq. counter func. on
Call ibwrt( analyzer%, ":INIT:TS" )                ' Start sweeping and
                                                   ' wait for sweep end and count end

Call ibwrt( analyzer%, ":CALC:MARK:FCO:FREQ?" )    ' Read out counter freq.
Call ibrd( analyzer%, CounterFreq$ )
Call MsgBox( "Marker counter freq. = " & CounterFreq$ )

End Sub
```

4.1.1.9 Channel Power Measurement

This section shows an example of measuring the power of the signal by using the Channel Power measurement function, which is a type of the power measurement functions.

Executing a Channel Power measurement and reading the result

```
Rem ---------------Measure channel power ---------------------------------------------
Public Sub MeasChanPower( )

ChannelPow$ = Space(100)

Call ibwrt( analyzer%, ":FREQ:CENT 800MHZ" )        ' Set carrier freq.
Call ibwrt( analyzer%, ":FREQ:SPAN 5MHZ" )

Rem ----------------- Prepare for channel power measurement ---------------------------
Call ibwrt( analyzer%, ":CPOW:DATA:MODE MAN" )      ' Set parameter mode to manual
Call ibwrt( analyzer%, ":CPOW:AVER:COUN 10" )       ' Set average times
Call ibwrt( analyzer%, ":CPOW:AVER ON")             ' Set average func. to ON
Call ibwrt( analyzer%, ":CPOW:AVER:MODE REP" )      ' Set meas mode to REPEAT
Call ibwrt( analyzer%, ":CPOW:WIND:POS  800MHZ" )   ' Set channel power window
Call ibwrt( analyzer%, ":CPOW:WIND:WIDT 1.288MHZ")

Call ibwrt( analyzer%, ":MEAS:CPOW?" )              ' Start measurement
Call ibrd( analyzer%,  ChannelPow$ )                ' Read out power

Call MsgBox( "Channel power  = " & ChannelPow$ & "dBm" )

Call ibwrt( analyzer%, ":CONF:NORM" )               ' Quit measurement
End Sub
```

4.1.1.10 ACP Measurement

This section shows an example of measuring the adjacent channel leakage power (ACP) of the signal by using the ACP measurement function, which is a type of the Power measurement functions.

Executing an ACP measurement and reading the result

```
Rem ----------------- Measure Adjacent Channel Power --------------------------
Public Sub MeasACP( )

ResultACP$ = Space(200)

Call ibwrt( analyzer%, ":FREQ:CENT 2GHZ" )          ' Set carrier freq.
Call ibwrt( analyzer%, ":FREQ:SPAN 25MHZ" )

Rem ----------------- Setting of Adjacent channel parameters -------------------
Call ibwrt( analyzer%, ":ACP:CSBW:DATA:DEL" )       ' Clear Channel Space param.
Call ibwrt( analyzer%, ":ACP:CBW 3.84MHZ" )         ' Set Channel Bandwidth
Call ibwrt( analyzer%, ":ACP:CSBW:DATA 5MHZ,3.84MHZ")' Set Adj. Channel param.
Call ibwrt( analyzer%, ":ACP:CSBW:DATA 10MHz,3.84MHZ")' Set Adj. Channel param.

Rem ----------------- Setting of Root Nyquist filter's parameters -------------
Call ibwrt( analyzer%, ":ACP:RNYQ:SRAT 3.84MHZ" )   ' Set Symbol rate of filter
Call ibwrt( analyzer%, ":ACP:RNYQ:RFAC 0.22" )      ' Set Roll off factor of filter
Call ibwrt( analyzer%, ":ACP:RNYQ ON" )             ' Set Nyq. Filter operation to on

Call ibwrt( analyzer%, ":ACP:AVER:COUN 10" )        ' Set average times
Call ibwrt( analyzer%, ":ACP:AVER ON" )             ' Set average func. to ON

Call ibwrt( analyzer%, ":MEAS:ACP?" )               ' Start measurement
Call ibrd( analyzer%,  ResultACP$ )                 ' Read out all meas. results of ACP

Call MsgBox( "ACP results : " & ResultACP$ )
Call ibwrt( analyzer%, ":CONF:NORM" )               ' Quit measurement
End Sub
```

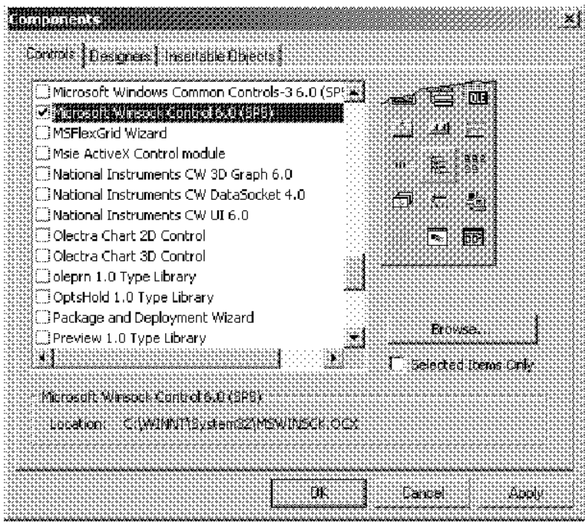
4.2 Basic Steps for LAN Control

This section describes step-by-step the operations required to control the LAN interface from Visual Basic. For the initialization of which depend on Visual Basic variables and the definition of function routines, follow the notational conventions of Visual Basic programs.

4.2.1 Reading the LAN Control Library for Visual Basic

To control a LAN interface board from a program written in Visual Basic, you must integrate the Winsock control provided by Microsoft into the Project of Visual Basic.

To use the Winsock control, make a setting so that WINSCK.OCX can be used in the setting of Visual Basic components. The following figure shows an example of adding WINSCK.OCX in Visual Basic 6.0. (Procedure: Select the Component (O) submenu from the Project (P) menu, check [Microsoft Winsock Control 6.0] on the list of the displayed component dialog, and click the OK button.)



After this setting is made, a Winsock control object is displayed in the tool box.

When you draw this Winsock control in the form, the LAN can be controlled by using the object drawn. After you draw the Winsock control and create an object, specify a unique object name. In a later part of this manual, explanation is given with the object name tcpClient.

4.2.1.1 Opening the Socket Interface (Initialization)

To communicate with this instrument through a LAN, you must first of all make a connection to the port of this instrument. To make a connection, you must specify the IP Address (or the host name) and the port number of this instrument in the properties of RemoteHost and RemotePort, respectively. In addition, specify the protocol (Protocol property) to be used in TCP (sckTCPProtocol). After that, a connection to this instrument is made by using the Connect method of the Winsock control. For the communication port number to communicate with this instrument, the number "5025" is used. Specify this number for the port number of the connection destination.

```
Rem ---------------- Connection LAN Interface ----------------------------
Public Sub ConnectTCP( )

tcpClient.
tcpClient.RemoteHost = "192.0.0.1"              ' Set IP Address of SA
tcpClient.Protocol = sckTCPProtocol             ' Set protocol to TCP
tcpClient.RemotePort = 5025                     ' Set port no. 5025 of SA


tcpClient.Connect                               ' Connect to SA's port

End Sub
```

MEMO:

- ***To connect to this instrument, the above operation must be executed once before performing LAN control. Once it is executed, the connection is maintained until connection close processing is performed (in the above case, by executing the tcpClient.Close method).***
- ***Since close processing is usually performed at the end of the program, explicit close processing is not necessarily required.***
- ***Since frequent repetitions of Connect and Close may damage the Winsock control, design the program so that control is made in one Connect processing whenever possible.***

4.2.1.2 Initializing this Instrument

The following program initializes this instrument before controlling the LAN:

```
Rem ----------------- Initialize Spectrum Analyzer ----------------------------
Public Sub InitSA( )

tcpClient.SendData "*CLS" " & vbCrLf                ' Reset status register
tcpClient.SendData "*RST" " & vbCrLf                ' Reset this instrument

End Sub
```

4.2.1.3 Simple Setting Commands

The following program makes simple settings of this instrument:

```
Rem ----------------- Brief setting of Spectrum Analyzer ----------------------------
Rem ----------------- Set Center freq. to 1.9984GHz, Span to 10MHz -------------
Rem ------------------ and Reference level to 10dBm              --------------
Public Sub SASetting( )

tcpClient.SendData ":FREQ:CENT 1.9984GHZ" & vbCrLf
tcpClient.SendData ":FREQ:SPAN 10MHZ" & vbCrLf
tcpClient.SendData ":DISP:TRAC:Y:RLEV  10DBM" & vbCrLf

End Sub
```

4.2.1.4 Reading the Setting Values

The following program reads the setting values of this instrument:

```
Rem ----------------- Read the setting value of Spectrum Analyzer ---------------
Public Sub ReadSASetting( )

CF$= Space$(32)                                   ' Prepare the text variable for read
tcpClient.SendData ":FREQ:CENT?" & vbCrLf         ' Read request of center freq.

Do While (tcpClient.BytesReceived = 0)            ' Wait for receiving a character
  DoEvents
Loop

tcpClient.GetData CF$                             ' Read setting value
SP$= Space$(32)                                   ' Prepare the text variable for read
tcpClient.SendData ":FREQ:SPAN?" & vbCrLf         ' Read request of span freq.

Do While (tcpClient.BytesReceived = 0)            ' Wait for receiving a character
  DoEvents
Loop

tcpClient.GetData SP$                             ' Read setting value

Rem -------------------- Display setting value ----------------------------------------
                                                -----
Call MsgBox( "Center freq.: " & CF$ & "Span freq.: " & SP$)

End Sub
```

4.2.1.5 Setting the Marker and Reading the Marker Value

The following program searches for the maximum level of the signal by using the marker and reads the level of the signal with the marker.

```
Rem ----------------- Read signal level using the marker function-------------------
Public Sub ReadMkrSignal( )

MKLevel$= Space$(32)                                ' Prepare the text variable for read

tcpClient.SendData ":CALC:MARK:FUNC ON"& vbCrLf     ' Turn on the marker
tcpClient.SendData ":CALC:MARK:MAX" & vbCrLf        ' Search peak point of signal
tcpClient.SendData ":CALC:MARK:X?" & vbCrLf         ' Query the marker frequency

Do While (tcpClient.BytesReceived = 0)              ' Wait for receiving a character
  DoEvents
Loop

tcpClient.GetData MKFreq$                           ' Read it

tcpClient.SendData ":CALC:MARK:Y?" & vbCrLf         ' Query the marker level

Do While (tcpClient.BytesReceived = 0)              ' Wait for receiving a character
  DoEvents
Loop

tcpClient.GetData MKlevel$                          ' Read it

Rem ------------------- Display the freq. and level of signal------------------------
Call MsgBox("Marker freq.: " & MKFreq$ & " Level: " & MKLevel$)

End Sub
```

4.2.1.6 ACP Measurement

This section shows an example of measuring the adjacent channel leakage power (ACP) of the signal by using the ACP measurement function, which is a type of the Power measurement functions.

```
Rem -----------------Measure Adjacent Channel Power -----------------------------------
Public Sub MeasACP( )

ResultACP$ = Space(200)

tcpClient.SendData ":FREQ:CENT 2GHZ" & vbCrLf       ' Set carrier freq.
tcpClient.SendData ":FREQ:SPAN 25MHZ" & vbCrLf

Rem ----------------- Setting of Adjacent channel parameters --------------
tcpClient.SendData ":ACP:CSBW:DATA:DEL" & vbCrLf    ' Clear Channel Space param.
tcpClient.SendData ":ACP:CBW 3.84MHz"& vbCrLf       ' Set Channel Bandwidth
tcpClient.SendData ":ACP:CSBW:DATA 5MHz,3.84MHz" & vbCrLf  'Adj. Channel param.
tcpClient.SendData ":ACP:CSBW:DATA 10MHz,3.84MHz" & vbCrLf ' Adj. Channel param.

Rem ----------------- Setting of Root Nyquist filter's parameters --------------
tcpClient.SendData ":ACP:RNYQ:SRAT 3.84MHz" & vbCrLf ' Set Symbol rate of filter
tcpClient.SendData ":ACP:RNYQ:RFAC 0.22" & vbCrLf   ' Set Roll off factor of filter
tcpClient.SendData ":ACP:RNYQ ON" & vbCrLf          ' Set Nyq. Filter operation to on

tcpClient.SendData ":ACP:AVER:COUN 10" & vbCrLf     ' Set average times
tcpClient.SendData ":ACP:AVER ON" & vbCrLf          ' Set average func. to ON

tcpClient.SendData ":MEAS:ACP?" & vbCrLf            ' Start measurement

Do While (tcpClient.BytesReceived = 0)              ' Wait for receiving a character
  DoEvents
Loop

tcpClient.GetData ResultACP$                        ' Read out all meas. results of ACP

Call MsgBox("ACP results : " & ResultACP$)

End Sub
```

5. SCPI COMMAND REFERENCE

This chapter describes the SCPI command reference for this instrument.

5.1 Command Reference Format

This section describes the format of explanations of each command described in this chapter.

Explanations of each command include the following items:

- Command
- Command syntax
- Function description
- Parameter
- Query reply
- Example
- Relevant command

- [Command syntax]

  The command syntax shows the syntax of a command sent from the external controller to this instrument. The syntax consists of a command part and a parameter part. The command part and parameter part are delimited by a space.

  When there are multiple parameters, they are delimited by commas (,). The three points (...) displayed between commas represent the parameter(s) omitted in the position.

  For example, the description <numeric value 1>,..., <numeric value 4> shows that four parameters,<numeric value 1>, <numeric value 2>, <numeric value 3>, and <numeric value 4>, are required.

  When the parameter is a character string type such as <character string>,<character string 1>, the parameter must be enclosed in double quotation marks (" "). When the parameter is <block>, it shows the block format data.

  The part written in lowercase alphabetical characters in the syntax shows that it can be omitted.

  For example, ":CALibration:CABLe" can be abbreviated to ":CAL:CABL".

  The marks used in the syntax are defined as follows:

< >:	Shows a parameter required for sending a command
[ ] :	Shows that the command is optional It can be omitted
{ }:	Shows that only one item is required to be selected from multiple items
|:	Written in curly brackets {..} and used as a delimiter for multiple items
<ch>:	Written in the command header and shows the target input channel number of the command The channel number can be omitted. However, when it is written, the channel number 1 is selected

<screen>:	Written in the command header and shows the target screen number of the command The screen number can be omitted. However, when it is written, a value from 1 to 2 is selected [{1|2}]
<trace>:	Written in the command header and shows the target trace number of the command The trace number can be omitted. However, when it is written, a value from 1 to 4 is selected [{1|2|3|4}]

For example, when a syntax below is specified, :CALC:CORR:EDEL:TIME 0.1 and :CALCULATE1:SELECTED:CORR:EDEL:TIME 25E-3 are valid.

Syntax: CALCulate{[1]|2|3|4}[:SELected]:CORRection:EDELay:TIME <numeric value>

- [Function description]

  The usage of commands and operation of this instrument when they are executed.

- [Parameter]

  Describes a parameter required for sending a command.

  When the parameter is a numeric type or a character (string) type, it is enclosed in angle brackets (< >).

  When the parameter is an optional type, it is enclosed in curly brackets { }.

  In this manual, parameter types are described in the following formats:

< int >:	A numeric value that can be input in the format NR1, NR2, or NR3 and rounded to an integer in this instrument
< real >:	A numeric value that can be input in the format NR1, NR2, or NR3 and rounded to a valid-digit real number in this instrument
< bool >:	String of OFF|ON
< str >:	A character string or alphanumeric symbols enclosed in quotation (' ') or double quotation (“ ”) marks
< block >:	Block data type The content of data is an 8-bit binary data array
< type >:	Character data selected from multiple types

- [Query reply]

  When there is a query reply to the command, the data format used for reading the query is described.

  Each parameter to be read is enclosed in curly brackets { }. When multiple items delimited by a vertical bar (|) exist in curly brackets { }, only one of those items is read out. When multiple parameters are read out, they are delimited by commas (,). The three points (...) displayed between commas represent the data omitted in the position. For example, the description <numeric value 1>,..., <numeric value 4> shows that four parameters <numeric value 1>, <numeric value 2>, <numeric value 3>, and <numeric value 4> are read.

  When the parameter to be read is enclosed in square brackets [ ], the parameter may be omitted, depending on the measurement result, etc.

When the parameter to be read is a value in a unit, the description like "Unit: dBm" is added to display the unit of the parameter value. However, only when the parameter is described in a level unit "dBm", the level unit selected at that time will be applied to the parameter.

- [Example]

  Simple use examples of commands are described.

  Use examples of commands are described in Visual Basic language, using the GPIB programming interface of National Instruments.

  The function OutputMsgs( ) used in the sentence example is a function used for displaying the query result, etc. Implement in accordance with the application.

- [Relevant commands]

  Describes the relevant commands when there are relevant commands.

5.2 Common Commands

This section describes IEEE common commands.

Command	Function	Reference Page
*CLS	Clears the status byte and related data	5-4
*DDT	Macro definition for GET	5-5
*ESE	Sets the standard event status enable register	5-6
*ESR?	Reads the standard event status register	5-7
*IDN?	Device inquiry	5-8
*OPC	Notice of completion of all running operations	5-9
*RCL	Loads device settings	5-10
*RST	Resets the device	5-10
*SAV	Saves the device settings	5-11
*SRE	Sets the service request enable register	5-11
*STB?	Reads the status byte register	5-12
*TRG	Triggers the device	5-14
*TST?	Self-test execution and inquiry for the results	5-14
*WAI	Waits for the completion of all running operations	5-15

5.2.1 *CLS

- [Command syntax] *CLS
- [Function description] Clears the status byte and related data

  *CLS clears the status data structure and cancels *OPC and *OPC? compulsorily. It also clears the error queue. This command does not clear the output buffer. Therefore, the MAV bit is not cleared when output data exists. However, executing this command at the beginning of a line clears data.Therefore, all statuses including the MAV are cleared.
  *CLS clears the following items:
  - Error queue
  - Status byte register
  - Standard event status register
  - Standard operation status event register
  - Questionable status event register
  - Measuring status event register
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, "*CLS")

5.2.2 *DDT

- [Command syntax] *DDT <block>

  *DDT?
- [Function description] Macro definition for GET

  The *DDT command defines the command sequence to be executed when the *TRG command or GET interface message is received. This means that the *TRG command operation is replaced by a series of commands written in <block> data. A sequence up to 255 characters in length can be defined.

  Defining zero-length block data (#10) by the *DDT command means that nothing is executed by the *TRG command or GET interface message. By executing the *RST command, the macro is canceled.

  The response to the query is block data.

  If the *DDT? command is executed when the macro is undefined, zero-length block data (#10) is returned.

  Do not use the *TRG command during this definition. If the *TRG command is used during definition by the *DDT command, the sequence set by the *DDT command, not the trigger, is called, and an endless loop occurs (in this case, the nesting limit is exceeded and a macro error occurs).
- [Parameter] <block>: Command sequence
- [Query reply] <block>
- [Example] In the following example, *DDT #211:MEAS:CPOW? is sent, and :MEAS:CPOW? is executed by the *TRG command:

```
'------------ Define the macro sequence ----------------------
Call ibwrt ( analyzer%, "*DDT #211:MEAS:CPOW?" )

------------------------- Execute it -----------------------------
Call ibwrt (analyzer%, "*TRG")
```

5.2.3 *ESE

- [Command syntax] *ESE <int>

  *ESE?
- [Function description] Sets the standard event status enable register

  Sets the enable register for the standard event status register. The standard event status register corresponding to the bit set to 1 of this register is reflected in the status byte register as the valid bit.

  For more information, refer to the description of the status data structure.
- [Parameter] < int > = 0 - 255
- [Query reply] NR1 (integer)

  0 - 255
- [Example] For example, when Device Dependent Error(bit3) and Operation Complete(bit0) are set to enable, *ESE 9 is set as a result of calculation by $2^3 + 2^0 = 8 + 1 = 9$. This example is programmed as follows:

  Call ibwrt (analyzer%, "*ESE 9")
- [Relevant commands] *ESR?

5.2.4 *ESR?

- [Command syntax] *ESR?
- [Function description] Reads the standard event status register

Reads the standard event status register value.

The standard event status register value is cleared after being read, and the corresponding status byte bit (bit5) is also cleared. For more information, refer to the description of the status data structure.

Table 5-1 Allocation in the Standard Event Register

bit		Description
7	Power on	Set to 1 when power is turned on.
6	-	Always 0
5	Command Error	Set to 1 when the parser detects a syntax error.
4	Execution Error	Set to 1 when execution of an instruction received as a GPIB command fails for some reason (e.g., the parameter is out of range).
3	Device Dependent Error	Command Error, Execution Error, QueryError Set to 1 when an error other than the above occurs.
2	Query Error	Set to 1 when no data exists or data is lost when the controller tries to read data from this instrument.
1	Request Control	Set to 1 when this instrument is required to be an active controller.
0	Operation Complete	Set to 1 when there is no command that is being executed for this instrument after the *OPC command is received.

- [Parameter] None
- [Query reply] NR1 (integer)
  0 - 255
- [Example]

```
res$ = Space$(128)
Call ibwrt(analyzer%, "*OPC")
Call ibwrt(analyzer%, "*ESE 63")
'----- read ESR
Call ibwrt(analyzer%, "*ESR?")
Call ibrd(analyzer%, res$)
'----- analyze the bit pattern
If (Val(Esr$) And 1) > 0 Then OutputMsgs "Operation Complete"
If (Val(Esr$) And 2) > 0 Then OutputMsgs "Request Control"
If (Val(Esr$) And 4) > 0 Then OutputMsgs "Query Error"
If (Val(Esr$) Adn 8) > 0 Then OutputMsgs "Device Dependent Error"
If (VAL(Esr$) And 16) > 0 Then OutputMsgs "Execution Error"
If (VAL(Esr$) And 32) > 0 Then OutputMsgs "Command Error"
```

- [Relevant commands] *ESE

5.2.5 *IDN?

- [Command syntax] *IDN?
- [Function description] Device inquiry

  Acquires the identification information of this instrument. Outputs the four items described in the query reply below in character string format.
- [Parameter] None
- [Query reply] "manufacturer, model , serial number , firmware level"

  manufacturer = ADVANTEST

  model = Model name

  serial number = Serial number

  firmware level = System version
- [Example]

```
res$ = Space$(128)
'----- read IDN
Call ibwrt(analyzer%, "*IDN?")
Call ibrd(analyzer%, res$)
OutputMsgs "Instrument ID: " & res$
```

5.2.6 *OPC

- [Command syntax] *OPC
  *OPC?
- [Function description] Notice of completion of all running operations

  When all running commands are completed, the *OPC command sets Operation Complete bit of the standard event status register to 1. If the next command is received before all running commands are completed, it waits for the command to be completed. This means that when there is no command that is being executed for this instrument after the *OPC command is received, setting of the standard event status register is performed. The *OPC? command writes 1 in the output buffer instead of the Operation Complete bit set by the *OPC command shown above. This means that the timing of receipt of the response from this instrument by the controller agrees with the command completion timing. Both the *OPC and *OPC? commands are cancelled by the DCL interface message, *CLS command, and *RST command.
- [Parameter] None
- [Query reply] 1
- [Example]

```
Call ibwrt(analyzer%, "SENS:FREQ:CENT 1GHZ")
Call ibwrt(analyzer%, "SENS:FREQ:SPAN 100MHZ")
Call ibwrt(analyzer%, ":INP:ATT 30DB")

Call ibwrt(analyzer%, "*OPC")
Esr$ = Space$(20)
Do                                  ' Loop until setting
 Call ibwrt(analyzer%, "*ESR?")     ' operation complete bit
 Call ibrd(analyzer%, Esr$)
Loop Undil ((VAL(Esr$) AND 1) > 0)
```

- [Relevant commands] *WAI

5.2.7 *RCL

- [Command syntax] *RCL{<int> | POFF}
- [Function description] Loads device settings

  Calls the setting conditions of this instrument from the specified internal register. When the register number is 0 or POFF, the set value used when the power was turned off last time is loaded.
- [Parameter] {<int> | POFF}

  <int> = Register number (0 to 9999)

  POFF = The set value used when the power was turned off last time
- [Query reply] None
- [Example] Call ibwrt(analyzer%, "*RCL 5") ' Load all setting param
- [Relevant commands] *SAV

5.2.8 *RST

- [Command syntax] *RST
- [Function description] Resets the device

  The *RST command resets this instrument. Actually, the following operations are performed:
  - Initializes this instrument.
  - Initializes the macro defined by the *DDT command.
  - Invalidates the *OPC and *OPC? commands.
  - Resets the trigger system.

  The following items are not affected:
  - GPIB bus status
  - GPIB address
  - Output buffer
  - Status data structure
  - Device correction data
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt(analyzer%, "*RST") ' Reset instrument
- [Relevant commands] SYSTem:PRESet:ALL

5.2.9 *SAV

- [Command syntax] *SAV <int>
- [Function description] Saves the device settings

  Saves the setting conditions of this instrument in the save register with the designated number. The save register transforms the measurement conditions and measurement data into a file and saves it on the hard disk installed in this instrument.
- [Parameter] <int> = Internal register number for save (0 to 9999)
- [Query reply] None
- [Example]
```
Call ibwrt(analyzer%, ":SENS:FREQ:CENT 2GHZ")
Call ibwrt(analyzer%, ":INP:ATT 10DB")
'----- Save all setting parameter
Call ibwrt(analyzer%, "*SAV 5")
```
- [Relevant commands] *RCL

5.2.10 *SRE

- [Command syntax] *SRE <int>
  *SRE?
- [Function description] Sets the service request enable register

  Sets the service request enable register. The status byte register corresponding to the bit set to 1 of this register is reflected in the MSS as the valid bit. The response data bit6 to the query is always 0. For more information, refer to the description of the status data structure. Also refer to the *STB? command.

  For example, when OPR(bit7), ESB(bit5), and MAV(bit4) are set to enable, * SRE 176 is set as a result of calculation by $2^7 + 2^5 + 2^4 = 128 + 32 + 16 = 176$.
- [Parameter] <int> = NR1 (integer) The sum of bit information to be set to enable
- [Query reply] NR1 (integer)
- [Example]
```
'----------- Set OPR, ESB and MAV bit ----------
Call ibwrt(analyzer%, "*SRE 176")
```

5.2.11 *STB?

- [Command syntax] *STB?
- [Function description] Reads the status byte register

Reads the contents of the status byte register.

The summary bit requested to be read out is the MSS. This register and the MSS are not cleared after being read out. For more information, refer to the description of the status data structure.

Table 5-2 Allocation in the Standard Event Register

bit		Description
7	OPR	The OPR is the summary of the standard operation status register.
6	MSS	The RQS is set to TRUE when the MSS of the status byte register is set to 1, but the MSS is the summary bit of all of the status data structure.
		The MSS cannot be read by a service request (but it is known that the MSS is 1 when RQS is 1).
		To read the MSS, the common command *STB? is used. With *STB?, bit0 to bit5, bit7 of the status byte register, and the MSS are read. In this case, the status byte register and the MSS are not cleared.
		The MSS is not set to 0 until all the unmasked factors in the status register structure are cleared.
5	ESB	The ESB is the summary of the standard event register.
4	MAV	The MAV is the summary bit of the output buffer.
		It is set to 1 when there is output data in the output buffer and set to 0 after data is read.
3	QUES	The QUES is the summary of the questionable status register.
2	DEV	The DEV is the summary of the device status register.
1 to 0		Always 0

- [Parameter] None
- [Query reply] <int> = NR1 (integer)

- [Example]

```
res$ = Space$(128)
Call ibwrt(analyzer%, "*OPC")
Call ibwrt(analyzer%, "*SRE 184")
Call ibwrt(analyzer%, "*STB?")
Call ibrd(analyzer%, res$)
'----- analyze the bit pattern
If (Val(res$) And 4) > 0 Then OutputMsgs "Device Summary was set."
If (Val(res$) And 8) > 0 Then OutputMsgs "Questionable Sum. was set."
If (Val(res$) And 32) > 0 Then OutputMsgs "ESB Summary was set."
If (Val(res$) And 128) > 0 Then OutputMsgs "OPR Summary was set."
```

5.2.12 *TRG

- [Command syntax] *TRG
- [Function description] Triggers the device

  The *TRG command triggers the device.

  It has the same effect as the GET interface message.

  When this instrument receives the *TRG command, it starts measurement. Both the *TRG command and the GET interface message are entered into the input buffer and processed in the order they are entered.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt(analyzer%, "*TRG")
- [Relevant commands] GET

5.2.13 *TST?

- [Command syntax] *TST?
- [Function description] Self-test execution and inquiry for the results

  The *TST? command causes this instrument to perform a self-test and output the results.

  A value of 0 is returned upon successful self-test completion, and a value other than 0 is returned when error codes exist.
- [Parameter] None
- [Query reply] < 0 | int >

  < int > = Error code
- [Example]

```
res$ = Space$(128)
Call ibwrt(analyzer%, ":SYST:SEL SAN")
Call ibwrt(analyzer%, "*TST?")
'----- Read result
Call ibrd(analyzer%, res$)
If res$ = "0" Then
    OutputMsgs "Good Status!"
Else
    OutputMsgs "Error occurred! Error Status=" & res$
End If
```

5.2.14 *WAI

- [Command syntax] *WAI
- [Function description] Waits for the completion of all running operations

  The *WAI command waits for the completion of all running commands.

  Once this command is executed, all of the following commands will have to wait until the current command is terminated.

  The *WAI command is cancelled by the DCL interface message.
- [Parameter] None
- [Query reply] None
- [Example]

```
'--------- Set the center freq. To preselector band ------
Call ibwrt (analyzer%, ":FREQ:CENT 6.0GHZ")

'-------- Take one sweep ----------------------------
Call ibwrt (analyzer%, ":INIT:IMM")

'---------- Wait for end of sweep -------------------------
Call ibwrt(analyzer%, "*WAI")

'--------- Set the marker to the top of signal -------------
Call ibwrt (analyzer%, ":CALC:MARK:MAX:PEAK")

'----------------- Send next command -------------------
```

- [Relevant commands] *OPC

5.3 Input Command

This section describes the Input subsystem.

In the Input subsystem, the commands related to the RF input, such as the RF attenuator, are defined.

Command	Function	Reference Page
:INPut<ch>		
:ATTenuation<screen>	Setting the input RF attenuator	5-16
:AUTO	Selecting the input RF attenuator setting mode	5-17
:MINimum	Setting the minimum set value of the input RF attenuator	5-17
:STATe	Minimum setting function mode for the input RF attenuator	5-18
:GAIN<screen>		
:STATe	Setting the input gain amp function mode	5-19

5.3.1 :INPut<ch>:ATTenuation<screen>

- [Command syntax] :INPut<ch>:ATTenuation<screen> <real>
  :INPut<ch>:ATTenuation<screen>?
- [Function description] Sets the input RF attenuator

  Sets the input RF attenuator.
- [Parameter] <real> Input RF attenuator value (dB)
- [Query reply] NR3 (Real value: Unit dB)
- [Example] '--------- Set input attanuator to 30dB for RF channel #1 ---------
  Call ibwrt (analyzer%, ":INP1:ATT1 30DB")
- [Relevant commands] :INPut<ch>:ATTenuation<screen>:AUTO
  :INPut<ch>:ATTenuation<screen>:MINimum
  :INPut<ch>:ATTenuation<screen>:MINimum:STATe

5.3.2 :INPut<ch>:ATTenuation<screen>:AUTO

- [Command syntax] :INPut<ch>:ATTenuation<screen>:AUTO < bool >
  :INPut<ch>:ATTenuation<screen>:AUTO?
- [Function description] Selects the input RF attenuator setting mode

  Sets the input RF attenuator setting mode.
- [Parameter] < bool > = { OFF | ON }
  OFF: MANUAL mode
  ON: AUTO mode
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":INP1:ATTenuation1:AUTO OFF")
  Call ibwrt (analyzer%, ":INP1:ATT1 20DB")
- [Relevant commands] :INPut<ch>:ATTenuation<screen>
  :INPut<ch>:ATTenuation<screen>:MINimum
  :INPut<ch>:ATTenuation<screen>:MINimum:STATe

5.3.3 :INPut<ch>:ATTenuation<screen>:MINimum

- [Command syntax] :INPut<ch>:ATTenuation<screen>:MINimum < real >
  :INPut<ch>:ATTenuation<screen>:MINimum?
- [Function description] Sets the minimum set value of the input RF attenuator

  Sets the minimum set value of the input RF attenuator.
- [Parameter] < real > = Minimum set value of the input RF attenuator (dB)
- [Query reply] NR3 (Real value: Unit dB)
- [Example] '------- Set the limit of minimum att. value to 10dB for safety ---------
  Call ibwrt (analyzer%, ":INP1:ATT1:MIN:STAT ON")
  Call ibwrt (analyzer%, ":INP1:ATT1:MIN 10DB")

  '----------------- Cann't set the RF att. value to 0dB --------------------
  Call ibwrt (analyzer%, ":INP1:ATT1 0DB") ' Cann't set
- [Relevant commands] :INPut<ch>:ATTenuation<screen>
  :INPut<ch>:ATTenuation<screen>:MINimum:STATe

5.3.4 :INPut<ch>:ATTenuation<screen>:MINimum:STATe

- [Command syntax] :INPut<ch>:ATTenuation<screen>:MINimum:STATe < bool >
  :INPut<ch>:ATTenuation<screen>:MINimum:STATe?
- [Function description] Sets the minimum setting function mode for the input RF attenuator

  Sets whether to validate or invalidate the minimum setting function for the input RF attenuator.

  When the minimum setting function is valid, even if a value of the attenuator lower than the minimum value set by the command :INPut:ATTenuation:MINimum is specified, the minimum value is maintained as the lowest limit.
- [Parameter] < bool > = { ON | OFF }
  OFF : Function invalid (no limit)
  ON : Function valid (limited)
- [Query reply] { OFF | ON }
- [Example] '------- Release the limitation of setting the RF att. value ---------
  Call ibwrt (analyzer%, ":INP1:ATT1:MIN:STAT OFF")
  '------------------ Set it to 0dB ----------------
  Call ibwrt (analyzer%, ":INP1:ATT1 0DB")
- [Relevant commands] :INPut<ch>:ATTenuation<screen>
  :INPut<ch>:ATTenuation<screen>:MINimum

5.3.5 :INPut<ch>:GAIN<screen>:STATe

- [Command syntax] :INPut<ch>:GAIN<screen>:STATe < bool >
  :INPut<ch>:GAIN<screen>:STATe?
- [Function description] Sets the input gain amp function mode

  Switches between ON and OFF of the input gain amp.

  When the input gain amp state is switched, the reference level setting range will change.Therefore, the reference level may automatically switch, depending on the state. The following shows the reference level setting range when the gain amp is ON or OFF:

  OFF: -170dBm - +60dBm

  ON: -170dBm - +30dBm
- [Parameter] <bool> = { OFF | ON }

  OFF: Input gain amp off

  ON: Input gain amp on
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":INP:GAIN:STAT ON")

5.4 Sense Commands

This section describes the Sense subsystem.
In the Sense subsystem, basic setting commands such as frequency and sweep time setting are defined.

MEMO: ***The following notation is used only in the Sense subsystem.***
<win>: Written in the command header and indicates the number of the measurement window, in which the command is executed.
The measurement window number can be specified from 1 to 10. The window number can also be specified by {1|2|3|4|5|6|7|8|9|10}.

Command	Function	Reference Page
[:SENSe<ch>]		
:FREQuency<screen>		
:CENTer	Sets the center frequency	5-29
:STEP	Sets the center frequency setting resolution in Up/Down operation	5-32
:AUTO	Sets the center frequency setting resolution mode in Up/ Down operation	5-33
:STARt	Sets the start frequency	5-29
:STOP	Sets the stop frequency	5-30
:SPAN	Sets the span frequency	5-30
:FULL	Sets the span to the maximum span	5-31
:PREVious	Returns the span to the setting before span change	5-31
:ZERO	Sets the span to zero span	5-32
:OFFSet	Sets the offset frequency against the center frequency	5-33
:STATe	Sets the offset frequency state against the center frequency	5-34
:CHANnel		
:NUMBer	Sets the channel number	5-35
[:SENSe<ch>]		
:BANDwidth<screen>		
[:RESolution]	Sets the resolution bandwidth (RBW)	5-35
:AUTO	Selects the resolution bandwidth (RBW) setting mode	5-36
:RATio	Sets the setting ratio between the span frequency and the resolution bandwidth (RBW)	5-40
:STATe	Sets the setting ratio mode between the span frequency and the resolution bandwidth (RBW)	5-41
:VIDeo	Sets the video bandwidth (VBW)	5-36
:AUTO	Selects the video bandwidth (VBW) setting mode	5-37
:RATio	Sets the setting ratio between the resolution bandwidth (RBW) and the video bandwidth (VBW)	5-38
:STATe	Sets the setting ratio mode between the resolution bandwidth (RBW) and the video bandwidth (VBW)	5-39
:PLL	Selects the band-pass filter width in the PLL circuit	5-42

Command	Function	Reference Page
[:SENSe<ch>]		
:COUPle<screen>		
:ALL		
:AUTO	Sets the coupling setting items to automatic setting mode (RBW, VBW, sweep time)	5-43
[:SENSe<ch>]		
ADC<screen>		
:DITHer	Sets the ADC Dither function	5-43
[:SENSe<ch>]		
:DETector<screen>		
:TRACe		
[:NUMBer<trace>]		
:FUNCtion	Selects the trace detector	5-44
:AUTO	Selects the trace detector determination mode	5-45
:AVERage<screen>		
:TYPE	Selects the average detection mode of the average detector	5-46
:AUTO	Sets the mode to be used when selecting the average detection mode of the average detector	5-47
[:SENSe<ch>]		
:PRESelector<screen>	Manually adjusts the pre-selector filter	5-49
:AUTO	Executes automatic adjustment of the pre-selector filter	5-48
[:SENSe<ch>]		
:SWEep<screen>		
:TIME	Sets the sweep time	5-49
:AUTO	Selects the sweep time setting mode	5-50
:WINDow	Sets the window sweep to ON or OFF	5-50
[:SENSe<ch>]		
:SWEep		
:GATE	Sets the gated sweep to ON or OFF	5-51
:DELay	Sets the gate signal position	5-51
:WIDTh	Sets the gate signal width	5-52
:AUTO	Switches the gate signal mode	5-52
:SOURce	Sets the gated sweep trigger	5-53
:SLOPe	Sets the trigger polarity of each trigger source	5-53
:LEVel		
:EXTernal	Sets the trigger level when using an EXT2 (external input terminal 2) trigger	5-54
:IF	Sets the trigger level for IF trigger	5-54

Command	Function	Reference Page
[:SENSe]		
:ROSCillator		
:SOURce		
:FREQuency	Sets the frequency of the external frequency reference	5-55
:AUTO	Switches the frequency reference standard (internal/external)	5-56
:ADJust		
:COARse	Coarse adjustment of the correction value for adjusting the internal 10 MHz frequency reference	5-57
:FINE	Fine adjustment of the correction value for adjusting the internal 10 MHz frequency reference	5-58
:SAVE	Saves the correction value for adjusting the internal 10 MHz frequency reference	5-59
:DEFault	Clears the correction value for adjusting the internal 10 MHz frequency reference	5-59
[:SENSe<ch>]		
:CORRection		
:CSET		
:STATe	Switches the RF input level correction function ON or OFF	5-60
:DATA	Enters the RF input level correction data	5-60
:DELete	Clears all the RF input level correction data	5-61
[:SENSe<ch>]		
:SWEep<screen>		
:COUNt	Sets the sweep averaging count and MAX/MIN HOLD count	5-61
:AANalog		
:SAMPle		
:COUNt	Sets the quasi analog function and sampling count	5-62
[:SENSe<ch>]		
:CPOWer<screen>		
:AVERage		
:COUNt	Sets the averaging count in channel power measurement	5-62
[:STATe]	Sets the averaging operation mode to ON or OFF in channel power measurement	5-63
:MODE	Specifies the operation type in averaging operation mode in channel power measurement	5-64
:WINDow	Sets the measurement window display to ON or OFF in channel power measurement	5-65
:POSition	Specifies the measurement window display position in channel power measurement	5-66
:WIDTh	Specifies the measurement window display width in channel power measurement	5-67

Command	Function	Reference Page
[:SENSe<ch>]		
:CPOWer<screen>		
:DATA		
:MODE	Specifies the measurement parameter setting mode in channel power measurement	5-68
:SAVE	Saves the measurement parameters in channel power measurement	5-69
[:SENSe<ch>]		
:APOWer<screen>		
:AVERage		
:COUNt	Sets the averaging count in average power measurement	5-70
[:STATe]	Sets the averaging operation mode to ON or OFF in average power measurement	5-70
:MODE	Specifies the operation type in averaging operation mode in average power measurement	5-71
:WINDow	Sets the measurement window display to ON or OFF in average power measurement	5-72
:POSition	Specifies the measurement window display position in average power measurement	5-73
:WIDTh	Specifies the measurement window display width in average power measurement	5-74
:DATA		
:MODE	Specifies the measurement parameter setting mode in average power measurement	5-75
:SAVE	Saves the measurement parameters in average power measurement	5-76
[:SENSe<ch>]		
:OBW<screen>		
:AVERage		
:COUNt	Sets the averaging count in OBW measurement	5-77
[:STATe]	Sets the averaging operation mode to ON or OFF in OBW measurement	5-77
:MODE	Specifies the operation type in averaging operation mode in OBW measurement	5-78
:PERCent	Specifies the OBW% value in OBW measurement	5-78
:DATA		
:MODE	Specifies the measurement parameter setting mode in OBW measurement	5-79
:SAVE	Saves the measurement parameters in OBW measurement	5-80

Command	Function	Reference Page
[:SENSe<ch>]		
:ACP		
:AVERage		
:COUNt	Sets the averaging count in ACP measurement	5-81
[:STATe]	Sets the averaging operation mode to ON or OFF in ACP measurement	5-81
:MODE	Specifies the operation type in averaging operation mode in ACP measurement	5-82
:DATA		
:MODE	Specifies the measurement parameter setting mode in ACP measurement	5-83
:SAVE	Saves the measurement parameters in ACP measurement	5-84
:CBWidth	Sets the carrier bandwidth that becomes the target of the reference power operation in ACP measurement	5-85
:CSBW		
:DATA	Sets the adjacent channel position and adjacent channel band in ACP measurement	5-85
:DELete	Initializes the adjacent channel position and adjacent channel band data in ACP measurement	5-86
:RNYQuist	Sets the Root Nyquist filter operation mode to ON or OFF in ACP measurement	5-87
:SRATe	Sets the Symbol Rate for Root Nyquist filter operation in ACP measurement	5-88
:RFACtor	Sets the Rolloff Factor for Root Nyquist filter operation in ACP measurement	5-88
:NCORrection		
[:STATe]	Sets the noise correction function to ON or OFF in ACP measurement	5-89
:POWer		
:LEVel		
:AUTO	Executes the Auto Level Set function in ACP measurement	5-89

Command	Function	Reference Page
[:SENSe<ch>]		
:MCACp		
:RNYQuist	Sets Root Nyquist filter operation mode to ON or OFF in Multi Carrier ACP measurement	5-90
:SRATe	Sets the Symbol Rate for Root Nyquist filter operation in Multi Carrier ACP measurement	5-91
:RFACtor	Sets the Rolloff Factor for Root Nyquist filter operation in Multi Carrier ACP measurement	5-91
:AVERage		
:COUNt	Sets the averaging count in Multi Carrier ACP measurement	5-92
[:STATe]	Sets the averaging operation mode to ON or OFF in Multi Carrier ACP measurement	5-92
:MODE	Specifies the operation type in averaging operation mode in Multi Carrier ACP measurement	5-93
:DATA		
:MODE	Specifies the measurement parameter setting mode in Multi Carrier ACP measurement	5-94
:SAVE	Saves the measurement parameters in Multi Carrier ACP measurement	5-95
:PARameter{1|2|...|16}		
:STATe	Sets the measurement carrier and adjacent channel to ON or OFF in Multi Carrier ACP measurement	5-96
:FREQuency	Sets the offset frequency of the measurement carrier and adjacent channel in Multi Carrier ACP measurement	5-97
:BWIDth	Sets the channel bandwidth of the measurement carrier and adjacent channel in Multi Carrier ACP measurement	5-98
:REFerence	Sets the reference power area of the measurement carrier and adjacent channel in Multi Carrier ACP measurement	5-99
:LIMit	Sets the limit value for checking pass/fail of the measurement result	5-100
:NCORrection		
[:STATe]	Sets the noise correction function to ON or OFF in Multi Carrier ACP measurement	5-101
:POWer		
:LEVel		
:AUTO	Executes the Auto Level Set function in Multi Carrier ACP measurement	5-101
:CARRier		
:ADJust	Sets the Carrier Freq Adjustment in Multi Carrier ACP measurement	5-102
:STATe	Sets the Carrier Freq Adjustment function to ON or OFF in Multi Carrier ACP measurement	5-102

<table>
<tr><th>Command</th><th>Function</th><th>Reference Page</th></tr>
<tr><td>[:SENSe<ch>]
:SPURious
:DATA
[:NUMBer{1|2|3}]</td><td>Registers the sweep parameters to be used to the Spurious table in spurious measurement</td><td>5-103</td></tr>
<tr><td>:ACTive</td><td>Selects the Spurious table used in spurious measurement</td><td>5-104</td></tr>
<tr><td>:DELete</td><td>Clears all data registered in the Spurious table used in spurious measurement</td><td>5-104</td></tr>
<tr><td>:MODE</td><td>Selects the table use mode of the Spurious table used in spurious measurement</td><td>5-105</td></tr>
<tr><td>:SAVE</td><td>Saves the Spurious table used in spurious measurement</td><td>5-106</td></tr>
<tr><td>[:SENSe<ch>]
:IM
:LIM
[:STATe]</td><td>Sets ON or OFF the Pass/Fail judgment for distorted signals in IM measurement</td><td>5-107</td></tr>
<tr><td>:ORDer</td><td>Sets the maximum order of the signal to be measured in IM measurement</td><td>5-106</td></tr>
<tr><td>:THReshold{3|5|7|9}</td><td>Sets the Pass/Fail judgment value for distorted signal in IM measurement</td><td>5-107</td></tr>
<tr><td>:DATA
:MODE</td><td>Specifies the measurement parameter setting mode for IM measurement</td><td>5-108</td></tr>
<tr><td>:SAVE</td><td>Stores the measurement parameters for IM measurement</td><td>5-109</td></tr>
<tr><td>[:SENSe<ch>]
:HARMonics
:FFReqencey</td><td>Sets the reference signal frequency in harmonics measurement</td><td>5-110</td></tr>
<tr><td>:STATe</td><td>Sets the reference signal frequency mode in harmonics measurement</td><td>5-110</td></tr>
<tr><td>:NUMBer</td><td>Harmonics order to be measured in harmonics measurement</td><td>5-111</td></tr>
<tr><td>[:SENSe<ch>]
:SEMask
:CBWidth</td><td>Sets the reference power operation width in Spectrum Emission Mask measurement</td><td>5-112</td></tr>
<tr><td>:RNYQuist</td><td>Sets the Root Nyquist filter operation mode to ON or OFF in Spectrum Emission Mask measurement</td><td>5-113</td></tr>
<tr><td>:SRATe</td><td>Sets the Symbol Rate for Root Nyquist filter operation in Spectrum Emission Mask measurement</td><td>5-114</td></tr>
<tr><td>:RFACtor</td><td>Sets the Rolloff Factor for Root Nyquist filter operation in Spectrum Emission Mask measurement</td><td>5-114</td></tr>
</table>

Command	Function	Reference Page
[:SENSe<ch>]		
:SEMask		
:RPOWer		
:MODE	Sets the reference power calculation mode in Spectrum Emission Mask measurement	5-116
:AVERage		
:COUNt	Sets the averaging count during average measurement in Spectrum Emission Mask measurement	5-117
[:STATe]	Sets the averaging measurement function to ON or OFF in Spectrum Emission Mask measurement	5-117
:DATA	Sets the parameter table for measurement in Spectrum Emission Mask measurement	5-115
:DELete	Clears all data in the parameter table for measurement in Spectrum Emission Mask measurement	5-116
:MODE	Selects the setting parameters used for Spectrum Emission Mask measurement	5-119
:SAVE	Saves the setting parameters used in Spectrum Emission Mask measurement	5-120
:POWer		
:LEVel		
:AUTO	Executes the Auto Level Set function in Spectrum Emission Mask measurement	5-121
[:SENSe<ch>]		
:FCOunt<screen>		
:AVERage		
:COUNt	Averaging count in the frequency counter function	5-121
[:STATe]	Sets averaging in the frequency counter function to ON or OFF	5-122
[:SENSe<ch>]		
:CCDF		
:BANDwidth		
[:RESolution]	Sets the resolution bandwidth (RBW) in the CCDF measurement	5-122
:POINt	Sets the number of measurement samples in the CCDF measurement	5-123
:GATE	Sets the gate function in the CCDF measurement to ON or OFF	5-123
:THReshold	Sets the threshold level of the gate function in the CCDF measurement	5-124

Command	Function	Reference Page
[:SENSe<ch>]		
:MAPower		
:WINDow		
[:NUMBer<win>]	Sets the window display to ON or OFF in the Multi-Average Power measurement	5-124
:POSition	Specifies the window display position in the Multi-Average Power measurement	5-125
:WIDTh	Specifies the window display width in the Multi-Average Power measurement	5-125
:ACTive	Specifies the active window in the Multi-Average Power measurement	5-126
:RESet	Sets all windows excluding window No.1 to OFF	5-126
:COUPling	Sets the window display, which is coupled with the Average Power in the Multi-Average Power measurement, to ON or OFF	5-127
:PRATio	Sets the Power Ratio measurement to ON or OFF in the Multi-Average Power measurement	5-127
:AVERage		
:COUNt	Sets the number of times averaging is performed in the Multi-Average Power measurement	5-128
[:STATe]	Sets the averaging calculation mode to ON or OFF in the Multi-Average Power measurement	5-128
:MODE	Specifies the calculation type of the averaging calculation mode in the Multi-Average Power measurement	5-129
:DATA		
:MODE	Specifies the measurement parameter setting mode in the Multi-Average Power measurement	5-130
:SAVE	Saves the measurement parameters in the Multi-Average Power measurement	5-131

5.4.1 [:SENSe<ch>]:FREQuency<screen>:CENTer

- [Command syntax] [:SENSe<ch>]:FREQuency<screen>:CENTer <real>
  [:SENSe<ch>]:FREQuency<screen>:CENTer?
- [Function description] Sets the center frequency

  This command sets the center frequency.
- [Parameter] <real> = Setting center frequency (GHz/MHz/kHz/Hz)
- [Query reply] NR3 (Real value: Unit Hz)
- [Example] Call ibwrt (analyzer%, ":FREQ:CENT 1GHZ") ' Set center freq.
  Call ibwrt (analyzer%, ":FREQ:SPAN 500MHZ") ' Set span freq.
- [Relevant commands] [:SENSe<ch>]:FREQuency<screen>:STARt
  [:SENSe<ch>]:FREQuency<screen>:STOP
  [:SENSe<ch>]:FREQuency<screen>:SPAN

5.4.2 [:SENSe<ch>]:FREQuency<screen>:STARt

- [Command syntax] [:SENSe<ch>]:FREQuency<screen>:STARt <real>
  [:SENSe<ch>]:FREQuency<screen>:STARt?
- [Function description] Sets the start frequency

  This command sets the start frequency.
- [Parameter] <real> = Setting start frequency (GHz/MHz/kHz/Hz)
- [Query reply] NR3 (Real value: Unit Hz)
- [Example] Call ibwrt (analyzer%, ":FREQ:STAR 500MHZ") ' Set start freq.
  Call ibwrt (analyzer%, ":FREQ:STOP 1GHZ") ' Set stop freq.
- [Relevant commands] [:SENSe<ch>]:FREQuency<screen>:CENTer
  [:SENSe<ch>]:FREQuency<screen>:STOP
  [:SENSe<ch>]:FREQuency<screen>:SPAN

5.4.3 [:SENSe<ch>]:FREQuency<screen>:STOP

• [Command syntax]	[:SENSe<ch>]:FREQuency<screen>:STOP <real> [:SENSe<ch>]:FREQuency<screen>:STOP?
• [Function description]	Sets the stop frequency. This command sets the stop frequency.
• [Parameter]	< real > = Setting stop frequency (GHz/MHz/kHz/Hz)
• [Query reply]	NR3 (Real value: Unit Hz)
• [Example]	Call ibwrt (analyzer%, ":FREQ:STAR 500MHZ") ' Set start freq. Call ibwrt (analyzer%, ":FREQ:STOP 1GHZ") ' Set stop freq.
• [Relevant commands]	[:SENSe<ch>]:FREQuency<screen>:CENTer [:SENSe<ch>]:FREQuency<screen>:STARt [:SENSe<ch>]:FREQuency<screen>:SPAN

5.4.4 [:SENSe<ch>]:FREQuency<screen>:SPAN

• [Command syntax]	[:SENSe<ch>]:FREQuency<screen>:SPAN < real > [:SENSe<ch>]:FREQuency<screen>:SPAN?
• [Function description]	Sets the span frequency This command sets the span frequency.
• [Parameter]	<real> = Setting span frequency (GHz/MHz/kHz/Hz)
• [Query reply]	NR3 (Real value: Unit Hz)
• [Example]	Call ibwrt (analyzer%, ":FREQ:CENT 2.1GHZ") ' Set center freq. Call ibwrt (analyzer%, ":FREQ:SPAN 10MHZ") ' Set span freq.
• [Relevant commands]	[:SENSe<ch>]:FREQuency<screen>:STARt [:SENSe<ch>]:FREQuency<screen>:STOP

5.4.5 [:SENSe<ch>]:FREQuency<screen>:SPAN:FULL

- [Command syntax] [:SENSe<ch>]:FREQuency<screen>:SPAN:FULL
- [Function description] Sets the span to the maximum span

  This command sets the span frequency to the maximum span.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":FREQ:SPAN:FULL")
- [Relevant commands] [:SENSe<ch>]:FREQuency<screen>:SPAN:PREVious
  [:SENSe<ch>]:FREQuency<screen>:SPAN
  [:SENSe<ch>]:FREQuency<screen>:SPAN:ZERO

5.4.6 [:SENSe<ch>]:FREQuency<screen>:SPAN:PREVious

- [Command syntax] [:SENSe<ch>]:FREQuency<screen>:SPAN:PREVious
- [Function description] Returns the span frequency to the setting before span change

  This command returns the span frequency to the value immediately before the span frequency last set.
- [Parameter] None
- [Query reply] None
- [Example]

```
Call ibwrt (analyzer%, ":FREQ:CENT 2.1GHZ")
'-------------------- Set span freq. to 10MHz ----------------------
Call ibwrt (analyzer%, ":FREQ:SPAN 10MHZ")
'------------ Change the span freq. setting start freq. -----------
Call ibwrt (analyzer%, ":FREQ:STAR 1.4GHZ")

'------------ Re-set the span freq. to 10MHz ---------------------
Call ibwrt (analyzer%, ":FREQ:SPAN:PREV")
```

- [Relevant commands] [:SENSe<ch>]:FREQuency<screen>:SPAN
  [:SENSe<ch>]:FREQuency<screen>:SPAN:FULL
  [:SENSe<ch>]:FREQuency<screen>:SPAN:ZERO

5.4.7 [:SENSe<ch>]:FREQuency<screen>:SPAN:ZERO

- [Command syntax] [:SENSe<ch>]:FREQuency<screen>:SPAN:ZERO
- [Function description] Sets the span to zero span

  This command sets the span frequency to zero span.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":FREQ:SPAN:ZERO")
- [Relevant commands] [:SENSe<ch>]:FREQuency<screen>:SPAN
  [:SENSe<ch>]:FREQuency<screen>:SPAN:FULL
  [:SENSe<ch>]:FREQuency<screen>:SPAN:PREVious

5.4.8 [:SENSe<ch>]:FREQuency<screen>:CENTer:STEP

- [Command syntax] [:SENSe<ch>]:FREQuency<screen>:CENTer:STEP < real >
  [:SENSe<ch>]:FREQuency<screen>:CENTer:STEP?
- [Function description] Sets the center frequency setting resolution in Up/Down operation

  This command sets the center frequency setting resolution (in Up/Down operation).
- [Parameter] <real> = Setting frequency resolution (GHz/MHz/kHz/Hz)
- [Query reply] NR3 (Real value: Unit Hz)
- [Example] Call ibwrt (analyzer%, ":FREQ:CENT:STEP 5MHZ")
  Call ibwrt (analyzer%, ":FREQ:CENT:STEP:AUTO OFF")
- [Relevant commands] [:SENSe<ch>]:FREQuency<screen>:CENTer:STEP:AUTO

5.4.9 [:SENSe<ch>]:FREQuency<screen>:CENTer:STEP:AUTO

- [Command syntax] [:SENSe<ch>]:FREQuency<screen>:CENTer:STEP:AUTO < bool >
  [:SENSe<ch>]:FREQuency<screen>:CENTer:STEP?
- [Function description] Sets the center frequency setting resolution mode in Up/Down operation

  This command selects the center frequency setting resolution mode in Up/Down operation.In Auto mode, the resolution is set to 1/10 of the span. In another mode, the specified value is used.
- [Parameter] < bool > = { OFF | ON }
  OFF: The resolution set in [:SENSe<ch>]:FREQuency<screen> :CENTer:STEP is used.
  ON: The center frequency setting resolution is set to 1/10 of the span.
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":FREQ:CENT:STEP:AUTO ON")
- [Relevant commands] [:SENSe<ch>]:FREQuency<screen>:CENTer:STEP

5.4.10 [:SENSe<ch>]:FREQuency<screen>:OFFSet

- [Command syntax] [:SENSe<ch>]:FREQuency<screen>:OFFSet < real >
  [:SENSe<ch>]:FREQuency<screen>:OFFSet?
- [Function description] Sets the offset frequency against the center frequency

  This command sets the display offset frequency against the actually set center frequency. When the offset frequency is enabled, the center frequency calculated from the following equation is set to this instrument as the actual setting value for the subsequent center frequency setting commands.

  When this command is used, the offset frequency function is automatically enabled.

  Actually set center frequency = Specified center frequency - Offset frequency

- [Parameter] <real> = Setting offset frequency (GHz/MHz/kHz/Hz)
- [Query reply] NR3 (Real value) (Hz)

- [Example]

```
'---------- Set the freq. offset to 100MHz ------------
Call ibwrt (analyzer%, ":FREQ:OFFS 100MHZ")
'---------- Activate the offset freq. --------------------

'------ Set the center freq. to 1.1GHz (real freq. 1GHz)
Call ibwrt (analyzer%, ":FREQ:CENT 1100MHZ")
```

- [Relevant commands] [:SENSe<ch>]:FREQuency<screen>:OFFSet:STATe

5.4.11 [:SENSe<ch>]:FREQuency<screen>:OFFSet:STATe

- [Command syntax] [:SENSe<ch>]:FREQuency<screen>:OFFSet:STATe < bool >
  [:SENSe<ch>]:FREQuency<screen>:OFFSet:STATe?
- [Function description] Sets the offset frequency state against the center frequency

  This command sets the display offset frequency state against the actually set center frequency. If ON is specified, the offset frequency is enabled.

  When the offset frequency is enabled, the center frequency calculated from the following equation is set to this instrument as the actual setting value for the subsequent center frequency setting commands.

  Actually set center frequency = Specified center frequency - Offset frequency

- [Parameter] < bool > = { OFF | ON }
  OFF: Offset frequency is disabled.
  ON: Offset frequency is enabled.
- [Query reply] { OFF | ON }
- [Example]

```
'---------- Activate the offset freq. --------------------
Call ibwrt (analyzer%, ":FREQ:OFFS:STAT ON")
```

- [Relevant commands] [:SENSe<ch>]:FREQuency<screen>:OFFSet

5.4.12 [:SENSe<ch>]:FREQuency<screen>:CHANnel:NUMBer

- [Command syntax] [:SENSe<ch>]:FREQuency<screen>:CHANnel:NUMBer < int >
  [:SENSe<ch>]:FREQuency<screen>:CHANnel:NUMBer?
- [Function description] Sets the channel number.

  When the channel number is set, the center frequency is automatically set by using the following formula.

  (Center frequency) = (Channel interval) × (Channel number + Channel offset) + (Start frequency)

  The parameter such as channel interval and the channel number setting range depend on the selected Standard. If the Standard selection is set to OFF, the channel number cannot be set.
- [Parameter] <int> = Integer value
- [Query reply] NR1 (Integer)
- [Example] Call ibwrt (analyzer%, ":FREQ:CHAN:NUMB 1000")
- [Relevant commands] [:SENSe<ch>]:FREQuency<screen>:CENTer
  :SYSTem:SELect:STANdard

5.4.13 [:SENSe<ch>]:BANDwidth<screen> | BWIDth<screen>[:RESolution]

- [Command syntax] [:SENSe<ch>]:BANDwidth<screen>[:RESolution] < real >
  [:SENSe<ch>]:BWIDth<screen>[:RESolution] < real >
  [:SENSe<ch>]:BANDwidth<screen>[:RESolution]?
  [:SENSe<ch>]:BWIDth<screen>[:RESolution]?
- [Function description] Sets the resolution bandwidth (RBW)

  This command sets the resolution bandwidth (RBW). Available RBW values are not sequential. Therefore, if a parameter value that was sent cannot be set as an RBW value, the nearest RBW value available is selected.
- [Parameter] <real> = Resolution bandwidth (MHz/kHz/Hz)
- [Query reply] NR3 (Real value: Unit Hz)
- [Example] '---------- Set the RBW filter to 100kHz -------------
  Call ibwrt (analyzer%, ":BAND:RES 100KHZ")
  or
  Call ibwrt (analyzer%, ":BWID:RES 100KHZ")
- [Relevant commands] [:SENSe<ch>]:BANDwidth<screen>[:RESolution]:AUTO
  [:SENSe<ch>]:BWIDth<screen>[:RESolution]:AUTO

5.4.14 [:SENSe<ch>]:BANDwidth<screen> | BWIDth<screen>[:RESolution]:AUTO

- [Command syntax] [:SENSe<ch>]:BANDwidth<screen>[:RESolution]:AUTO <bool>
  [:SENSe<ch>]:BWIDth<screen>[:RESolution]:AUTO <bool>
  [:SENSe<ch>]:BANDwidth<screen>[:RESolution]:AUTO?
  [:SENSe<ch>]:BWIDth<screen>[:RESolution]:AUTO?
- [Function description] Selects the resolution bandwidth (RBW) setting mode

  This command selects the resolution bandwidth (RBW) setting mode. In AUTO state, this instrument determines and sets the RBW value automatically, depending on the span frequency. In MANUAL state, the value is fixed to the specified RBW value.
- [Parameter] < bool > = { OFF | ON }
  ON : Set to AUTO
  OFF: Set to MANUAL
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":BAND:RES:AUTO ON")
- [Relevant commands] [:SENSe<ch>]:BANDwidth<screen>[:RESolution]
  [:SENSe<ch>]:BWIDth<screen>[:RESolution]

5.4.15 [:SENSe<ch>]:BANDwidth<screen> | BWIDth<screen>:VIDeo

- [Command syntax] [:SENSe<ch>]:BANDwidth<screen>:VIDeo < real >
  [:SENSe<ch>]:BWIDth<screen>:VIDeo < real >
  [:SENSe<ch>]:BANDwidth<screen>:VIDeo?
  [:SENSe<ch>]:BWIDth<screen>:VIDeo?
- [Function description] Sets the video bandwidth (VBW)

  This command sets the video bandwidth (VBW). Available VBW values are not sequential. Therefore, if a parameter value that was sent cannot be set as a VBW value, the nearest VBW value available is selected.
- [Parameter] <real> = Setting video bandwidth (MHz/kHz/Hz)
- [Query reply] NR3 (Real value: Unit Hz)
- [Example] '---------- Set the Video filter to 300kHz -------------
  Call ibwrt (analyzer%, ":BAND:VID 300KHZ")
  or
  Call ibwrt (analyzer%, ":BWID:VID 300KHZ")
- [Relevant commands] [:SENSe<ch>]:BANDwidth<screen>:VIDeo:AUTO
  [:SENSe<ch>]:BWIDth<screen>:VIDeo:AUTO

5.4.16 [:SENSe<ch>]:BANDwidth<screen> | BWIDth<screen>:VIDeo:AUTO

- [Command syntax] [:SENSe<ch>]:BANDwidth<screen>:VIDeo:AUTO <bool>
  [:SENSe<ch>]:BWIDth<screen>:VIDeo:AUTO <bool>
  [:SENSe<ch>]:BANDwidth<screen>:VIDeo:AUTO?
  [:SENSe<ch>]:BWIDth<screen>:VIDeo:AUTO?
- [Function description] Selects the video bandwidth (VBW) setting mode

  This command selects the video bandwidth (VBW) setting mode.
  In AUTO state, this instrument determines and sets the VBW value automatically, depending on the following parameter setting values.
  In MANUAL state, the value is fixed to the specified VBW value.
  (Related parameters: Span frequency, RBW, measurement mode)
- [Parameter] < bool > = { OFF | ON }
  ON : Set to AUTO
  OFF: Set to MANUAL
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":BAND:VID:AUTO OFF")
- [Relevant commands] [:SENSe<ch>]:BANDwidth<screen>:VIDeo
  [:SENSe<ch>]:BWIDth<screen>:VIDeo

5.4.17 [:SENSe<ch>]:BANDwidth<screen> | BWIDth<screen>:VIDeo:RATio

- [Command syntax] [:SENSe<ch>]:BANDwidth<screen>:VIDeo:RATio < real >
  [:SENSe<ch>]:BWIDth<screen>:VIDeo:RATio < real >
  [:SENSe<ch>]:BANDwidth<screen>:VIDeo:RATio?
  [:SENSe<ch>]:BWIDth<screen>:VIDeo:RATio?
- [Function description] Sets the ratio between the resolution bandwidth (RBW) and the video bandwidth (VBW)

  This command sets the ratio between the resolution bandwidth (RBW) and the video bandwidth (VBW).
  In accordance with this ratio, the VBW value against the RBW value is determined.
  This ratio is defined by (VBW/RBW).

- [Parameter] <real> = VBW/RBW ratio
  Setting range: 0.001 - 10.000
- [Query reply] NR3 (Real value)
- [Relevant commands] '---------- Set the VBW ratio to 3times of RBW ---------------
  Call ibwrt (anlayzer%, ":BAND:VID:RAT 3.0")
- [Relevant commands] [:SENSe<ch>]:BANDwidth<screen>:VIDeo:RATio:STATe

5.4.18 [:SENSe<ch>]:BANDwidth<screen> | BWIDth<screen>:VIDeo:RATio:STATe

- [Command syntax] [:SENSe<ch>]:BANDwidth<screen>:VIDeo:RATio:STATe < bool >
  [:SENSe<ch>]:BWIDth<screen>:VIDeo:RATio:STATe < bool >
  [:SENSe<ch>]:BANDwidth<screen>:VIDeo:RATio:STATe?
  [:SENSe<ch>]:BWIDth<screen>:VIDeo:RATio:STATe?
- [Function description] Sets the mode of the ratio between the resolution bandwidth (RBW) and the video bandwidth (VBW)

  This command sets the mode of the ratio between the resolution bandwidth (RBW) and the video bandwidth (VBW). In AUTO state, the VBW value is usually determined against the RBW value in the ratio "1:1" except in states such as power value measurement.

  In MANUAL state, it is determined in accordance with the ratio set in [:SENSe<ch>]:BANDwidth<screen>:VIDeo:RATio command.
- [Parameter] < bool > = { OFF | ON }
  ON: AUTO state
  OFF: MANUAL state
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, :BAND:VID:RAT:STAT ON")
- [Relevant commands] [:SENSe<ch>]:BANDwidth<screen>:VIDeo:RATio

5.4.19 [:SENSe<ch>]:BANDwidth<screen> | BWIDth<screen>[:RESolution]:RATio

- [Command syntax]
  [:SENSe<ch>]:BANDwidth<screen>[:RESolution]:RATio < real >
  [:SENSe<ch>]:BWIDth<screen>[:RESolution]:RATio < real >
  [:SENSe<ch>]:BANDwidth<screen>[:RESolution]:RATio?
  [:SENSe<ch>]:BWIDth<screen>[:RESolution]:RATio?
- [Function description]
  Sets the ratio between the span frequency and the resolution bandwidth (RBW)

  This command sets the ratio between the span frequency and the resolution bandwidth (RBW).
  In accordance with this ratio, the RBW value against the span frequency is determined.
  This setting ratio is defined by (span frequency/RBW).
- [Parameter]
  <real> = Ratio of Span frequency to RBW
  Setting range: 2 - 1000
- [Query reply]
  NR3 (Real value)
- [Example]
  '-------- Set the RBW ratio to 1/10 of span -------
  Call ibwrt (analyzer%, ":BAND:RES:RAT 10")
- [Relevant commands]
  [:SENSe<ch>]:BANDwidth<screen>:VIDeo:RATio:STATe

5.4.20 [:SENSe<ch>]:BANDwidth<screen>[:RESolution]:RATio:STATe [:SENSe<ch>]:BWIDth<screen>[:RESolution]:RATio:STATe

- [Command syntax] [:SENSe<ch>]:BANDwidth<screen>:VIDeo:RATio:STATe < bool >
  [:SENSe<ch>]:BWIDth<screen>:VIDeo:RATio:STATe < bool >
  [:SENSe<ch>]:BANDwidth<screen>:VIDeo:RATio:STATe?
  [:SENSe<ch>]:BWIDth<screen>:VIDeo:RATio:STATe?
- [Function description] Mode setting of the ratio between the span frequency and the resolution bandwidth (RBW)

  This command sets the mode of the ratio between the span frequency and the resolution bandwidth (RBW). In AUTO state, the ratio of the span frequency to the RBW value is always "10:1."

  In MANUAL state, it is determined in accordance with the ratio set in [:SENSe<ch>]:BANDwidth<screen>[:RESolution]:RATio command.
- [Parameter] < bool > = { OFF | ON }
  ON: AUTO state
  OFF: MANUAL state
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":BAND:RES:RAT:STAT ON")
- [Relevant commands] [:SENSe<ch>]:BANDwidth<screen>[:RESolution]:RATio

5.4.21 [:SENSe<ch>]:BANDwidth<screen> | BWIDth<screen>:PLL

- [Command syntax] [:SENSe<ch>]:BANDwidth<screen>:PLL < type >
  [:SENSe<ch>]:BWIDth<screen>:PLL < type >
  [:SENSe<ch>]:BANDwidth<screen>:PLL?
  [:SENSe<ch>]:BWIDth<screen>:PLL?
- [Function description] Selects the loop filter width in the PLL circuit

  This command sets the loop filter width in the PLL circuit inside this instrument. The phase noise characteristic from the carrier varies depending on this filter width. Select the filter width from NARROW, MEDIUM, and WIDE, depending on how far from the carrier the adjacent phase noise to be reduced is.

  The default is AUTO mode, in which the optimum filter width is selected depending on the setting span.
- [Parameter] < type > = { AUTO | NARRow | MEDium | WIDE }

  AUTO: Sets the filter width automatically so that the optimum phase noise characteristics corresponding to the frequency span can be obtained

  NARRow:Sets the filter width to Narrow
  The phase noise around 100 kHz from the carrier is improved

  MEDium:Sets the filter width to Medium

  WIDE: Sets the filter width to Wide
  The phase noise around 10 kHz from the carrier is improved
- [Query reply] { AUTO | NARR | MED | WIDE }
- [Example] Call ibwrt (analyzer%, ":BAND:PLL NARR")
- [Relevant commands] [:SENSe<ch>]:FREQuency<screen>:SPAN

5.4.22 [:SENSe<ch>]:COUPle<screen>:ALL:AUTO

- [Command syntax] [:SENSe<ch>]:COUPle<screen>:ALL:AUTO
- [Function description] Sets the coupling setting items to automatic setting mode

  This command sets the setting mode of RBW, VBW, and the sweep time to automatic setting mode. By using this command, this instrument automatically calculates and sets RBW, VBW, and the sweep time based on the frequency span.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":COUP:ALL:AUTO")
- [Relevant commands] [:SENSe<ch>]:FREQuency<screen>:SPAN
  [:SENSe<ch>]:BANDwidth<screen>[:RESolution]:AUTO
  [:SENSe<ch>]:BWIDth<screen>[:RESolution]:AUTO
  [:SENSe<ch>]:BANDwidth<screen>:VIDeo:AUTO
  [:SENSe<ch>]:BWIDth<screen>:VIDeo:AUTO

5.4.23 [:SENSe<ch>]:ADC<screen>:DITHer

- [Command syntax] [:SENSe<ch>]:ADC<screen>:DITHer <bool>
  [:SENSe<ch>]:ADC<screen>:DITHer?
- [Function description] Sets the ADC Dither function

  This command sets the ADC Dither function to ON or OFF.

  When the ADC Dither is set to ON, the ADC linearity for low level signal is improved and the internally-occurred intermodulation distortion is effectively restrained. When measuring the intermodulation distortion, set ADC Dither to ON.

  On the other hand, the ADC Dither adversely affects the displayed average noise level. For the lower noise level measurement, set the ADC Dither to OFF.
- [Parameter] < bool > = { OFF | ON }
  OFF: ADC Dither is off.
  ON: ADC Dither is on.
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":ADC:DITH ON")
- [Relevant commands]

5.4.24 [:SENSe<ch>]:DETector<screen>:TRACe[:NUMBer<trace>]:FUNCtion

- [Command syntax] [:SENSe<ch>]:DETector<screen>:TRACe[:NUMBer<trace>]:FUNCtion < type >

  [:SENSe<ch>]:DETector<screen>:TRACe[:NUMBer<trace>]:FUNCtion?
- [Function description] Selects the trace detector

  This command selects the detector of the specified trace. When the average detector is selected, one averaging method specified in Average Type (RMS/Video/Voltage) is set.
- [Parameter] < type > = { NORMal | POSitive | NEGative | SAMPle | AVERage }

  NORMal: Normal detector

  POSitive: Positive detector

  NEGative: Negative detector

  SAMPle: Sample detector

  AVERage: Average detector(RMS, Video, or Voltage)
- [Query reply] { NORM | POS | NEG | SAMP | AVER }
- [Example] '------ Set the detector of trace no.1 to positive detector ------

  Call ibwrt (analyzer%, ":DET:TRAC:NUMB1:FUNC POS")
- [Relevant commands] [:SENSe<ch>]:DETector<screen>:TRACe[:NUMBer<trace>]:FUNCtion:AUTO

  [:SENSe<ch>]:AVERage<screen>:TYPE

5.4.25 [:SENSe<ch>]:DETector<screen>:TRACe[:NUMBer<trace>]:FUNCtion :AUTO

- [Command syntax] [:SENSe<ch>]:DETector<screen>:TRACe[:NUMBer<trace>] :FUNCtion:AUTO < bool >

  [:SENSe<ch>]:DETector<screen>:TRACe[:NUMBer<trace>] :FUNCtion:AUTO?

- [Function description] Selects the trace detector determination mode

  This command sets the determination mode of the detector of the specified trace. When AUTO ON is selected, an appropriate detector is automatically set, depending on the measurement mode.

  When AUTO OFF is selected, the specified detector is operated regardless of the measurement mode and the detector is not changed unless another detector is specified.

- [Parameter] < bool > = { OFF | ON }

  OFF: In this mode, the detector is not changed automatically depending on the measurement mode.

  ON: In this mode, an appropriate detector is set automatically, depending on the measurement mode such as power measurement, OBW measurement, or adjacent channel leakage power (ACP).

- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":DET:TRAC:NUMB1:FUNC:AUTO OFF")

  '---------------- Set the trace detector manualy ----------------

  Call ibwrt (analyzer%, ":DET:TRAC:NUMB1:FUNC SAMP")

- [Relevant commands] [:SENSe<ch>]:DETector<screen>:TRACe[:NUMBer<trace>] :FUNCtion

5.4.26 [:SENSe<ch>]:AVERage<screen>:TYPE

- [Command syntax]
  [:SENSe<ch>]:AVERage<screen>:TYPE < type >
  [:SENSe<ch>]:AVERage<screen>:TYPE?
- [Function description]
  Selects the average detection mode of the average detector

  This command specifies the average mode used when selecting the average detector.
  There are three types of average modes: RMS, Video, and Voltage.
- [Parameter]
  < type > = { RMS | VIDeo | VOLTage }
  RMS: Set to the RMS average detection.
  VIDeo: Set to the Video average detection.
  VOLTage: Set to the Voltage average detection.
- [Query reply]
  { RMS | VID | VOLT }
- [Example]
  '--------- Set to the RMS average type manualy ----------
  Call ibwrt (analyzer%, ":AVER:TYPE RMS")
- [Relevant commands]
  [:SENSe<ch>]:DETector<screen>:TRACe[:NUMBer<trace>]:FUNCtion
  [:SENSe<ch>]:DETector<screen>:TRACe[:NUMBer<trace>]:FUNCtion:AUTO

5.4.27 [:SENSe<ch>]:AVERage<screen>:TYPE:AUTO

- [Command syntax] [:SENSe<ch>]:AVERage<screen>:TYPE:AUTO < bool >
  [:SENSe<ch>]:AVERage<screen>:TYPE:AUTO?
- [Function description] Sets the mode to be used when selecting the average detection mode of the average detector

  This command sets the mode of interlocking the average detection mode setting with the measurement function to ON or OFF.

  When the AUTO mode is set to ON, an appropriate average detection mode is set, depending on the measurement function such as power measurement or the adjacent channel leakage power measurement. When AUTO mode is set to OFF, the mode is fixed to the average detection specified with the [:SENSe<ch>]:AVERage<screen>:TYPE command and it is not interlocked with the measurement function.
- [Parameter] < bool > = { OFF | ON }
  OFF: Set to AUTO mode off of the average detection mode.
  ON: Set to AUTO mode on of the average detection mode.
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":AVER:TYPE:AUTO ON")
- [Relevant commands] [:SENSe<ch>]:AVERage<screen>:TYPE

5.4.28 [:SENSe<ch>]:PRESelector<screen>:AUTO

- [Command syntax] [:SENSe<ch>]:PRESelector<screen>:AUTO
- [Function description] Executes automatic adjustment of the pre-selector filter

  This command executes automatic adjustment of the pre-selector filter to improve the level accuracy at the frequency position at which the marker is placed.

  Adjustment is made so that the signal level at the marker position is maximized. When this command is received, this instrument suspends other measurements and starts auto-adjustment of the pre-selector filter.

  After automatic adjustment is completed, the "CALIbration" bit in the standard operation status register is set.
- [Parameter] None
- [Query reply] None
- [Example]

```
'--------- Set the center freq. To preselector band ------
Call ibwrt (analyzer%, ":FREQ:CENT 6.0GHZ")
'--------- Set the marker to the top of signal -------------
Call ibwrt (analyzer%, ":CALC:MARK:MAX:PEAK")

'-------- Execute the adjustment of pre-selector ------
Call ibwrt (analyzer%, ":PRES:AUTO")

'----------------- Send next command --------------------
```

- [Relevant commands] :STATus:OPERation:EVENt?
  [:SENSe<ch>]:PRESelector<screen>

5.4.29 [:SENSe<ch>]:PRESelector<screen>

- [Command syntax] [:SENSe<ch>]:PRESelector<screen> < int >
  [:SENSe<ch>]:PRESelector<screen>?
- [Function description] Manually adjusts the pre-selector filter

  This command sets the adjustment value of the pre-selector filter at the frequency position at which the marker is placed. As the adjustment value of the pre-selector filter is set, the level at the marker position fluctuates.While reading the marker value, make an adjustment so that the level value is maximized.
- [Parameter] <int> = Adjustment value
  Setting range: -100 - 100
- [Query reply] NR1 (Integer)
- [Example]
  '--------- Set the center freq. To preselector band ------
  Call ibwrt (analyzer%, ":FREQ:CENT 6.0GHZ")
  '--------- Set the marker to the top of signal -------------
  Call ibwrt (analyzer%, ":CALC:MARK:MAX:PEAK")

  '------------- Adjust the pre-selector manualy ---------
  Call ibwrt (analyzer%, ":PRES 78")
- [Relevant commands] [:SENSe<ch>]:PRESelector<screen>:AUTO

5.4.30 [:SENSe<ch>]:SWEep<screen>:TIME

- [Command syntax] [:SENSe<ch>]:SWEep<screen>:TIME < real >
  [:SENSe<ch>]:SWEep<screen>:TIME?
- [Function description] Sets the sweep time

  This command sets the sweep time. The sweep time that can be set varies depending on the setting state of the span frequency.
- [Parameter] <real> = Sweep time (s/ms/μs/ns)
  Zero span: 1 μsec - 6000 sec
  Frequency span: 10 msec - 2000 sec
- [Query reply] NR3 (Real value: Unit s)
- [Example] Call ibwrt (analyzer%, ":SWE:TIME 200MS")
- [Relevant commands] [:SENSe<ch>]:SWEep<screen>:TIME:AUTO

5.4.31 [:SENSe<ch>]:SWEep<screen>:TIME:AUTO

- [Command syntax] [:SENSe<ch>]:SWEep<screen>:TIME:AUTO < bool >
  [:SENSe<ch>]:SWEep<screen>:TIME:AUTO?
- [Function description] Selects the sweep time setting mode

  This command specifies the sweep time setting mode.
- [Parameter] < bool > = { OFF | ON }
  OFF: Sweep time MANUAL setting mode
  ON: Sweep time AUTO setting mode
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":SWE:TIME:AUTO OFF")
- [Relevant commands] [:SENSe<ch>]:SWEep<screen>:TIME

5.4.32 [:SENSe<ch>]:SWEep<screen>:WINDow

- [Command syntax] [:SENSe<ch>]:SWEep<screen>:WINDow <bool>
  [:SENSe<ch>]:SWEep<screen>:WINDow?
- [Function description] Sets the window sweep to ON or OFF

  This command sets the window sweep to ON or OFF.
  If ON is selected, within the range specified in the measuring window is swept. If OFF is selected, the whole span range is swept.
- [Parameter] < bool > = { OFF | ON }
  OFF: Window sweep OFF
  ON: Window sweep ON
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":SWE:WIND OFF")
- [Relevant commands] :CALCulate<ch>:WINDow<screen>:POSition
  :CALCulate<ch>:WINDow<screen>:WIDTh

5.4.33 [:SENSe<ch>]:SWEep:GATE

- [Command syntax] [:SENSe<ch>]:SWEep:GATE <bool>
  [:SENSe<ch>]:SWEep:GATE?
- [Function description] Sets the gated sweep to ON or OFF

  This command sets the gated sweep to ON or OFF.

  If ON is selected, the sweep is performed according to the conditions that have been set (gate position, gate width). If OFF is selected, the gated sweep is canceled.
- [Parameter] < bool > = { OFF | ON }
  OFF: Gated sweep OFF
  ON: Gated sweep ON
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":SWE:GATE OFF")
- [Relevant commands] [:SENSe<ch>]:SWEep:GATE:DELay
  [:SENSe<ch>]:SWEep:GATE:WIDTh
  [:SENSe<ch>]:SWEep:GATE:WIDTh:AUTO

5.4.34 [:SENSe<ch>]:SWEep:GATE:DELay

- [Command syntax] [:SENSe<ch>]:SWEep:GATE:DELay <real >
  [:SENSe<ch>]:SWEep:GATE:DELay?
- [Function description] Sets the gate signal position

  This command sets the delay time from a trigger point. The delay time is used as the gate position at the gated sweep.
- [Parameter] < real > = Gate position (s/ms/μs/ns)
  Setting range: 0 nsec - 1 sec
- [Query reply] NR3 (real number value in s)
- [Example] Call ibwrt (analyzer%, ":SWE:GATE:DEL 10MS")
- [Relevant commands] [:SENSe<ch>]:SWEep:GATE
  [:SENSe<ch>]:SWEep:GATE:WIDTh
  [:SENSe<ch>]:SWEep:GATE:WIDTh:AUTO

5.4.35 [:SENSe<ch>]:SWEep:GATE:WIDTh

- [Command syntax] [:SENSe<ch>]:SWEep:GATE:WIDTh <real>
  [:SENSe<ch>]:SWEep:GATE:WIDTh?
- [Function description] Sets the gate signal width

  This command sets the gate width at the gated sweep.
- [Parameter] < real > = Gate position (s/ms/μs/ns)
  Setting range: 50 μs - 1 sec
- [Query reply] NR3 (real number value in s)
- [Example] Call ibwrt (analyzer%, ":SWE:GATE:WIDT 500US")
- [Relevant commands] [:SENSe<ch>]:SWEep:GATE
  [:SENSe<ch>]:SWEep:GATE:DELay
  [:SENSe<ch>]:SWEep:GATE:WIDTh:AUTO

5.4.36 [:SENSe<ch>]:SWEep:GATE:WIDTh:AUTO

- [Command syntax] [:SENSe<ch>]:SWEep:GATE:WIDTh:AUTO < bool >
  [:SENSe<ch>]:SWEep:GATE:WIDTh:AUTO?
- [Function description] Switches the gate signal mode

  This command switches the gate signal mode setting between auto and manual when performing a gated sweep.

  In the Auto state, the gate width is automatically adjusted to correspond with the rectangular part of the gate signal resource. A signal, whose ON period is variable, is automatically adjusted.
- [Parameter] < bool > = { OFF | ON }
  OFF: Set to MANUAL state
  ON: Set to AUTO state
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":SWE:GATE:WIDT:AUTO OFF")
- [Relevant commands] [:SENSe<ch>]:SWEep:GATE
  [:SENSe<ch>]:SWEep:GATE:DELay
  [:SENSe<ch>]:SWEep:GATE:WIDTh

5.4.37 [:SENSe<ch>]:SWEep:GATE:SOURce

- [Command syntax] [:SENSe<ch>]:SWEep:GATE:SOURce <type >
  [:SENSe<ch>]:SWEep:GATE:SOURce?
- [Function description] Sets the gated sweep trigger

  This command sets the trigger source at the gated sweep.
- [Parameter] < type > = { IMMediate | IF | EXT1 | EXT2 | LINK }

  IMMediate: Free-run mode without trigger setting
  IF: IF trigger
  EXT1: EXT1 input signal trigger
  EXT2: EXT2 input signal trigger
  LINK: LINK trigger
- [Query reply] { IMM | IF | EXT1 | EXT2 | LINK }
- [Example] Call ibwrt (analyzer%, ":SWE:GATE:SOUR IF")
- [Relevant commands] [:SENSe<ch>]:SWEep:GATE:SLOPe
  [:SENSe<ch>]:SWEep:GATE:LEVel:EXTernal
  [:SENSe<ch>]:SWEep:GATE:LEVel:IF

5.4.38 [:SENSe<ch>]:SWEep:GATE:SLOPe

- [Command syntax] [:SENSe<ch>]:SWEep:GATE:SLOPe < type >
  [:SENSe<ch>]:SWEep:GATE:SLOPe?
- [Function description] Sets the trigger polarity of each trigger source

  This command sets the trigger polarity of the set trigger sources. The trigger polarity can be set for the following four types of trigger sources:
  - IF trigger
  - EXT1 trigger
  - EXT2 trigger
  - LINK trigger
- [Parameter] < type > = { NEGative | POSitive }

  NEGative: Falling or negative polarity
  POSitive: Rising or positive polarity
- [Query reply] { NEG | POS }
- [Example] Call ibwr (analyzer%, ":SWE:GATE:SLOP POS")
- [Relevant commands] [:SENSe<ch>]:SWEep:GATE:SOURce

5.4.39 [:SENSe<ch>]:SWEep:GATE:LEVel:EXTernal

- [Command syntax] [:SENSe<ch>]:SWEep:GATE:LEVel:EXTernal <real >
  [:SENSe<ch>]:SWEep:GATE:LEVel:EXTernal <real >?
- [Function description] Sets the trigger level when using an EXT2 (external input terminal 2) trigger

  This command sets the trigger level value when using an EXT2 trigger with a voltage value.
- [Parameter] < real > = Voltage value
  Setting range: 0 - 5 V
- [Query reply] NR3 (real number value in V)
- [Example] Call ibwr (analzyzer%, ":SWE:GATE:LEV:EXT 4.5")
- [Relevant commands] [:SENSe<ch>]:SWEep:GATE:SOURce
  [:SENSe<ch>]:SWEep:GATE:SLOPe

5.4.40 [:SENSe<ch>]:SWEep:GATE:LEVel:IF

- [Command syntax] [:SENSe<ch>]:SWEep:GATE:LEVel:IF <real >
  [:SENSe<ch>]:SWEep:GATE:LEVel:IF?
- [Function description] Sets the trigger level when using an IF trigger

  Sets the trigger level value when using an IF trigger in percentage terms.
- [Parameter] < real > = A percentage value for the IF input scale
  Setting range: 0 - 100 %
- [Query reply] NR3 (real number value in %)
- [Example] Call ibwrt (analyzer%, ":SWE:GATE:LEV:IF 55PCT")
- [Relevant commands] [:SENSe<ch>]:SWEep:GATE:SOURc
  [:SENSe<ch>]:SWEep:GATE:SLOPe

5.4.41 [:SENSe]:ROSCillator:SOURce:FREQuency

- [Command syntax] [:SENSe]:ROSCillator:SOURce:FREQuency < real >
  [:SENSe]:ROSCillator:SOURce:FREQuency?
- [Function description] Sets the frequency of the frequency reference

  This command not only sets the frequency of the frequency reference signal for synchronization of this instrument but also switches the mode to the external reference follow mode.
- [Parameter] <real> = Frequency (MHz/kHz/Hz)
  Setting range: 5 MHz - 20 MHz
  Setting resolution: 1 Hz
- [Query reply] NR3 (Real value: Unit Hz)
- [Example] '---------- Set the frequency of the external reference ----------
  Call ibwrt (analyzer%, ":ROSC:SOUR:AUTO OFF")
  Call ibwrt (analyzer%, ":ROSC:SOUR:FREQ 19MHZ")
- [Relevant commands] [:SENSe]:ROSCillator:SOURce:AUTO

5.4.42 [:SENSe]:ROSCillator:SOURce:AUTO

- [Command syntax] [:SENSe]:ROSCillator:SOURce:AUTO < bool >
  [:SENSe]:ROSCillator:SOURce:AUTO?
- [Function description] Switches the frequency reference standard (internal/external).

  This command switches the reference source mode of the frequency reference signal for synchronization of this instrument between two modes: the internal/external 10 MHz reference automatic switching mode and the external reference signal fixed mode.

  In the internal/external 10 MHz reference automatic switching mode, the reference frequency is fixed to 10 MHz. If an external 10 MHz reference source signal is input to the EXT REF IN connector on the rear side, it is detected automatically and the mode is switched to external.

  In the external reference signal fixed mode, it is possible to make the external reference frequency follow the specified value by setting any frequency between 5 MHz and 20 MHz by using the [:SENSe]:ROSCillator:SOURce:FREQuency command. In the external reference signal fixed mode, be sure to input the specified reference frequency signal from outside to the EXT REF IN connector on the rear side.
- [Parameter] < bool > = { OFF | ON }
  OFF: External reference signal fixed mode
  ON: Internal/external 10 MHz reference automatic switching mode
- [Query reply] { OFF | ON }
- [Example] '---------- Set the frequency of the external reference ----------
  Call ibwrt (analyzer%, ":ROSC:SOUR:AUTO OFF")
  Call ibwrt (analyzer%, ":ROSC:SOUR:FREQ 19.666MHZ")
- [Relevant commands] [:SENSe]:ROSCillator:SOURce:FREQuency

5.4.43 [:SENSe]:ROSCillator:SOURce:ADJust:COARse

- [Command syntax] [:SENSe]:ROSCillator:SOURce:ADJust:COARse < int >
  [:SENSe]:ROSCillator:SOURce:ADJust:COARse?
- [Function description] Coarse adjustment of the correction value for adjusting the internal 10 MHz frequency reference

  This command makes a coarse adjustment of the correction value for adjusting the 10 MHz frequency reference for synchronization of this instrument. A coarse adjustment of the frequency of the 10 MHz frequency reference is performed by this operation.
- [Parameter] <int> = Coarse adjustment value
  Setting range: 0 - 4095
- [Query reply] NR1 (Integer: No unit)
- [Example] '------ Adjust hardware of the internal reference signal & Save it -------
  Call ibwrt (analyzer%, ":ROSC:SOUR:ADJ:COAR 1024")
  Call ibwrt (analyzer%, ":ROSC:SOUR:ADJ:SAVE")
- [Relevant commands] [:SENSe]:ROSCillator:SOURce:ADJust:FINE
  [:SENSe]:ROSCillator:SOURce:ADJust:SAVE
  [:SENSe]:ROSCillator:SOURce:ADJust:DEFault

5.4.44 [:SENSe]:ROSCillator:SOURce:ADJust:FINE

- [Command syntax] [:SENSe]:ROSCillator:SOURce:ADJust:FINE < int >
  [:SENSe]:ROSCillator:SOURce:ADJust:FINE?
- [Function description] Fine adjustment of the correction value for adjusting the internal 10 MHz frequency reference

  This command makes a fine adjustment of the correction value for adjusting the 10 MHz frequency reference for synchronization of this instrument. A fine adjustment of the frequency of the 10 MHz frequency reference is performed by this operation. This command is enabled when the high stability reference source option is installed.
- [Parameter] <int> = Fine adjustment value
  Setting range: 0 - 4095
- [Query reply] NR1 (Integer: No unit)
- [Example] '------ Adjust hardware of the internal reference signal & Save it -------
  Call ibwrt (analyzer%, ":ROSC:SOUR:ADJ:FINE 1024")
  Call ibwrt (analyzer%, ":ROSC:SOUR:ADJ:SAVE")
- [Relevant commands] [:SENSe]:ROSCillator:SOURce:ADJust:COARse
  [:SENSe]:ROSCillator:SOURce:ADJust:SAVE
  [:SENSe]:ROSCillator:SOURce:ADJust:DEFault

5.4.45 [:SENSe]:ROSCillator:SOURce:ADJust:SAVE

- [Command syntax] [:SENSe]:ROSCillator:SOURce:ADJust:SAVE
- [Function description] Saves the correction value for adjusting the internal 10 MHz frequency reference

  This command saves the correction value for adjustment that was adjusted by the [:SENSe]:ROSCillator:SOURce:ADJust command in the internal backup memory.

  Even if the power is turned off after this value is saved, this correction value for adjustment is applied when the power is turned on next time.
- [Parameter] None
- [Query reply] None
- [Example] '------ Adjust hardware of the internal reference signal & Save it -------

  Call ibwrt (analyzer%, ":ROSC:SOUR:ADJ:COAR 65")

  Call ibwrt (analyzer%, ":ROSC:SOUR:ADJ:SAVE")
- [Relevant commands] [:SENSe]:ROSCillator:SOURce:ADJust:COARse

  [:SENSe]:ROSCillator:SOURce:ADJust:FINE

  [:SENSe]:ROSCillator:SOURce:ADJust:DEFault

5.4.46 [:SENSe]:ROSCillator:SOURce:ADJust:DEFault

- [Command syntax] [:SENSe]:ROSCillator:SOURce:ADJust:DEFault
- [Function description] Clears the correction value for adjusting the internal 10 MHz frequency reference

  This command returns the correction value for adjustment that was adjusted by the [:SENSe]:ROSCillator:SOURce:ADJust command to the factory setting. This command also returns the data in the internal backup memory to the data at factory shipment.
- [Parameter] None
- [Query reply] None
- [Example] '------ Adjust hardware of the internal reference signal & Save it -------

  Call ibwrt (analyzer%, ":ROSC:SOUR:ADJ:DEF")
- [Relevant commands] [:SENSe]:ROSCillator:SOURce:ADJust:COARse

  [:SENSe]:ROSCillator:SOURce:ADJust:FINE

  [:SENSe]:ROSCillator:SOURce:ADJust:DEFault

5.4.47 [:SENSe<ch>]:CORRection:CSET:STATe

- [Command syntax] [:SENSe<ch>]:CORRection:CSET:STATe < bool >
  [:SENSe<ch>]:CORRection:CSET:STATe?
- [Function description] Switches the RF input level correction function ON or OFF

  This command switches the input level correction function ON or OFF.

  The level correction data should be set for each of up to 400 frequency points by the [:SENSe<ch>]:CORRection:CSET:DATA command.
- [Parameter] < bool > = { OFF | ON }
  OFF: Turns off the level correction function.
  ON: Turns on the level correction function.
- [Query reply] { OFF | ON }
- [Example]

```
'------------------------ Prepare correction data -------------------------
Call ibwrt (analyzer%, ":CORR:CSET:DATA 500MHZ, 0.35DB")
Call ibwrt (analyzer%, ":CORR:CSET:DATA 610MHZ, 0.45DB")
Call ibwrt (analyzer%, ":CORR:CSET:DATA 700MHZ, 0.34DB")
'----------- Apply correction data ---------------------------------------
Call ibwrt (analyzer%, ":CORR:STAT ON")
```

- [Relevant commands] [:SENSe<ch>]:CORRection:CSET:DATA

5.4.48 [:SENSe<ch>]:CORRection:CSET:DATA

- [Command syntax] [:SENSe<ch>]:CORRection:CSET:DATA < real , real >
- [Function description] Enters the RF input level correction data

  This command enters the level correction data used in the input level correction function.

  Level correction data consists of the frequency position to be corrected and its correction level (dB). At the time of setting it, enter the frequency value and its correction value in a pair. As correction data, up to 400 points can be set.
- [Parameter] < real , real > = Frequency (GHz/MHz/kHz/Hz), level (dB)
  Delimit frequency data and correction level data by a comma (,).
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CORR:CSET:DATA 500MHZ, 0.35DB")
- [Relevant commands] [:SENSe<ch>]:CORRection:CSET:STATe
  [:SENSe<ch>]:CORRection:CSET:DELete

5.4.49 [:SENSe<ch>]:CORRection:CSET:DELete

- [Command syntax] [:SENSe<ch>]:CORRection:CSET:DELete
- [Function description] Clears all the RF input level correction data

  This command clears all the level correction data used in the input level correction function.

  Unless the correction data group set at this point is cleared before starting the input of level correction data with the [:SENSe<ch>]:CORRection :CSET:DATA command, the entered data is added to the previous correction data.

  When creating a new correction table, it is necessary to delete all the previous data with this command.
- [Parameter] None
- [Query reply] None
- [Example]

```
'----------------------- Entry the correction data -------------------------
Call ibwrt (analyzer%, ":CORR:CSET:DATA 500MHZ, 0.35DB")
Call ibwrt (analyzer%, ":CORR:CSET:DATA 610MHZ, 0.45DB")
Call ibwrt (analyzer%, ":CORR:CSET:DATA 700MHZ, 0.34DB")

'----------------------- Delete all correction data ------------------------
Call ibwrt (analyzer%,":CORR:CSET:DEL")
```

- [Relevant commands] [:SENSe<ch>]:CORRection:CSET:DATA

5.4.50 [:SENSe<ch>]:SWEep<screen>:COUNt

- [Command syntax] [:SENSe<ch>]:SWEep<screen>:COUNt < int >
  [:SENSe<ch>]:SWEep<screen>:COUNt?
- [Function description] Sets the sweep averaging count and MAX or MIN HOLD count

  This command sets the sweep averaging count and MAX or MIN HOLD count.

  After sweep is performed the number of times specified as the sweep count with this command, the AVERaging bit in the standard operation status register is set.
- [Parameter] <int> = Averaging count
- [Query reply] NR1 (Integer)
- [Example] Call ibwrt (analyzer%, ":SWE:COUN 10")
- [Relevant commands] :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}]:MODE

5.4.51 [:SENSe<ch>]:AANalog:SAMPle:COUNt

- [Command syntax] [:SENSe<ch>]:AANalog:SAMPle:COUNt < int >
  [:SENSe<ch>]:AANalog:SAMPle:COUNt?
- [Function description] Sets the quasi analog function and sampling count

  This command sets the sampling count for the vertical axis at the time of quasi analog display.

  After the display as the same number of times as the sampling count specified by this command is complete, the AVERaging bit in the standard operation status register is set.
- [Parameter] <int> = Sampling count
  Setting range: 2 - 32
- [Query reply] NR1 (Integer)
- [Example] Call ibwrt (analyzer%, ":AAN:SAMP:COUN 20")
- [Relevant commands] :DISPlay<ch>[:WINDow<screen>]:TRACe:AANalog:STATe

5.4.52 [:SENSe<ch>]:CPOWer<screen>:AVERage:COUNt

- [Command syntax] [:SENSe<ch>]:CPOWer<screen>:AVERage:COUNt < int >
  [:SENSe<ch>]:CPOWer<screen>:AVERage:COUNt?
- [Function description] Sets the averaging count in channel power measurement

  This command sets the averaging count at channel power measurement.

  After sweep is repeated the number of times specified here as the averaging count and the operation of the measured value is completed, the AVERaging and MEASuring bits in the standard operation status register are set.
- [Parameter] <int> = Averaging count
- [Query reply] NR1 (Integer)
- [Example] Call ibwrt (analyzer%, ":CPOW:AVER:COUN 5")
- [Relevant commands] [:SENSe<ch>]:CPOWer<screen>:AVERage[:STATe]
  [:SENSe<ch>]:CPOWer<screen>:AVERage:MODE

5.4.53 [:SENSe<ch>]:CPOWer<screen>:AVERage[:STATe]

- [Command syntax] [:SENSe<ch>]:CPOWer<screen>:AVERage[:STATe] < bool >
  [:SENSe<ch>]:CPOWer<screen>:AVERage[:STATe]?
- [Function description] Sets the averaging operation mode to ON or OFF in channel power measurement

  This command sets the averaging operation mode to ON or OFF in channel power measurement. After the averaging operation mode is switched to ON or OFF, the channel power operation that was performed up to then is reset.
- [Parameter] < bool > = { OFF | ON }
  OFF: Ordinary operation mode
  ON: Averaging operation mode
- [Query reply] { OFF | ON }
- [Example]

```
'----------- Set the average count to 10 times -------------------------
Call ibwrt (analyzer%, ":CPOW:AVER:COUN 10")
------------ Set the Power measuremnt mode to average mode -----
Call ibwrt (analyzer%, ":CPOW:AVER:STAT ON")
```

- [Relevant commands] [:SENSe<ch>]:CPOWer<screen>:AVERage:COUNt
  [:SENSe<ch>]:CPOWer<screen>:AVERage:MODE

5.4.54 [:SENSe<ch>]:CPOWer<screen>:AVERage:MODE

- [Command syntax] [:SENSe<ch>]:CPOWer<screen>:AVERage:MODE < type >
  [:SENSe<ch>]:CPOWer<screen>:AVERage:MODE?
- [Function description] Specifies the operation type in averaging operation mode in channel power measurement

  This command specifies the operation type after the set averaging count is reached in channel power measurement.

  There are two operation types in terms of the operation after the set averaging count is reached. In the Continuous type, averaging operation is continued by using the moving average. In the Repeat type, the processing count is reset and averaging operation is performed repeatedly the number of times specified as the averaging count.
- [Parameter] < type > = { CONTinuous | REPeat }

  CONTinuous: Moving average after the averaging count is reached.

  REPeat: Averaging operation is performed repeatedly.
- [Query reply] { CONT | REP }
- [Example] Call ibwrt (analyzer%, ":CPOW:AVER:MODE CONT")
- [Relevant commands] [:SENSe<ch>]:CPOWer<screen>:AVERage:COUNt
  [:SENSe<ch>]:CPOWer<screen>:AVERage[:STATe]

5.4.55 [:SENSe<ch>]:CPOWer<screen>:WINDow

- [Command syntax]
  [:SENSe<ch>]:CPOWer<screen>:WINDow < bool >
  [:SENSe<ch>]:CPOWer<screen>:WINDow?
- [Function description]
  Sets the measurement window display to ON or OFF in channel power measurement

  This command switches the measurement window display indicating the measurement band in channel power measurement between ON and OFF.

  When the measurement window is displayed (ON):

  The measurement frequency range in power measurement is within the displayed window.

  When the measurement window is not displayed (OFF):

  The measurement frequency range in power measurement is the span frequency set on the screen.
- [Parameter]
  < bool > = { OFF | ON }
  OFF: Window display is off.
  ON: Window display is on.
- [Query reply]
  { OFF | ON }
- [Example]
  Call ibwrt (analyzer%, ":CPOW:WIND ON")
- [Relevant commands]
  [:SENSe<ch>]:CPOWer<screen>:WINDow:POSition
  [:SENSe<ch>]:CPOWer<screen>:WINDow:WIDTh

5.4.56 [:SENSe<ch>]:CPOWer<screen>:WINDow:POSition

- [Command syntax]
  [:SENSe<ch>]:CPOWer<screen>:WINDow:POSition < real >
  [:SENSe<ch>]:CPOWer<screen>:WINDow:POSition?
- [Function description]
  Specifies the measurement window display position in channel power measurement

  This command specifies the frequency position of the measurement window used in channel power measurement. The specified frequency becomes the center frequency in the measurement window.

  When the measurement window is displayed (ON):

  The target frequency range in power measurement is within the displayed window.

  When the measurement window is not displayed (OFF):

  The target frequency range in power measurement is the span frequency set on the screen.
- [Parameter]
  <real> = Frequency (GHz/MHz/kHz/Hz)
- [Query reply]
  NR3 (Real value: Unit Hz)
- [Example]
  Call ibwrt (analyzer%, ":FREQ:CENT 1GHZ")
  Call ibwrt (analyzer%, ":FREQ:SPAN 200MHZ")
  '--------- Set the measurement band width for power meas. --------
  Call ibwrt (analyzer%, ":CPOW:WIND:POS 1GHZ")
  Call ibwrt (analyzer%, ":CPOW:WIND:WIDT 40MHZ")
  Call ibwrt (analyzer%, ":CPOW:WIND ON")
- [Relevant commands]
  [:SENSe<ch>]:CPOWer<screen>:WINDow
  [:SENSe<ch>]:CPOWer<screen>:WINDow:WIDTh

5.4.57 [:SENSe<ch>]:CPOWer<screen>:WINDow:WIDTh

- [Command syntax] [:SENSe<ch>]:CPOWer<screen>:WINDow:WIDTh < real >
  [:SENSe<ch>]:CPOWer<screen>:WINDow:WIDTh ?
- [Function description] Specifies the measurement window display width in channel power measurement

  This command specifies the frequency width in the measurement window used in channel power measurement.

  The power in the frequency section specified here is calculated.

  With the frequency position specified with the window display position specification command [:SENSe<ch>]:CPOWer<screen>:WINDow:POSition as the center, the window for the frequency width specified with this command is displayed.
- [Parameter] <real> = Frequency (GHz/MHz/kHz/Hz)
- [Query reply] NR3 (Real value: Unit Hz)
- [Example] Call ibwrt (analyzer%, ":CPOW:WIND:POS 100MHZ")
  Call ibwrt (analyzer%, ":CPOW:WIND:WIDT 10MHZ")
  Call ibwrt (analyzer%, ":CPOW:WIND ON")
- [Relevant commands] [:SENSe<ch>]:CPOWer<screen>:WINDow
  [:SENSe<ch>]:CPOWer<screen>:WINDow:POSition

5.4.58 [:SENSe<ch>]:CPOWer<screen>:DATA:MODE

- [Command syntax] [:SENSe<ch>]:CPOWer<screen>:DATA:MODE < type >
  [:SENSe<ch>]:CPOWer<screen>:DATA:MODE?
- [Function description] Specifies the measurement parameter setting mode in channel power measurement

  This command specifies the setting modes of the parameters used in channel power measurement. There are two modes as shown below.

  Default: Measures in the preset parameter state.

  Manual: Measures in the parameter state before entering the channel power measurement mode.

  For more information on the measurement parameters used here, refer to the function explanation of the command that saves the parameter values used in default mode [:SENSe<ch>]:CPOWer<screen>:DATA:SAVE.
- [Parameter] < type > = { DEFault | MANual }

  DEFault: Performs a measurement by using the parameter values saved by the [:SENSe<ch>]:CPOWer<screen>:DATA:SAVE command.

  MANual: Performs a measurement by using the setting value before entering the channel power measurement mode.
- [Query reply] { DEF | MAN }
- [Example] Call ibwrt (analyzer%, ":CPOW:DATA:MODE MAN")
- [Relevant commands] [:SENSe<ch>]:CPOWer<screen>:DATA:SAVE

5.4.59 [:SENSe<ch>]:CPOWer<screen>:DATA:SAVE

- [Command syntax] [:SENSe<ch>]:CPOWer<screen>:DATA:SAVE
- [Function description] Saves the measurement parameters in channel power measurement

  This command saves the current values of the parameters used in channel power measurement as the values used in default mode.

  The parameters saved by this command are shown below.
  - ON/OFF of the measurement window and window position/width
  - RBW value
  - VBW value
  - Sweep time
  - Span frequency
  - Averaging ON/OFF and averaging count
  - Averaging repeat mode
  - Trace status
  - Trace detector
  - Trigger
  - Gated sweep
- [Parameter] None
- [Query reply] None
- [Example]

```
'-------- Set each parameter for power measurement --------
Call ibwrt (analyzer%, ":CPOW:AVER:MODE CONT")
Call ibwrt (analyzer%, ":CPOW:AVER:COUN 20")
Call ibwrt (analyzer%, ":CPOW:AVER ON")

'-------- Save the parameter for channel power meas ---------
Call ibwrt (analyzer%, ":CPOW:DATA:SAVE")
```

- [Relevant commands] [:SENSe<ch>]:CPOWer<screen>:DATA:MODE

5.4.60 [:SENSe<ch>]:APOWer<screen>:AVERage:COUNt

- [Command syntax] [:SENSe<ch>]:APOWer<screen>:AVERage:COUNt < int >
  [:SENSe<ch>]:APOWer<screen>:AVERage:COUNt?
- [Function description] Sets the averaging count in average power measurement

  This command sets the averaging count in average power measurement.

  After sweep is repeated the number of times specified here as the averaging count and the operation of the measured value is completed, the AVERaging and MEASuring bits in the standard operation status register are set.
- [Parameter] <int> = Averaging count
- [Query reply] NR1 (Integer)
- [Example] Call ibwrt (analyzer%, ":APOW:AVER:COUN 5")
- [Relevant commands] [:SENSe<ch>]:APOWer<screen>:AVERage[:STATe]
  [:SENSe<ch>]:APOWer<screen>:AVERage:MODE

5.4.61 [:SENSe<ch>]:APOWer<screen>:AVERage[:STATe]

- [Command syntax] [:SENSe<ch>]:APOWer<screen>:AVERage[:STATe] < bool >
  [:SENSe<ch>]:APOWer<screen>:AVERage[:STATe]?
- [Function description] Sets the averaging operation mode to ON or OFF in average power measurement

  This command sets the averaging operation mode to ON or OFF in average power measurement. After the averaging operation mode is switched to ON or OFF, the average power operation that was performed up to then is reset.
- [Parameter] < bool > = { OFF | ON }
  ON: Averaging operation mode
  OFF: Ordinary operation mode
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":APOW:AVER:COUN 15")
  Call ibwrt (analyzer%, ":APOW:AVER:STAT ON")
- [Relevant commands] [:SENSe<ch>]:APOWer<screen>:AVERage:COUNt
  [:SENSe<ch>]:APOWer<screen>:AVERage:MODE

5.4.62 [:SENSe<ch>]:APOWer<screen>:AVERage:MODE

• [Command syntax]	[:SENSe<ch>]:APOWer<screen>:AVERage:MODE < type > [:SENSe<ch>]:APOWer<screen>:AVERage:MODE?
• [Function description]	Specifies the operation type in averaging operation mode in average power measurement This command specifies the operation type after the set averaging count is reached in average power measurement. There are two operation types in terms of the operation after the set averaging count is reached. In the Continuous type, averaging operation is continued by using the moving average. In the Repeat type, the processing count is reset and averaging operation is performed repeatedly the number of times specified as the averaging count.
• [Parameter]	< type > = { CONTinuous | REPeat } CONTinuous: Moving average after the averaging count is reached. REPeat: Averaging operation is performed repeatedly.
• [Query reply]	{ CONT | REP }
• [Example]	Call ibwrt (analyzer%. ":APOW:AVER:MODE CONT")
• [Relevant commands]	[:SENSe<ch>]:APOWer<screen>:AVERage:COUNt [:SENSe<ch>]:APOWer<screen>:AVERage[:STATe]

5.4.63 [:SENSe<ch>]:APOWer<screen>:WINDow

- [Command syntax] [:SENSe<ch>]:APOWer<screen>:WINDow < bool >
  [:SENSe<ch>]:APOWer<screen>:WINDow
- [Function description] Sets the measurement window display to ON or OFF in average power measurement

  This command sets the measurement window display that indicates the frequency width measured in average power measurement to ON or OFF.

  When the measurement window is displayed (ON):

  The measurement frequency range in power measurement is within the displayed window.

  When the measurement window is not displayed (OFF):

  The measurement frequency range in power measurement is all of the band set on the screen.
- [Parameter] < bool > = { OFF | ON }

  OFF: Window display is off.

  ON: Window display is on.
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":APOW:WIND ON")
- [Relevant commands] [:SENSe<ch>]:APOWer<screen>:WINDow:POSition
  [:SENSe<ch>]:APOWer<screen>:WINDow:WIDTh

5.4.64 [:SENSe<ch>]:APOWer<screen>:WINDow:POSition

- [Command syntax]

  [:SENSe<ch>]:APOWer<screen>:WINDow:POSition < real >

  [:SENSe<ch>]:APOWer<screen>:WINDow:POSition?

- [Function description]

  Specifies the measurement window display position in average power measurement

  This command specifies the frequency position (when the frequency display is used) or the time position (when the time display is used) in the measurement window used in average power measurement.

  The specified frequency is the frequency at the center of the measurement window, while the time specified at the time of time display is the offset time from the left of the screen and at the left end of the displayed window.

  When the measurement window is displayed (ON):

  The measurement frequency (time) range in power measurement is within the displayed window.

  When the measurement window is not displayed (OFF):

  The target frequency (time) range in power measurement is the span frequency (or time) set in the screen.

- [Parameter]

  <real> = Frequency (GHz/MHz/kHz/Hz)

  or

  <real> = Time (s/ms/μs/ns)

- [Query reply]

  NR3 (Real value)

  Frequency span in Hz

  Zero span in s

- [Example]

```
'----------- Set the span mode to zero span -----------------
Call ibwrt (analyzer%, ":FREQ:SPAN 0HZ")
Call ibwrt (analyzer%, ":SWE:TIME 1S")
'---------- Set the meas. window's position ----------------
Call ibwrt (analyzer%, ":APOW:WIND:POS 400MS")
```

- [Relevant commands]

  [:SENSe<ch>]:APOWer<screen>:WINDow

  [:SENSe<ch>]:APOWer<screen>:WINDow:WIDTh

5.4.65 [:SENSe<ch>]:APOWer<screen>:WINDow:WIDTh

- [Command syntax] [:SENSe<ch>]:APOWer<screen>:WINDow:WIDTh < real >
  [:SENSe<ch>]:APOWer<screen>:WINDow:WIDTh?
- [Function description] Specifies the measurement window display width in average power measurement

  This command specifies the width of the measurement window used in average power measurement. The power in the measurement section specified here is calculated.

  The unit of the window width set with this command varies depending on the span state when making a setting.

  For the zero span, the window for the time width specified with this command is displayed with the time position specified with the window display position specification command as the starting point.

  For the frequency span, the window for the frequency width specified with this command is displayed with the frequency position specified with the window display position specification command [:SENSe<ch>]:APOWer<screen>:WINDow:POSition as the center.
- [Parameter] <real> = Frequency (GHz/MHz/kHz/Hz)
  or
  <real> = Time (s/ms/μs/ns)
- [Query reply] NR3 (Real value)
  Frequency span in Hz
  Zero span in s
- [Example]
  ```
  '---------- Set the span mode to zero span ----------------
  Call ibwrt (analyzer%, ":FREQ:SPAN 0HZ")
  Call ibwrt (analyzer%, ":SWE:TIME 1S")
  '--------- Set the meas. window's position ---------------
  Call ibwrt (analyzer%, ":APOW:WIND:POS 400MS")
  Call ibwrt (analyzer%, ":APOW:WIND:WIDT 200MS")
  ```
- [Relevant commands] [:SENSe<ch>]:APOWer<screen>:WINDow
  [:SENSe<ch>]:APOWer<screen>:WINDow:POSition

5.4.66 [:SENSe<ch>]:APOWer<screen>:DATA:MODE

- [Command syntax] [:SENSe<ch>]:APOWer<screen>:DATA:MODE < type >
  [:SENSe<ch>]:APOWer<screen>:DATA:MODE?
- [Function description] Specifies the measurement parameter setting mode in average power measurement

  This command specifies the setting modes of the parameters used in average power measurement. There are two modes as shown below.

  Default: Measures in the preset parameter state.

  Manual: Measures in the parameter state before entering the average power measurement mode.

  For more information on the measurement parameters used here, refer to the function explanation of the command that saves the parameter values used in default mode [:SENSe<ch>]:APOWer<screen>:DATA:SAVE.
- [Parameter] < type > = { DEFault | MANual }

  DEFault: Performs a measurement by using the parameter values saved by the [:SENSe<ch>]:APOWer<screen>:DATA:SAVE command.

  MANual: Performs a measurement by using the setting value before entering the average power measurement mode.
- [Query reply] { DEF | MAN }
- [Example] Call ibwrt (analyzer%, ":APOW:DATA:MODE DEF")
- [Relevant commands] [:SENSe<ch>]:APOWer<screen>:DATA:SAVE

5.4.67 [:SENSe<ch>]:APOWer<screen>:DATA:SAVE

- [Command syntax] [:SENSe<ch>]:APOWer<screen>:DATA:SAVE
- [Function description] Saves the measurement parameters in average power measurement

  This command saves the current values of the parameters used in average power measurement as the values used in default mode.

  The parameters saved by this command are shown below.
  - ON/OFF of the measurement window and window position/width
  - RBW value
  - VBW value
  - Sweep time
  - Span frequency
  - Averaging ON/OFF and averaging count
  - Averaging repeat mode
  - Trace status
  - Trace detector
  - Trigger and trigger delay value
  - Gated sweep
- [Parameter] None
- [Query reply] None
- [Example]

```
'-------- Set each parameter for average power meas. --------
Call ibwrt (analyzer%, ":APOW:AVER:MODE CONT")
Call ibwrt (analyzer%, ":APOW:AVER:COUN 20")
Call ibwrt (analyzer%, ":APOW:AVER ON")

'------------- Save it for average power meas ------------------
Call ibwrt (analyzer%, ":APOW:DATA:SAVE")
```

- [Relevant commands] [:SENSe<ch>]:APOWer<screen>:DATA:MODE

5.4.68 [:SENSe<ch>]:OBW<screen>:AVERage:COUNt

- [Command syntax] [:SENSe<ch>]:OBW<screen>:AVERage:COUNt < int >
  [:SENSe<ch>]:OBW<screen>:AVERage:COUNt?
- [Function description] Sets the averaging count in OBW measurement

  This command sets the averaging count in OBW measurement.

  After sweep is repeated the number of times specified here as the averaging count and the operation of the measured value is completed, the AVERaging and MEASuring bits in the standard operation status register are set.
- [Parameter] <int> = Averaging count
- [Query reply] NR1 (Integer)
- [Example] Call ibwrt (analyzer%, ":OBW:AVER:COUN 5")
- [Relevant commands] [:SENSe<ch>]:OBW<screen>:AVERage[:STATe]
  [:SENSe<ch>]:OBW<screen>:AVERage:MODE

5.4.69 [:SENSe<ch>]:OBW<screen>:AVERage[:STATe]

- [Command syntax] [:SENSe<ch>]:OBW<screen>:AVERage[:STATe] < bool >
  [:SENSe<ch>]:OBW<screen>:AVERage[:STATe]?
- [Function description] Sets the averaging operation mode to ON or OFF in OBW measurement

  This command sets the averaging operation mode to ON or OFF in OBW measurement.

  After the averaging operation mode is switched to ON or OFF, the OBW operation that was performed up to then is reset.
- [Parameter] < bool > = { OFF | ON }
  OFF: Ordinary operation mode
  ON: Averaging operation mode
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":OBW:AVER:STAT ON")
- [Relevant commands] [:SENSe<ch>]:OBW<screen>:AVERage:COUNt
  [:SENSe<ch>]:OBW<screen>:AVERage:MODE

5.4.70 [:SENSe<ch>]:OBW<screen>:AVERage:MODE

- [Command syntax] [:SENSe<ch>]:OBW<screen>:AVERage:MODE < type >
  [:SENSe<ch>]:OBW<screen>:AVERage:MODE?
- [Function description] Specifies the operation type in averaging operation mode in OBW measurement

  This command specifies the operation type after the set averaging count is reached in OBW measurement.

  There are two operation types in terms of the operation after the set averaging count is reached.In the Continuous type, averaging operation is continued by using the moving average.In the Repeat type, the processing count is reset and averaging operation is performed repeatedly the number of times specified as the averaging count.
- [Parameter] < type > = { CONTinuous | REPeat }

  CONTinous: Moving average after the averaging count is reached.

  REPeat: Averaging operation is performed repeatedly.
- [Query reply] { CONT | REP }
- [Example] Call ibwrt (analyzer%, ":OBW:AVER:MODE CONT")
- [Relevant command] [:SENSe<ch>]:OBW<screen>:AVERage:COUNt
  [:SENSe<ch>]:OBW<screen>:AVERage[:STATe]

5.4.71 [:SENSe<ch>]:OBW<screen>:PERCent

- [Command syntax] [:SENSe<ch>]:OBW<screen>:PERCent < real >
  [:SENSe<ch>]:OBW<screen>:PERCent?
- [Function description] Specifies the OBW% value in OBW measurement

  Specify the OBW% value used in OBW measurement.
- [Parameter] <real> = OBW% value

  Setting range: 0.1% - 99.9%
- [Query reply] NR3 (Real value: Unit %)
- [Example] Call ibwrt (analyzer%, ":OBW:PERC 99.1")
- [Relevant commands]

5.4.72 [:SENSe<ch>]:OBW<screen>:DATA:MODE

- [Command syntax] [:SENSe<ch>]:OBW<screen>:DATA:MODE < type >
  [:SENSe<ch>]:OBW<screen>:DATA:MODE?
- [Function description] Specifies the measurement parameter setting mode in OBW measurement

  This command specifies the setting modes of the parameters used in OBW measurement.

  There are two modes as shown below.

  Default: Measures in the preset parameter state.

  Manual: Measures in the parameter state before entering the OBW measurement mode.

  For more information on the measurement parameters used here, refer to the function explanation of the command [:SENSe<ch>]:OBW<screen>:DATA:SAVE, which saves the parameter values used in default mode.
- [Parameter] < type > = { DEFault | MANual }

  DEFault: Performs a measurement by using the parameter values saved by the [:SENSe<ch>]:OBW<screen>:DATA:SAVE command.

  MANual: Performs a measurement by using the setting value before entering the OBW measurement mode.
- [Query reply] { DEF | MAN }
- [Example] Call ibwrt (analyzer%, ":OBW:DATA:MODE DEF")
- [Relevant commands] [:SENSe<ch>]:OBW<screen>:DATA:SAVE

5.4.73 [:SENSe<ch>]:OBW<screen>:DATA:SAVE

- [Command syntax] [:SENSe<ch>]:OBW<screen>:DATA:SAVE
- [Function description] Saves the measurement parameters in OBW measurement

  This command saves the current values of the parameters used in OBW measurement as the values used in default mode.

  The parameters saved by this command are shown below.
  - OBW% value
  - RBW value
  - VBW value
  - Sweep time
  - Span frequency
  - Averaging ON/OFF and averaging count
  - Averaging repeat mode
  - Trace status
  - Trace detector
  - Trigger
  - Gated sweep
- [Parameter] None
- [Query reply] None
- [Example]

```
'------------- Set OBW meas. parameters --------------
Call ibwrt (analyzer%, ":OBW:PERC 99.9")
Call ibwrt (analyzer%, ":OBW:AVER:COUN 10")
Call ibwrt (analyzer%, ":OBW:AVER:MODE REP")
Call ibwrt (analyzer%, ":OBW:AVER:STAT ON")

'----- Save OBW meas. parameters as Default --------
Call ibwrt (analyzer%, ":OBW:DATA:SAVE")
```

- [Relevant commands] [:SENSe<ch>]:OBW<screen>:DATA:MODE

5.4.74 [:SENSe<ch>]:ACP:AVERage:COUNt

- [Command syntax] [:SENSe<ch>]:ACP:AVERage:COUNt < int >
  [:SENSe<ch>]:ACP:AVERage:COUNt?
- [Function description] Sets the averaging count in ACP measurement

  This command sets the averaging count in the adjacent channel leakage power (ACP) measurement.

  After sweep is repeated the number of times specified here as the averaging count and the operation of the measured value is completed, the AVERaging and MEASuring bits in the standard operation status register are set.
- [Parameter] <int> = Averaging count
- [Query reply] NR1 (Integer)
- [Example] Call ibwrt (analyzer%, ":ACP:AVER:COUN 8")
- [Relevant commands] [:SENSe<ch>]:ACP:AVERage[:STATe]
  [:SENSe<ch>]:ACP:AVERage:MODE

5.4.75 [:SENSe<ch>]:ACP:AVERage[:STATe]

- [Command syntax] [:SENSe<ch>]:ACP:AVERage[:STATe] < bool >
  [:SENSe<ch>]:ACP:AVERage[:STATe]?
- [Function description] Sets the averaging operation mode to ON or OFF in ACP measurement

  This command sets the averaging operation mode to ON or OFF in the adjacent channel leakage power (ACP) measurement.

  After the averaging operation mode is switched to ON or OFF, the ACP operation that was performed up to then is reset.
- [Parameter] < bool > = { OFF | ON }
  OFF: Ordinary operation mode
  ON: Averaging operation mode
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":ACP:AVER ON")
- [Relevant commands] [:SENSe<ch>]:ACP:AVERage:COUNt
  [:SENSe<ch>]:ACP:AVERage:MODE

5.4.76 [:SENSe<ch>]:ACP:AVERage:MODE

- [Command syntax] [:SENSe<ch>]:ACP:AVERage:MODE < type >
  [:SENSe<ch>]:ACP:AVERage:MODE?
- [Function description] Specifies the operation type in averaging operation mode in ACP measurement

  This command specifies the operation type after the set averaging count is reached in adjacent channel leakage power (ACP) measurement.

  There are two operation types in terms of the operation after the set averaging count is reached. In the Continuous type, averaging operation is continued by using the moving average. In the Repeat type, the processing count is reset and averaging operation is performed repeatedly the number of times specified as the averaging count.
- [Parameter] < type > = { CONTinuous | REPeat }

  CONTinuous: Moving average after the averaging count is reached.

  REPeat: Averaging operation is performed repeatedly.
- [Query reply] { CONT | REP }
- [Example] Call ibwrt (analyzer%, ":ACP:AVER:MODE CONT")
- [Relevant commands] [:SENSe<ch>]:ACP:AVERage:COUNt
  [:SENSe<ch>]:ACP:AVERage[:STATe]

5.4.77 [:SENSe<ch>]:ACP:DATA:MODE

- [Command syntax] [:SENSe<ch>]:ACP:DATA:MODE < type >
  [:SENSe<ch>]:ACP:DATA:MODE?
- [Function description] Specifies the measurement parameter setting mode in ACP measurement

  This command specifies the setting modes of the parameters used in adjacent channel leakage power (ACP) measurement.

  There are two modes as shown below.

  Default: Measures in the preset parameter state.

  Manual: Measures in the parameter state before entering the ACP measurement mode.

  For more information on the measurement parameters used here, refer to the function explanation of the command [:SENSe<ch>]:ACP:DATA:SAVE, which saves the parameter values used in default mode.
- [Parameter] < type > = { DEFault | MANual }

  DEFault: Performs a measurement by using the parameter values saved by the [:SENSe<ch>]:ACP:DATA:SAVE command.

  MANual: Performs a measurement by using the setting value before entering the ACP measurement mode.
- [Query reply] { DEF | MAN }
- [Example] Call ibwrt (analyzer%, ":ACP:DATA:MODE DEF")
- [Relevant commands] [:SENSe<ch>]:ACP:DATA:SAVE

5.4.78 [:SENSe<ch>]:ACP:DATA:SAVE

- [Command syntax] [:SENSe<ch>]:ACP:DATA:SAVE
- [Function description] Saves the measurement parameters in ACP measurement.

  This command saves the current values of the parameters used in adjacent channel leakage power (ACP) measurement as the values used in default mode.

  The parameters saved by this command are shown below.
  - All parameters set in the CS/BS Setup
  - All parameters set in the Root Nyquist Filter Setup
  - RBW value
  - VBW value
  - Sweep time
  - Span frequency
  - Averaging ON/OFF and averaging count
  - Averaging repeat mode
  - Noise correction ON/OFF
  - Trace status
  - Trace detector
  - Trigger
  - Gated sweep
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":ACP:DATA:SAVE")
- [Relevant command] [:SENSe<ch>]:ACP:DATA:MODE

5.4.79 [:SENSe<ch>]:ACP:CBWidth

- [Command syntax] [:SENSe<ch>]:ACP:CBWidth < real >
  [:SENSe<ch>]:ACP:CBWidth?
- [Function description] Sets the carrier bandwidth that becomes the target of the reference power operation in ACP measurement

  This command specifies the frequency band for the operation of the carrier power that becomes the reference power in adjacent channel leakage power (ACP).
- [Parameter] <real> = Frequency (GHz/MHz/kHz/Hz)
- [Query reply] NR3 (Real value: Unit Hz)
- [Example] Call ibwrt (analyzer%, ":ACP:CBW 3.84MHZ")
- [Relevant commands]

5.4.80 [:SENSe<ch>]:ACP:CSBW:DATA

- [Command syntax] [:SENSe<ch>]:ACP:CSBW:DATA < real >,< real >
- [Function description] Sets the adjacent channel position and adjacent channel band in ACP measurement

  This command sets the adjacent channel position (CS) and measurement band (BW) that become the targets of ACP measurement.

  The adjacent channel position is set as the offset frequency from the carrier frequency. In ACP measurement, for one specified offset frequency, the relative power values of the adjacent channels in the negative and positive frequency offset positions from the carrier frequency are measured.

  Up to five adjacent channel positions can be set with this offset frequency.

  This command adds data after the last pair of the adjacent channel position and the measurement band that have already been set.
- [Parameter] < real >,< real > = Offset frequency, measurement band (GHz/MHz/kHz/Hz)
- [Query reply] NR3, NR3 (real value: Unit Hz: real value: Unit Hz)
- [Example] Call ibwr (analyzer%, ":ACP:CSBW:DATA 5MHZ, 3.84MHZ")
  Call ibwt (analyzer%, ":ACP:CSBW:DATA10MHZ, 3.84MHZ")
- [Relevant commands] [:SENSe<ch>]:ACP:CSBW:DATA:DELete

5.4.81 [:SENSe<ch>]:ACP:CSBW:DATA:DELete

- [Command syntax] [:SENSe<ch>]:ACP:CSBW:DATA:DELete
- [Function description] Initializes the table of the adjacent channel position and band data in ACP measurement

  This command clears all the adjacent channel positions and measurement bands set with the [:SENSe<ch>]:ACP:CSBW:DATA command.

  To set adjacent channel positions and measurement bands again, send required adjacent channel data one by one with the [:SENSe<ch>]:ACP:CSBW:DATA command after sending this command.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":ACP:CSBW:DATA:DEL")
- [Relevant commands] [:SENSe<ch>]:ACP:CSBW:DATA

5.4.82 [:SENSe<ch>]:ACP:RNYQuist

- [Command syntax] [:SENSe<ch>]:ACP:RNYQuist < bool >
  [:SENSe<ch>]:ACP:RNYQuist?
- [Function description] Sets the Root Nyquist filter operation mode to ON or OFF in ACP measurement

  This command sets whether Root Nyquist filter operation is used for the power calculation in ACP measurement.

  When the parameter is set to ON, Root Nyquist filter operation is performed in all power calculations.

  In Root Nyquist filter operation, calculation is made in the operation band determined by the set Symbol Rate and Rolloff Factor values.
- [Parameter] < bool > = { OFF | ON }

  OFF: Root Nyquist filter operation mode off
  Calculate by using the carrier band width set with the [:SENSe<ch>]:ACP:CBWidth command.
  Calculate by using the adjacent channel measurement band value set with the [:SENSe<ch>]:ACP:CSBW:DATA command.

  ON: Root Nyquist filter operation mode on
- [Query reply] { OFF | ON }
- [Example]

```
'--------- Set the root nyquist filter's parameters ----------------
Call ibwrt (analyzer%, ":ACP:RNYQ:SRAT 2.84MHZ")
Call ibwrt (analyzer%, ":ACP:RNYQ:RFAC 0.22")
'------------ Available root nyquist filter -------------------
Call ibwrt (analyzer%, ":ACP:RNYQ ON")
```

- [Relevant commands] [:SENSe<ch>]:ACP:RNYQuist:SRATe
  [:SENSe<ch>]:ACP:RNYQuist:RFACtor

5.4.83 [:SENSe<ch>]:ACP:RNYQuist:SRATe

- [Command syntax] [:SENSe<ch>]:ACP:RNYQuist:SRATe < real >
  [:SENSe<ch>]:ACP:RNYQuist:SRATe?
- [Function description] Sets the Symbol Rate for Root Nyquist filter operation in ACP measurement

  This command sets the Symbol Rate value used in Root Nyquist filter operation mode.
  The Symbol Rate value is set as the frequency that is the reciprocal of the rate.
- [Parameter] <real> = Frequency (GHz/MHz/kHz/Hz)
- [Query reply] NR3 (Real value: Unit Hz)
- [Example] Call ibwrt (analyzer%, ":ACP:RNYQ:SRAT 1.2288MHZ")
- [Relevant commands] [:SENSe<ch>]:ACP:RNYQuist
  [:SENSe<ch>]:ACP:RNYQuist:RFACtor

5.4.84 [:SENSe<ch>]:ACP:RNYQuist:RFACtor

- [Command syntax] [:SENSe<ch>]:ACP:RNYQuist:RFACTor < real >
  [:SENSe<ch>]:ACP:RNYQuist:RFACTor?
- [Function description] Sets the Rolloff Factor for Root Nyquist filter operation in ACP measurement

  This command sets the Rolloff Factor value used when the Root Nyquist filter operation is used.
- [Parameter] <real> = Rolloff Factor
  Setting range: 0.01 - 0.99
- [Query reply] NR3 (Real value)
- [Example] Call ibwr (analyzer%, ":ACP:RNYQ:RFAC 0.4")
- [Relevant commands] [:SENSe<ch>]:ACP:RNYQuist
  [:SENSe<ch>]:ACP:RNYQuist:SRATe

5.4.85 [:SENSe<ch>]:ACP:NCORrection[:STATe]

- [Command syntax] [:SENSe<ch>]:ACP:NCORrection[:STATe] < bool >
  [:SENSe<ch>]:ACP:NCORrection[:STATe]?
- [Function description] Sets the noise correction function to ON or OFF in ACP measurement

  This instrument has the function of measuring its internal noise level and correcting it to extend the dynamic range in this measurement. This command sets ON and OFF of the correction function.

  Normally, the noise correction is off by default.

  If the noise correction function is set to ON, correction data acquisition is performed automatically when entering this measurement mode and changing measurement parameters.
- [Parameter] < bool > = { OFF | ON }
  OFF: Noise correction function is off.
  ON: Noise correction function is on.
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":ACP:NCOR:STAT ON")
- [Relevant commands]

5.4.86 [:SENSe<ch>]:ACP:POWer:LEVel:AUTO

- [Command syntax] [:SENSe<ch>]:ACP:POWer:LEVel:AUTO
- [Function description] Executes the Auto Level Set function in ACP measurement.

  Sets the reference level and ATT to the optimum value in accordance with the signal to be measured. The signal to be measured is set to the optimum value assuming that the WCDMA wave is input.

  After the automatic adjustment is complete, the "RANGing" bit in the standard operation status register is set.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":ACP:POW:LEV:AUTO")
  Call ibwrt (analyzer%, "*WAI")
- [Relevant commands] [:SENSe<ch>]:MCACp:POWer:LEVel:AUTO
  [:SENSe<ch>]:SEMask:POWer:LEVel:AUTO

5.4.87 [:SENSe<ch>]:MCACp:RNYQuist

- [Command syntax] [:SENSe<ch>]:MCACp:RNYQuist < bool >
  [:SENSe<ch>]:MCACp:RNYQuist?
- [Function description] Sets Root Nyquist filter operation mode to ON or OFF in Multi Carrier ACP measurement

  This command sets whether Root Nyquist filter operation is used for the power calculation in Multi Carrier ACP measurement.

  When the parameter is set to ON, Root Nyquist filter operation is performed in power calculation of all carriers and in power calculation of all adjacent channels.

  In Root Nyquist filter operation, calculation is made in the operation band determined by the set Symbol Rate and Rolloff Factor values.
- [Parameter] < bool > = { OFF | ON }

  OFF: Root Nyquist filter operation mode off
  Calculate by using the measurement carrier and adjacent channel bandwidth set with the [:SENSe<ch>]:MCACp:PARameter{1|2|....|16}:BWIDth command.

  ON: Root Nyquist filter operation mode on
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":MCAC:RNYQ:RFAC 0.22")
  Call ibwrt (analyzer%, ":MCAC:RNYQ:SRAT 3.84MHZ")
  Call ibwrt (analyzer%, ":MCAC:RNYQ ON")
- [Relevant commands] [:SENSe<ch>]:MCACp:RNYQuist:SRATe
  [:SENSe<ch>]:MCACp:RNYQuist:RFACtor

5.4.88 [:SENSe<ch>]:MCACp:RNYQuist:SRATe

- [Command syntax] [:SENSe<ch>]:MCACp:RNYQuist:SRATe < real >
  [:SENSe<ch>]:MCACp:RNYQuist:SRATe?
- [Function description] Sets the Symbol Rate for Root Nyquist filter operation in Multi Carrier ACP measurement.

  This command sets the Symbol Rate value used in Root Nyquist filter operation mode.
  The Symbol Rate value is set as the frequency that is the reciprocal of the rate.
- [Parameter] <real> = Frequency (GHz/MHz/kHz/Hz)
- [Query reply] NR3 (Real value: Unit Hz)
- [Example] Call ibwrt (analyzer%, ":MCAC:RNYQ:SRAT 3.84MHZ")
  Call ibwrt (analyzer%, ":MCAC:RNYQ:RFAC 0.22")
  Call ibwrt (analyzer%, ":MCAC:RNYQ ON")
- [Relevant commands] [:SENSe<ch>]:MCACp:RNYQuist
  [:SENSe<ch>]:MCACp:RNYQuist:RFACtor

5.4.89 [:SENSe<ch>]:MCACp:RNYQuist:RFACtor

- [Command syntax] [:SENSe<ch>]:MCACp:RNYQuist:RFACtor < real >
  [:SENSe<ch>]:MCACp:RNYQuist:RFACtor?
- [Function description] Sets the Rolloff Factor for Root Nyquist filter operation in Multi Carrier ACP measurement

  This command sets the Rolloff Factor value used when the Root Nyquist filter operation is used.
- [Parameter] < real > = Rolloff Factor
  Setting range: 0.01 - 0.99
- [Query reply] NR3 (Real value)
- [Example] Call ibwrt (analyzer%, ":MCAC:RNYQ:SRAT 3.84MHZ")
  Call ibwrt (analyzer%, ":MCAC:RNYQ:RFAC 0.22")
  Call ibwrt (analyzer%, ":MCAC:RNYQ ON")
- [Relevant commands] [:SENSe<ch>]:MCACp:RNYQuist
  [:SENSe<ch>]:MCACp:RNYQuist:SRATe

5.4.90 [:SENSe<ch>]:MCACp:AVERage:COUNt

- [Command syntax] [:SENSe<ch>]:MCACp:AVERage:COUNt < int >
  [:SENSe<ch>]:MCACp:AVERage:COUNt?
- [Function description] Sets the averaging count in Multi Carrier ACP measurement

  This command sets the averaging count in Multi Carrier ACP measurement.

  After sweep is repeated the number of times specified here as the averaging count and the operation of the measured value is completed, the AVERaging and MEASuring bits in the standard operation status register are set.
- [Parameter] <int> = Averaging count
- [Query reply] NR1 (Integer)
- [Example] Call ibwrt (analyzer%, ":MCAC:AVER:COUN 5")
- [Relevant commands] [:SENSe<ch>]:MCACp:AVERage[:STATe]
  [:SENSe<ch>]:MCACp:AVERage:MODE

5.4.91 [:SENSe<ch>]:MCACp:AVERage[:STATe]

- [Command syntax] [:SENSe<ch>]:MCACp:AVERage[:STATe] < bool >
  [:SENSe<ch>]:MCACp:AVERage[:STATe]?
- [Function description] Sets the averaging operation mode to ON or OFF in Multi Carrier ACP measurement

  This command sets the averaging operation mode to ON or OFF in Multi Carrier ACP measurement. After the average mode is switched to ON or OFF, the Multi Carrier ACP operation that was performed up to then is reset.
- [Parameter] < bool > = { OFF | ON }
  OFF: Ordinary operation mode
  ON: Averaging operation mode
- [Query reply] { OFF | ON }
- [Example] '------------ Set the average count to 10 times --------------------------
  Call ibwrt (analyzer%, ":MCAC:AVER:COUN 10")
  ------------- Set the Power measuremnt mode to average mode -----
  Call ibwrt (analyzer%, ":MCAC:AVER:STAT ON")
- [Relevant commands] [:SENSe<ch>]:MCACp:AVERage:COUNt
  [:SENSe<ch>]:MCACp:AVERage:MODE

5.4.92 [:SENSe<ch>]:MCACp:AVERage:MODE

- [Command syntax] [:SENSe<ch>]:MCACp:AVERage:MODE < type >
  [:SENSe<ch>]:MCACp:AVERage:MODE?
- [Function description] Specifies the operation type in averaging operation mode in Multi Carrier ACP measurement

  This command specifies the operation type after the set averaging count is reached in Multi Carrier ACP measurement.

  There are two operation types in terms of the operation after the set averaging count is reached. In the Continuous type, averaging operation is continued by using the moving average. In the Repeat type, the processing count is reset and averaging operation is performed repeatedly the number of times specified as the averaging count.
- [Parameter] < type > = { CONTinuous | REPeat }

  CONTinuous: Moving average after the averaging count is reached.

  REPeat: Averaging operation is performed repeatedly.
- [Query reply] { CONT | REP }
- [Example] Call ibwrt (analyzer%, ":MCAC:AVER:MODE CONT")
- [Relevant commands] [:SENSe<ch>]:MCACp:AVERage:COUNt
  [:SENSe<ch>]:MCACp:AVERage[:STATe]

5.4.93 [:SENSe<ch>]:MCACp:DATA:MODE

- [Command syntax] [:SENSe<ch>]:MCACp:DATA:MODE < type >
  [:SENSe<ch>]:MCACp:DATA:MODE?
- [Function description] Specifies the measurement parameter setting mode in Multi Carrier ACP measurement

  This command specifies the setting modes of the parameters used in Multi Carrier ACP measurement. There are two modes as shown below.

  Default: Measures in the preset parameter state.

  Manual: Measures in the parameter state before entering the Multi Carrier ACP measurement mode.

  For more information on the measurement parameters used here, refer to the function explanation of the command that saves the parameter values used in default mode [:SENSe<ch>]:MCACp:DATA:SAVE.
- [Parameter] < type > = { DEFault | MANual }

  DEFault: Performs a measurement by using the parameter values saved by the [:SENSe<ch>]:MCACp:DATA:SAVE command.

  MANual: Performs a measurement by using the setting value before entering the Multi Carrier ACP measurement mode.
- [Query reply] { DEF | MAN }
- [Example] Call ibwrt (analyzer%, ":MCAC:DATA:MODE MAN")
- [Relevant commands] [:SENSe<ch>]:MCACp:DATA:SAVE

5.4.94 [:SENSe<ch>]:MCACp:DATA:SAVE

- [Command syntax] [:SENSe<ch>]:MCACp:DATA:SAVE
- [Function description] Saves the measurement parameters in Multi Carrier ACP measurement

  This command saves the current values of the parameters used in Multi Carrier ACP measurement as the values used in default mode.

  The parameters saved by this command are shown below.
  - All parameters set in the Ref/Offs Setup
  - All parameters set in the Root Nyquist Filter Setup
  - Result display mode
  - RBW value
  - VBW value
  - Sweep time
  - Averaging ON/OFF and averaging count
  - Averaging repeat mode
  - Noise correction ON/OFF
  - Trace status
  - Trace detector
  - Trigger
  - Gated sweep
- [Parameter] None
- [Query reply] None
- [Example]

```
'-------- Set each parameter for power measurement --------
Call ibwrt (analyzer%, ":MCAC:AVER:MODE CONT")
Call ibwrt (analyzer%, ":MCAC:AVER:COUN 20")
Call ibwrt (analyzer%, ":MCAC:AVER ON")

'-------- Save the parameter for Multi Carrier ACP meas ---------
Call ibwrt (analyzer%, ":MCAC:DATA:SAVE")
```

- [Relevant commands] [:SENSe<ch>]:MCACp:DATA:MODE

5.4.95 [:SENSe<ch>]:MCACp:PARameter{1|2|...|16}:STATe

- [Command syntax] [:SENSe<ch>]:MCACp:PARameter{1|2|...|16}:STATe < bool >
  [:SENSe<ch>]:MCACp:PARameter{1|2|...|16}:STATe?
- [Function description] Sets the measurement carrier and adjacent channel to ON or OFF in Multi Carrier ACP

  In this function, up to 16 carriers and adjacent channel areas can be specified. By using the specified carriers or adjacent channel areas, execution of power calculation and leakage power calculation is set to ON or OFF.

  For carriers or adjacent channels set to ON, power calculation is performed in accordance with the specified parameters.

  For carriers or adjacent channels set to OFF, the areas can be excluded from the target of power calculation.

  By this function, measurement of carriers that are not output among continuous channels can be omitted.
- [Parameter] < bool > = { OFF | ON }
- [Query reply] { OFF | ON }
- [Example]
  ```
  '-------- Make configuration of the multi carrier output status --------
  Call ibwrt (analyzer%, ":MCAC:PAR1:STAT ON")
  Call ibwrt (analyzer%, ":MCAC:PAR2:STAT OFF")
  Call ibwrt (analyzer%, ":MCAC:PAR3:STAT ON")
  Call ibwrt (analyzer%, ":MCAC:PAR6:STAT ON")
  ```
- [Relevant commands] [:SENSe<ch>]:MCACp:PARameter{1|2|...|16}:FREQuency
  [:SENSe<ch>]:MCACp:PARameter{1|2|...|16}:BWIDth
  [:SENSe<ch>]:MCACp:PARameter{11|12|...|16}:REFerence
  [:SENSe<ch>]:MCACp:PARameter{11|12|...|16}:LIMit

5.4.96 [:SENSe<ch>]:MCACp:PARameter{1|2|...|16}:FREQuency

- [Command syntax] [:SENSe<ch>]:MCACp:PARameter{1|2|...|16}:FREQuency < real >
  [:SENSe<ch>]:MCACp:PARameter{1|2|...|16}:FREQuency?
- [Function description] Sets the offset frequency of the measurement carrier and adjacent channel in Multi Carrier ACP

  This command sets the offset frequency of the carrier or adjacent channel output. The offset frequency refers to the offset frequency from the specified reference carrier.

  In this function, any offset frequency of up to 16 carrier and adjacent channel areas can be set.
- [Parameter] <real> = Offset frequency (GHz/MHz/kHz/Hz)
- [Query reply] NR3 (Real value: Unit Hz)
- [Example]

```
'-------- Make configuration of the multi carrier freq.--------
Call ibwrt (analyzer%, ":MCAC:PAR1:FREQ 0MHz")
Call ibwrt (analyzer%, ":MCAC:PAR2:FREQ 5MHz")
Call ibwrt (analyzer%, ":MCAC:PAR3:FREQ -5MHz")
Call ibwrt (analyzer%, ":MCAC:PAR4:FREQ 10MHz")
Call ibwrt (analyzer%, ":MCAC:PAR5:FREQ -10MHz")
'-------- Make configuration of ACP area.--------
Call ibwrt (analyzer%, ":MCAC:PAR11:FREQ -25MHz")
Call ibwrt (analyzer%, ":MCAC:PAR12:FREQ 25MHz")
Call ibwrt (analyzer%, ":MCAC:PAR13:FREQ -30MHz")
Call ibwrt (analyzer%, ":MCAC:PAR14:FREQ 30MHz")
```

- [Relevant commands] [:SENSe<ch>]:MCACp:PARameter{1|2|...|16}:STATe
  [:SENSe<ch>]:MCACp:PARameter{1|2|...|16}:BWIDth
  [:SENSe<ch>]:MCACp:PARameter{11|12|...|16}:REFerence
  [:SENSe<ch>]:MCACp:PARameter{11|12|...|16}:LIMit

5.4.97 [:SENSe<ch>]:MCACp:PARameter{1|2|...|16}:BWIDth

- [Command syntax] [:SENSe<ch>]:MCACp:PARameter{1|2|...|16}:BWIDth < real >
  [:SENSe<ch>]:MCACp:PARameter{1|2|...|16}:BWIDth?
- [Function description] Sets the channel bandwidth of the measurement carrier and adjacent channel in Multi Carrier ACP

  This command specifies the power operation band of the specified carrier or adjacent channel.

  Power operation is performed in the operation band set with the specified offset frequency as the center. If Root Nyquist filter operation mode is set to ON by the [:SENSe<ch>]:MCACp:RNYQuist command, however, calculation is performed in the band that is dependent on the Root Nyquist filter parameter.
- [Parameter] <real> = Power operation bandwidth (GHz/MHz/kHz/Hz)
- [Query reply] NR3 (Real value: Unit Hz)
- [Example]

```
'-------- Make configuration of the multi carrier freq.--------
Call ibwrt (analyzer%, ":MCAC:PAR1:BWID 3.84MHz")
Call ibwrt (analyzer%, ":MCAC:PAR2:BWID 2. 5MHz")
Call ibwrt (analyzer%, ":MCAC:PAR3:BWID 3.84MHz")
Call ibwrt (analyzer%, ":MCAC:PAR4:BWID 5MHz")
```

- [Relevant commands] [:SENSe<ch>]:MCACp:PARameter{1|2|...|16}:FREQuency
  [:SENSe<ch>]:MCACp:PARameter{1|2|...|16}:STATe
  [:SENSe<ch>]:MCACp:PARameter{11|12|...|16}:REFerence
  [:SENSe<ch>]:MCACp:PARameter{11|12|...|16}:LIMit

5.4.98 [:SENSe<ch>]:MCACp:PARameter{11|12|...|16}:REFerence

- [Command syntax] [:SENSe<ch>]:MCACp:PARameter{11|12|...|16}:REFerence < int >
  [:SENSe<ch>]:MCACp:PARameter{11|12|...|16}:REFerence?
- [Function description] Sets the reference power area of the measurement carrier and adjacent channel in Multi Carrier ACP

  This command specifies the reference power area (the number on the table) when the power operation mode is relative mode in the specified carrier or adjacent channel.

  For the number to set, specify the number on the setting table for this function.
- [Parameter] <int> = Reference power area number
  Setting range: 1-10
- [Query reply] NR1 (Integer: No unit)
- [Example] '-------- Make configuration of the multi carrier data table.--------
  Call ibwrt (analyzer%, ":MCAC:PAR11:REF 1")
  Call ibwrt (analyzer%, ":MCAC:PAR12:REF 3")
- [Relevant commands] [:SENSe<ch>]:MCACp:PARameter{1|2|...|16}:STATe
  [:SENSe<ch>]:MCACp:PARameter{1|2|...|16}:FREQuency
  [:SENSe<ch>]:MCACp:PARameter{1|2|...|16}:BWIDth
  [:SENSe<ch>]:MCACp:PARameter{11|12|...|16}:LIMit

5.4.99 [:SENSe<ch>]:MCACp:PARameter{11|12|...|16}:LIMit

- [Command syntax] [:SENSe<ch>]:MCACp:PARameter{11|12|...|16}:LIMit < real >
  [:SENSe<ch>]:MCACp:PARameter{11|12|...|16}:LIMit?
- [Function description] Sets the limit value for checking pass/fail of the measurement carrier and adjacent channel measurement result in Multi Carrier ACP measurement

  The relative value data of the power value obtained in the specified adjacent channel area is compared with the limit value set with this command. For the units of all the limit values that can be set with this command, relative values (dB) are used.
- [Parameter] <real> = Limit value (dB)
  Setting range: -100.0 - 100.0
- [Query reply] NR3 (Real value: Unit dB)
- [Example] '-------- Make configuration of the multi carrier data table.--------
  Call ibwrt (analyzer%, ":MCAC:PAR11:LIM -45DB")
  Call ibwrt (analyzer%, ":MCAC:PAR12:LIM -60DB")
- [Relevant commands] [:SENSe<ch>]:MCACp:PARameter{1|2|...|16}:STATe
  [:SENSe<ch>]:MCACp:PARameter{1|2|...|16}:FREQuency
  [:SENSe<ch>]:MCACp:PARameter{11|12|...|16}:BWIDth
  [:SENSe<ch>]:MCACp:PARameter{11|12|...|16}:REFerence

5.4.100 [:SENSe<ch>]:MCACp:NCORrection[:STATe]

- [Command syntax] [:SENSe<ch>]:MCACp:NCORrection[:STATe] < bool >
  [:SENSe<ch>]:MCACp:NCORrection[:STATe]?
- [Function description] Sets the noise correction function to ON or OFF in Multi Carrier ACP measurement

  This instrument has the function of measuring its internal noise level and correcting it to extend the dynamic range in this measurement. This command sets ON and OFF of the correction function.

  Normally, the noise correction is OFF by default.

  If the noise correction function is set to ON, correction data acquisition is performed automatically when entering this measurement mode and changing measurement parameters.
- [Parameter] < bool > = { OFF | ON }
  OFF: Noise correction function is off.
  ON: Noise correction function is on.
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":MCAC:NCOR:STAT ON")
- [Relevant commands]

5.4.101 [:SENSe<ch>]:MCACp:POWer:LEVel:AUTO

- [Command syntax] [:SENSe<ch>]:MCACp:POWer:LEVel:AUTO
- [Function description] Executes the Auto Level Set function in Multi Carrier ACP measurement.

  Sets the reference level and ATT to the optimum value according to the signal to be measured. The signal to be measured is set to the optimum value assuming that four waves of WCDMA are input.

  After the automatic adjustment is complete, the "RANGing" bit in the standard operation status register is set.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":MCAC:POW:LEV:AUTO")
  Call ibwrt (analyzer%, "*WAI")
- [Relevant commands] [:SENSe<ch>]:ACP:POWer:LEVel:AUTO
  [:SENSe<ch>]:SEMask:POWer:LEVel:AUTO

5.4.102 [:SENSe<ch>]:MCACp:CARRier:ADJust

- [Command syntax] [:SENSe<ch>]:MCACp:CARRier:ADJust < real >
  [:SENSe<ch>]:MCACp:CARRier:ADJust?
- [Function description] Sets the Carrier Freq Adjustment in multi carrier ACP measurement.

  Sets the frequency value which adjusts the multi carrier display position.
- [Parameter] < real > = Carrier Freq Adjustment (GHz/MHz/kHz/Hz)
- [Query reply] NR3 (Real value: Unit Hz)
- [Example] Call ibwrt (analyzer%, ":MCAC:CARR:ADJ 5MHZ")
  Call ibwrt (analyzer%, ":MCAC:CARR:ADJ:STAT ON")
- [Relevant commands] [:SENSe<ch>]:MCACp:CARRier:ADJust:STATe
  [:SENSe<ch>]:FREQuency<screen>:CENTer
  [:SENSe<ch>]:FREQuency<screen>:CHANnel:NUMBer

5.4.103 [:SENSe<ch>]:MCACp:CARRier:ADJust:STATe

- [Command syntax] [:SENSe<ch>]:MCACp:CARRier:ADJust:STATe < bool >
  [:SENSe<ch>]:MCACp:CARRier:ADJust:STATe?
- [Function description] Sets the Carrier Freq Adjustment to ON or OFF in multi carrier ACP measurement.

  Sets the function which adjusts the multi carrier display position to ON or OFF. When this function is set to ON and the center frequency is input, the center frequency is set to (center frequency input value) + (Carrier Freq Adjustment value).
- [Parameter] < bool > = { OFF | ON }
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":MCAC:CARR:ADJ 5MHZ")
  Call ibwrt (analyzer%, ":MCAC:CARR:ADJ:STAT ON")
- [Relevant commands] [:SENSe<ch>]:MCACp:CARRier:ADJust
  [:SENSe<ch>]:FREQuency<screen>:CENTer
  [:SENSe<ch>]:FREQuency<screen>:CHANnel:NUMBer

5.4.104 [:SENSe<ch>]:SPURious:DATA[:NUMBer{1|2|3}]

- [Command syntax]
  [:SENSe<ch>]:SPURious:DATA [:NUMBer{1|2|3}] < real1 >, < real2 >, <bool3>, <real3 > , <bool4> , < real4 >, <bool5>, < real5 >, <real6>, <bool7>, <real7>, <bool8>,<real9>
- [Function description]
  Registers parameters to the Spurious table in spurious measurement

  This command registers 13 types of parameters used in spurious measurement to the Spurious table. The 13 types of parameters to be specified are: sweep start frequency, stop frequency, RBW AUTO/MANUAL, RBW, VBW AUTO/MANUAL, VBW, sweep time AUTO/MANUAL, sweep time, reference level, input ATT AUTO/MANUAL, input ATT, Preamp ON/OFF, and spurious level judgment values. With these 13 types of parameters as a set, 15 sets of settings can be registered in the table. If patterns of more than 15 sets are sent, 16^{th} and later patterns are ignored.

  When executing spurious measurement, spurious search measurement is performed in the order of registration in the table.
- [Parameter]
  <real1> = Sweep start frequency (GHz/MHz/kHz/Hz)
  <real2> = Sweep stop frequency (GHz/MHz/kHz/Hz)
  <bool3> = { OFF | ON } RBW AUTO/MANUAL
  <real3> = RBW (MHz/kHz/Hz)
  <bool4> = { OFF | ON } VBW AUTO/MANUAL
  <real4> = VBW (MHz/kHz/Hz)
  <bool5> = { OFF | ON } Sweep time AUTO/MANUAL
  <real5> = Sweep time (s/ms/μs)
  <real6> = Reference level (dBm)
  <bool7> = { OFF | ON } input ATT AUTO/MANUAL
  <real7> = Input attenuator (dB)
  <bool8> = { OFF | ON } Preamp ON/OFF
  <real9> = Spurious level judgment value (dBm)
- [Query reply]
  None
- [Example]
  Call ibwrt (analyzer%, ":SPUR:DATA 10MHZ, 900MHZ, OFF, 300KHZ, OFF, 300KHZ, OFF, 1.1S, 5DBM, ON, 10DB, OFF, 0DBM")
  Call ibwrt (analyzer%, ":SPUR:DATA 900MHZ, 1.9GHZ, OFF, 100KHZ, OFF, 100KHZ, OFF, 2S , 5DBM, ON, 10DB, OFF, 0DBM")
  Call ibwrt (analyzer%, ":SPUR:DATA 1.9GHZ, 8GHZ, OFF, 300KHZ, OFF, 100KHZ, OFF, 1.4S, 5DBM, ON, 10DB, OFF, 0DBM")
- [Relevant commands]
  [:SENSe<ch>]:SPURious:DATA[:NUMBer{1|2|3 }]:DELete

5.4.105 [:SENSe<ch>]:SPURious:DATA[:NUMBer{1|2|3}]:ACTive

- [Command syntax] [:SENSe<ch>]:SPURious:DATA[:NUMBer{1|2|3}]:ACTive
- [Function description] Selects the table used in spurious measurement

  For spurious measurement, three spurious tables are provided. Of the three tables, select the table to be used for measurement.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":SPUR:DATA:NUMB1:ACT")
- [Relevant commands] [:SENSe<ch>]:SPURious:DATA[:NUMBer{1|2|3}]
  [:SENSe<ch>]:SPURious:DATA[:NUMBer{1|2|3}]:DELete

5.4.106 [:SENSe<ch>]:SPURious:DATA[:NUMBer{1|2|3}]:DELete

- [Command syntax] [:SENSe<ch>]:SPURious:DATA:DELete
- [Function description] Deletes all data registered in the spurious table in spurious measurement

  This command deletes all the patterns of the measurement parameters registered in the spurious table used in spurious measurement.

  When registering new patterns with the [:SENSe<ch>]:SPURious:DATA command, delete all the patterns with this command before adding new patterns.

  When there are any patterns that have already been registered, the [:SENSe<ch>]:SPURious:DATA command performs additional registration only.
- [Parameter] None
- [Query reply] None
- [Example] '---------- Initialize spurious table ------------------------
  Call ibwrt (analyzer%, ":SPUR:DATA:DEL")

  '----------- After then , set data into spurious table -----
  Call ibwrt (analyzer%, ":SPUR:DATA 900MHZ, 1.9GHZ, OFF, 100KHZ,
  OFF, 100KHZ, OFF, 2S , 5DBM, ON, 10DB, OFF, 0DBM")
  Call ibwrt (analyzer%, ":SPUR:DATA 1.9GHZ, 8GHZ, OFF, 300KHZ,
  OFF, 100KHZ, OFF, 1.4S, 5DBM, ON, 10DB, OFF, 0DBM")
- [Relevant commands] [:SENSe<ch>]:SPURious:DATA[:NUMBer{1|2|3}]:ACTive
  [:SENSe<ch>]:SPURious:DATA[:NUMBer{1|2|3}]

5.4.107 [:SENSe<ch>]:SPURious:DATA:MODE

- [Command syntax] [:SENSe<ch>]:SPURious:DATA:MODE < type >
  [:SENSe<ch>]:SPURious:DATA:MODE?
- [Function description] Specifies the the setting mode of the parameter in spurious measurement

  This command is used to specify the setting mode of the parameter used in the Spurious measurement.

  There are two modes as shown below:

  Default: Measures in the preset parameter state.

  Manual: Measures in the parameter state that is set before the measurement mode enters into the Spurious measurement mode.

  For more information on the measurement parameter used here, refer to the function description of the command [:SENSe<ch>]:SPURious:DATA:SAVE, which saves the parameter values used in the Default mode.
- [Parameter] < type > = { DEFault | MANual }

  DEFault: Measures by using the parameter values saved by the [:SENSe<ch>]:SPURious:DATA:SAVE command.

  MANual: Measures by using the setting values that are used before the measurement mode enters into the Spurious measurement mode.
- [Query reply] { DEF | MAN }
- [Example] Call ibwrt (analyzer%, ":SPUR:DATA:MODE MAN")
- [Relevant commands] [:SENSe<ch>]:SPURious:DATA:SAVE

5.4.108 [:SENSe<ch>]:SPURious:DATA:SAVE

- [Command syntax] [:SENSe<ch>]:SPURious:DATA:SAVE
- [Function description] Saves the parameter in spurious measurement

  This command is used to save the current values of the parameter used in the Spurious measurement as the values used in the Default mode. The parameter saved by this command is shown below:
  - Parameter in the Spurious table
  - Spurious table number to be selected
  - Trace status
  - Trace detector
  - Trigger
  - Gated sweep
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":SPUR:DATA:SAVE")
- [Relevant commands] [:SENSe<ch>]:SPURious:DATA:MODE

5.4.109 [:SENSe<ch>]:IM:ORDer

- [Command syntax] [:SENSe<ch>]:IM:ORDer < int >
  [:SENSe<ch>]:IM:ORDer?
- [Function description] Sets the maximum order in IM measurement

  This command sets the maximum order of the signal to be measured in IM measurement. Only odd orders may be set.
- [Parameter] <int> = Measurement distortion signal order
  Setting range: 3, 5, 7, 9
- [Query reply] NR1 (Integer)
- [Example] Call ibwr (analyzer%, ":IM:ORD 9")
- [Relevant commands] [:SENSe<ch>]:IM:THReshold{3|5|7|9}

5.4.110 [:SENSe<ch>]:IM:THReshold{3|5|7|9}

- [Command syntax] [:SENSe<ch>]:IM:THReshold{3|5|7|9} < real >
  [:SENSe<ch>]:IM:THReshold{3|5|7|9}?
- [Function description] Sets the pass/fail judgment values of the distortion signal in IM measurement

  This command sets the judgment value for the distortion signal value measured after IM measurement. Compares the measured value with the set judgment value and judges as fail for the measured value that is larger than the judgment value.
- [Parameter] <real> = Judgment value (dB)
- [Query reply] NR3 (Real value: Unit dB)
- [Example] Call ibwrt (analyzer%, ":IM:THR3 -55DB")
- [Relevant commands] [:SENSe<ch>]:IM:ORDer

5.4.111 [:SENSe<ch>]:IM:LIM:STATe

- [Command syntax] [:SENSe<ch>]:IM:LIM:STATe < bool >
  [:SENSe<ch>]:IM:LIM:STATe?
- [Function description] Sets the pass/fail judgment function to ON or OFF in IM measurement

  This command sets the pass/fail judgment function for the distortion signal value measured after IM measurement to ON or OFF. The judgment value is set by the [:SENSe<ch>]:IM:THReshold{3|5|7|9} command.
- [Parameter] < bool > = { OFF | ON }
  OFF: Pass/Fail judgment is not performed.
  ON: Pass/Fail judgment is performed.
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":IM:LIM:STAT ON")
- [Relevant commands] [:SENSe<ch>]:IM:THReshold{3|5|7|9}

5.4.112 [:SENSe<ch>]:IM:DATA:MODE

- [Command syntax] [:SENSe<ch>]:IM:DATA:MODE < type >
  [:SENSe<ch>]:IM:DATA:MODE?
- [Function description] Specifies the measurement parameter setting mode in IM measurement

  This command specifies the setting modes of the parameters used in IM measurement.

  There are two modes as shown below.

  Default: Measures in the preset parameter state.

  Manual: Measures in the parameter state before entering the IM measurement mode.

  For more information on the measurement parameters used here, refer to the function explanation of the command [:SENSe<ch>]:IM:DATA:SAVE which saves the parameter values used in default mode.
- [Parameter] < type > = { DEFault | MANual }

  DEFault: Performs a measurement by using the parameter values saved by the [:SENSe<ch>]:IM:DATA:SAVE command.

  MANual: Performs a measurement by using the setting value before entering the IM measurement mode.
- [Query reply] { DEF | MAN }
- [Example] Call ibwrt (analyzer%, ":IM:DATA:MODE DEF")
- [Relevant commands] [:SENSe<ch>]:IM:DATA:SAVE

5.4.113 [:SENSe<ch>]:IM:DATA:SAVE

- [Command syntax] [:SENSe<ch>]:ACP:DATA:SAVE
- [Function description] Saves the measurement parameter in IM measurement.

  This command saves the current values of the parameters used in IM measurement as the values used in default mode

  The parameters saved by this command are shown below.
  - Measurement distortion signal order
  - Pass/Fail judgment value
  - Pass/Fail judgment ON/OFF
  - RBW value
  - VBW value
  - Sweep time
  - Span frequency
  - Averaging ON/OFF and averaging count
  - Averaging repeat mode
  - Trace status
  - Trace detector
  - Trigger
  - Gated sweep
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":IM:DATA:SAVE")
- [Relevant commands] [:SENSe<ch>]:IM:DATA:MODE

5.4.114 [:SENSe<ch>]:HARMonics:FFRequency

- [Command syntax] [:SENSe<ch>]:HARMonics:FFRequency < real >
  [:SENSe<ch>]:HARMonics:FFRequency?
- [Function description] Sets the reference signal frequency in harmonic measurement

  This command sets the frequency that becomes the reference during harmonic measurement. The frequency set in this command is enabled when ON is specified in the [:SENSe<ch>]:HARMonics:FFReqency:STATe command. When ON is selected, the frequency range is set automatically by this frequency and the harmonics order set in the [:SENSe<ch>]:HARMonics:NUMBer command.
- [Parameter] <real> = Reference frequency (GHz/MHz/kHz/Hz)
- [Query reply] NR3 (Real value: Unit Hz)
- [Example] Call ibwrt (analyzer%, ":HARM:FFR 834MHZ")
- [Relevant commands] [:SENSe<ch>]:HARMonics:FFRequency:STATe
  [:SENSe<ch>]:HARMonics:NUMBer

5.4.115 [:SENSe<ch>]:HARMonics:FFRequency:STATe

- [Command syntax] [:SENSe<ch>]:HARMonics:FFRequency:STATe < bool >
  [:SENSe<ch>]:HARMonics:FFRequency:STATe?
- [Function description] Sets the reference signal frequency mode in harmonic measurement

  This command sets the selection mode of the frequency that becomes the reference during harmonic measurement. There are two selection modes: AUTO mode, in which the center frequency before starting harmonic measurement is set to the reference signal frequency automatically, and MANUAL mode, in which the preset frequency is set as the reference signal frequency.
- [Parameter] < bool > = { OFF | ON }
  OFF: AUTO mode
  ON: MANUAL mode
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":HARM:FFR:STAT ON")
- [Relevant commands] [:SENSe<ch>]:HARMonics:FFRequnecy
  [:SENSe<ch>]:HARMonics:NUMBer

5.4.116 [:SENSe<ch>]:HARMonics:NUMBer

- [Command syntax] [:SENSe<ch>]:HARMonics:NUMBer < int >
  [:SENSe<ch>]:HARMonics:NUMBer?
- [Function description] Harmonics order to be measured in harmonic measurement

  This command sets the order of harmonics to be measured during harmonic measurement. The span frequency at the time of measurement is determined in accordance with the order set in this command and the carrier frequency.
- [Parameter] <int> = Order
  Setting range: 2 - 10
- [Query reply] NR1 (Integer)
- [Example] Call ibwrt (analyzer%, ":HARM:NUMB 5")
- [Relevant commands] [:SENSe<ch>]:HARMonics:FFRequency
  [:SENSe<ch>]:HARMonics:FFRequency:STATe

5.4.117 [:SENSe<ch>]:SEMask:CBWidth

- [Command syntax] [:SENSe<ch>]:SEMask:CBWidth < real >
  [:SENSe<ch>]:SEMask:CBWidth?
- [Function description] Sets the reference power operation width in Spectrum Emission Mask measurement

  This command sets the bandwidth for reference power operation used in Spectrum Emission Mask measurement.

  Perform power operation in the frequency band set with this command and set it as the reference power of the mask function in Spectrum Emission Mask measurement.

  If Root Nyquist filter operation is specified by the

  [:SENSe<ch>]:SEMask:RNYQuist command, the reference power is calculated in the bandwidth determined by the parameter of the Root Nyquist filter.

  If PEAK is selected as the operation mode of the reference power by the [:SENSe<ch>]:SEMask:RPOWer:MODE command, the maximum power in the frequency band set in this command becomes the reference power.
- [Parameter] <real> = Frequency (GHz/MHz/kHz/Hz)
- [Query reply] NR3 (Real value: Unit Hz)
- [Example] Call ibwrt (analyzer%, ":SEM:CBW 3.84MHZ")
- [Relevant commands] [:SENSe<ch>]:SEMask:RNYQuist
  [:SENSe<ch>]:SEMask:RNYQuist:SRATe
  [:SENSe<ch>]:SEMask:RNYQuist:RFACtor
  [:SENSe<ch>]:SEMask:RPOWer:MODE

5.4.118 [:SENSe<ch>]:SEMask:RNYQuist

- [Command syntax] [:SENSe<ch>]:SEMask:RNYQuist < bool >
  [:SENSe<ch>]:SEMask:RNYQuist?
- [Function description] Sets the Root Nyquist filter operation mode to ON or OFF in Spectrum Emission Mask measurement

  This command sets whether Root Nyquist filter operation is used for the reference power calculation in Spectrum Emission Mask measurement.

  When this parameter is set to ON: The Root Nyquist filter operation is used in carrier power operation when obtaining the reference power.

  In Root Nyquist filter operation, calculation is made in the operation band determined by the set Symbol Rate and Rolloff Factor values.
- [Parameter] < bool > = { OFF | ON }

  OFF: Sets the Root Nyquist filter calculation mode to OFF. Calculates the power by using the bandwidth set by [:SENSe<ch>]:SEMask:CBWidth command.

  ON: Sets the Root Nyquist filter calculation mode to ON.
- [Query reply] { OFF | ON }
- [Example]
  '-------------- Set root nyquist filter parameter's ------------------
  Call ibwrt (analyzer%, ":SEM:RNYQ:SRAT 3.84MHZ")
  Call ibwrt (analyzer%, ":SEM:RNYQ:RFAC 0.22")
  '------ Set the Root Nyquist filter condition to ON ---------------
  Call ibwrt (analyzer%, ":SEM:RNYQ ON")
- [Relevant commands] [:SENSe<ch>]:SEMask:CBWidth
  [:SENSe<ch>]:SEMask:RNYQuist:SRATe
  [:SENSe<ch>]:SEMask:RNYQuist:RFACtor

5.4.119 [:SENSe<ch>]:SEMask:RNYQuist:SRATe

- [Command syntax] [:SENSe<ch>]:SEMask:RNYQuist:SRATe < real >
  [:SENSe<ch>]:SEMask:RNYQuist:SRATe?
- [Function description] Sets the Symbol Rate for Root Nyquist filter operation in Spectrum Emission Mask measurement

  This command sets the Symbol Rate value used in Root Nyquist filter operation mode.
  The Symbol Rate value is set as the frequency that is the reciprocal of the rate.
- [Parameter] <real> = Frequency (GHz/MHz/kHz/Hz)
- [Query reply] NR3 (Real value: Unit Hz)
- [Example] Call ibwrt (analyzer%, ":SEM:RNYQ:SRAT 1.228MHZ")
- [Relevant commands] [:SENSe<ch>]:SEMask:CBWidth
  [:SENSe<ch>]:SEMask:RNYQuist
  [:SENSe<ch>]:SEMask:RNYQuist:RFACtor

5.4.120 [:SENSe<ch>]:SEMask:RNYQuist:RFACtor

- [Command syntax] [:SENSe<ch>]:SEMask:RNYQuist:RFACtor < real >
  [:SENSe<ch>]:SEMask:RNYQuist:RFACtor?
- [Function description] Sets the Rolloff Factor for Root Nyquist filter operation in Spectrum Emission Mask measurement

  This command sets the Rolloff Factor value used when the Root Nyquist filter operation is used.
- [Parameter] <real> = Rolloff Factor
  Setting range: 0.01 - 0.99
- [Query reply] NR3 (Real value)
- [Example] Call ibwrt (analyzer%, ":SEM:RNYQ:RFAC 0.11")
- [Relevant commands] [:SENSe<ch>]:SEMask:CBWidth
  [:SENSe<ch>]:SEMask:RNYQuist
  [:SENSe<ch>]:SEMask:RNYQuist:SRATe

5.4.121 [:SENSe<ch>]:SEMask:DATA

- [Command syntax] [:SENSe<ch>]:SEMask:DATA < real1 >, < real2 >, < real3 >, < real4 >, < real5 >, < real6 >, < real7 >, < type >
- [Function description] Sets the parameter table for measurement in Spectrum Emission Mask measurement

  This command makes settings in the Offset Setup table, which defines the parameters required in Spectrum Emission Mask measurement. In this command, set parameters related to one offset frequency as a set. To set multiple patterns in the table, send this command several times. To set all the table data again, it is necessary to initialize the present table state by the [:SENSe<ch>]:SEMask:DATA:DELete command.
- [Parameter] <real1> = Offset Start frequency (GHz/MHz/kHz/Hz)

  <real2> = Offset Stop frequency (GHz/MHz/kHz/Hz)

  <real3> = Integral band (GHz/MHz/kHz/Hz)

  <real4> = Absolute level judgment value start value (dBm)

  <real5> = Absolute level judgment value stop value (dBm)

  <real6> = Relative level judgment value start value (dB)

  <real7> = Relative level judgment value stop value (dB)

  <type> = { ABS | REL | AAR | AOR }

  ABS: Judges only with the absolute level judgment value.

  REL: Judges only with the relative level judgment value.

  AAR: Judges with the AND condition of the absolute and relative level judgment values.

  AOR: Judges with the OR condition of the absolute and relative level judgment values.
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":SEM:DATA:DEL")

  Call ibwrt (analyzer%, ":SEM:DATA 2.715MHZ,3.215MHZ,1MHZ, -35DBM, -45dBM, -15DB, -25DB, AOR")
- [Relevant commands] [:SENSe<ch>]:SEMask:DATA:DELete

5.4.122 [:SENSe<ch>]:SEMask:DATA:DELete

- [Command syntax] [:SENSe<ch>]:SEMask:DATA:DELete
- [Function description] Clears all data in the parameter table for measurement in Spectrum Emission Mask measurement

  This command initializes the Offset Setup table, which defines the parameters required in Spectrum Emission Mask measurement. Data set by initialization is all cleared.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":SEM:DATA:DEL")
- [Relevant commands] [:SENSe<ch>]:SEMask:DATA

5.4.123 [:SENSe<ch>]:SEMask:RPOWer:MODE

- [Command syntax] [:SENSe<ch>]:SEMask:RPOWer:MODE < type >

  [:SENSe<ch>]:SEMask:RPOWer:MODE?
- [Function description] Sets the reference power calculation mode in Spectrum Emission Mask measurement

  This command sets the calculation method of the reference power value that becomes the reference of the relative value mask.

  There are two calculation methods: the Channel power mode, in which power operation is performed in the specified band, and the Peak power mode, in which the peak power value in the specified bandwidth is used.
- [Parameter] < type > = { CHANnel | PEAK }

  CHANnel: Channel power mode

  PEAK: Peak power mode
- [Query reply] { CHAN | PEAK }
- [Example] Call ibwrt (analyzer%, ":SEM:RPOW:MODE CHAN")
- [Relevant command] [:SENSe<ch>]:SEMask:CBWidth

5.4.124 [:SENSe<ch>]:SEMask:AVERage:COUNt

- [Command syntax] [:SENSe<ch>]:SEMask:AVERage:COUNt < int >
  [:SENSe<ch>]:SEMask:AVERage:COUNt?
- [Function description] Sets the averaging count in Spectrum Emission Mask measurement

  This command sets the averaging count in Spectrum Emission Mask measurement.

  After sweep is repeated the number of times specified here as the averaging count and the operation of the measured value is completed, the AVERaging and MEASuring bits in the standard operation status register are set.
- [Parameter] <int> = Averaging count
- [Query reply] NR1 (Integer)
- [Example] Call ibwrt (analyzer%, ":SEM:AVER:COUN 10")
- [Relevant commands] [:SENSe<ch>]:SEMask:AVERage[:STATe]

5.4.125 [:SENSe<ch>]:SEMask:AVERage[:STATe]

- [Command syntax] [:SENSe<ch>]:SEMask:AVERage:[:STATe] < bool >
  [:SENSe<ch>]:SEMask:AVERage:[:STATe]?
- [Function description] Sets the averaging measurement function to ON or OFF in Spectrum Emission Mask measurement

  This command is used to set the averaging calculation mode in the Spectrum Emission Mask measurement to ON or OFF.

  If the averaging calculation mode is switched ON and OFF, the Spectrum Emission Mask calculation process that has been performed until then is reset.
- [Parameter] < bool > = { OFF | ON }
  OFF: Averaging function is set to off.
  ON: Averaging function is set to on.
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":SEM:AVER:STAT ON")
- [Relevant commands] [:SENSe<ch>]:SEMask:AVERage[:STATe]

5.4.126 [:SENSe<ch>]:SEMask:AVERage:MODE

- [Command syntax] [:SENSe<ch>]:SEMask:AVERage:MODE < type >
  [:SENSe<ch>]:SEMask:AVERage:MODE?
- [Function description] Specifies the operation type in average operation mode in Spectrum Emission Mask measurement.

  This command specifies the operation type after the set averaging count is reached in Spectrum Emission Mask measurement.

  There are two operation types in terms of the operation after the set averaging count is reached. In the Continuous type, averaging operation is continued by using the moving average. In the Repeat type, the processing count is reset and averaging operation is performed repeatedly the number of times specified as the averaging count.
- [Parameter] < type > = { CONTinuous | REPeat }
  CONTinuous: Moving average after the averaging count is reached.
  REPeat: Averaging operation is performed repeatedly.
- [Query reply] { CONT | REP }
- [Example] Call ibwrt (analyzer%, ":SEM:AVER:MODE CONT")
- [Relevant commands] [:SENSe<ch>]:SEMask:AVERage:COUNt
  [:SENSe<ch>]:SEMask:AVERage[:STATe]

5.4.127 [:SENSe<ch>]:SEMask:DATA:MODE

- [Command syntax] [:SENSe<ch>]:SEMask:DATA:MODE < type >
  [:SENSe<ch>]:SEMask:DATA:MODE?
- [Function description] Specifies the measurement parameter setting mode in the Spectrum Emission Mask measurement.

  This command is used to specify the setting mode of the parameter used in the Spectrum Emission Mask measurement. There are two modes as shown below:

  Default: Measures in the preset parameter state.

  Manual: Measures in the parameter state that is set before the measurement mode enters into the Spectrum Emission Mask measurement mode.
- [Parameter] < type > = { DEFault | MANual }

  DEFault: Measures by using the parameter values saved by the [:SENSe<ch>]:SEMask:DATA:SAVE command.

  MANual: Measures by using the setting values that are used before the measurement mode enters into the Spectrum Emission Mask measurement mode.
- [Query reply] { DEF | MAN }
- [Example] Call ibwrt (analyzer%, ":SEM:DATA:MODE DEF")
- [Relevant commands] [:SENSe<ch>]:SEMask:DATA:SAVE

5.4.128 [:SENSe<ch>]:SEMask:DATA:SAVE

- [Command syntax] [:SENSe<ch>]:SEMask:DATA:SAVE
- [Function description] Saves the setting parameter in Spectrum Emission Mask measurement

  This command is used to save the current values of the parameter used in the Spectrum Emission Mask measurement as the values used in the Default mode.

  The parameter saved by this command is shown below:
  - All of the parameter set in Ref Power Setup
  - All of the parameter set in Offset Setup
  - RBW value
  - VBW value
  - Sweep time
  - Span frequency
  - Averaging ON or OFF status and the number of times averaging is performed
  - Averaging repeat mode
  - Trace status
  - Trace detector
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":SEM:DATA:SAVE")
- [Relevant command] [:SENSe<ch>]:SEMask:DATA:MODE

5.4.129 [:SENSe<ch>]:SEMask:POWer:LEVel:AUTO

- [Command syntax] [:SENSe<ch>]:SEMask:POWer:LEVel:AUTO
- [Function description] Executes the Auto Level Set function in Spectrum Emission Mask measurement.

  Sets the reference level and ATT to the optimum value according to the signal to be measured. The signal to be measured is set to the optimum value assuming that the WCDMA wave is input.

  After the automatic adjustment is complete, the "RANGing" bit in the standard operation status register is set.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":SEM:POW:LEV:AUTO")

  Call ibwrt (analyzer%, "*WAI")
- [Relevant command] [:SENSe<ch>]:ACP:POWer:LEVel:AUTO

  [:SENSe<ch>]:MCACp:POWer:LEVel:AUTO

5.4.130 [:SENSe<ch>]:FCOunt<screen>:AVERage:COUNt

- [Command syntax] [:SENSe<ch>]:FCOunt<screen>:AVERage:COUNt < int >

  [:SENSe<ch>]:FCOunt<screen>:AVERage:COUNt?
- [Function description] Sets the averaging count in count operation in the frequency counter function

  This command sets the averaging count in count operation when measuring the frequency by the counter. The greater the averaging count, the longer the measurement time becomes, but the measuring accuracy is improved.
- [Parameter] <int> = Averaging count

  Setting range: 2 - 100
- [Query reply] NR1 (Integer)
- [Example] Call ibwrt (analyzer%, ":FCO:AVER:COUN 20")
- [Relevant commands] [:SENSe<ch>]:FCOunt<screen>:AVERage[:STATe]

5.4.131 [:SENSe<ch>]:FCOunt<screen>:AVERage[:STATe]

- [Command syntax] [:SENSe<ch>]:FCOunt<screen>:AVERage[:STATe]
  [:SENSe<ch>]:FCOunt<screen>:AVERage[:STATe]?
- [Function description] Sets the averaging function in count operation to ON or OFF in the frequency counter function

  This command sets the averaging function to ON or OFF when measuring the frequency by the counter. Compared with the mode that does not use the averaging function, the mode using the averaging function takes longer measurement time but the measurement accuracy is improved.
- [Parameter] < bool > = { OFF | ON }
  OFF: Averaging function is set to off.
  ON: Averaging function is set to on.
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":FCO:AVER ON")
- [Relevant commands] [:SENSe<ch>]:FCOunt<screen>:AVERage:COUNt

5.4.132 [:SENSe<ch>]:CCDF:BANDwidth | BWIDth[:RESolution]

- [Command syntax] [:SENSe<ch>]:CCDF:BANDwidth[:RESolution] < real >
  [:SENSe<ch>]:CCDF:BWIDth[:RESolution] < real >
  [:SENSe<ch>]:CCDF:BANDwidth[:RESolution]?
  [:SENSe<ch>]:CCDF:BWIDth[:RESolution]?
- [Function description] Sets the resolution bandwidth (RBW) in the CCDF measurement

  This command sets the resolution bandwidth (RBW). Available RBW values are not sequential. Therefore, if a parameter value that was sent cannot be set as an RBW value, the nearest RBW value available is selected.
- [Parameter] < real > = Resolution bandwidth (MHz/kHz/Hz)
  Setting range when the optional wideband demodulator is not installed:
  100 kHz to 20 MHz (1, 2, 3, and 5 sequences)
  Setting range when the optional wideband demodulator is installed:
  100 kHz to 20 MHz (1, 2, 3, and 5 sequences) and 50 MHz
- [Query reply] NR3 (Real value: Unit Hz)
- [Example] Call ibwrt (analyzer%, ":CCDF:BAND:RES 100KHZ")
- [Relevant commands] [:SENSe<ch>]:CCDF:POINt
  [:SENSe<ch>]:CCDF:GATE
  [:SENSe<ch>]:CCDF:GATE:THReshold

5.4.133 [:SENSe<ch>]:CCDF:POINt

- [Command syntax] [:SENSe<ch>]:CCDF:POINt <int >
  [:SENSe<ch>]:CCDF:POINt?
- [Function description] Sets the number of measurement samples in the CCDF measurement

  This command sets the number of measurement samples in the CCDF measurement.
- [Parameter] <int > = Number of measurement samples
  Setting range: 1 k - 2 G
- [Query reply] NR1 (Interger)
- [Example] Call ibwrt (analyzer%, ":CCDF:POIN 1000")
- [Relevant commands] [:SENSe<ch>]:CCDF:BANDwidth[:RESolution]
  [:SENSe<ch>]:CCDF:BWIDth[:RESolution]
  [:SENSe<ch>]:CCDF:GATE
  [:SENSe<ch>]:CCDF:GATE:THReshold

5.4.134 [:SENSe<ch>]:CCDF:GATE

- [Command syntax] [:SENSe<ch>]:CCDF:GATE <bool>
  [:SENSe<ch>]:CCDF:GATE?
- [Function description] Sets the gate function in the CCDF measurement to ON or OFF

  This command sets the CCDF measurement gate function to ON or OFF.
  If ON is selected, the CCDF measurement is performed when the input signal is larger than the threshold level.
  If OFF is selected, the gate function is canceled.
- [Parameter] <bool> = {OFF | ON}
  OFF: Gate function OFF
  ON: Gate function ON
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":CCDF:GATE ON")
- [Relevant commands] [:SENSe<ch>]:CCDF:BANDwidth[:RESolution]
  [:SENSe<ch>]:CCDF:BWIDth[:RESolution]
  [:SENSe<ch>]:CCDF:POINt
  [:SENSe<ch>]:CCDF:GATE:THReshold

5.4.135 [:SENSe<ch>]:CCDF:GATE:THReshold

- [Command syntax] [:SENSe<ch>]:CCDF:GATE:THReshold <real>
  [:SENSe<ch>]:CCDF:GATE:THReshold?
- [Function description] Sets the threshold level of the gate function in the CCDF measurement
  This command sets the threshold level of the gate function in the CCDF measurement.
- [Parameter] <real> = Threshold level (dB)
- [Query reply] NR3 (Real value: Unit dB)
- [Example] Call ibwrt (analyzer%, ":CCDF:GATE:THR -40DB")
- [Relevant commands] [:SENSe<ch>]:CCDF:BANDwidth[:RESolution]
  [:SENSe<ch>]:CCDF:BWIDth[:RESolution]
  [:SENSe<ch>]:CCDF:POINt
  [:SENSe<ch>]:CCDF:GATE

5.4.136 [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]

- [Command syntax] [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>] <bool>
  [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]?
- [Function description] Sets the window display to ON or OFF in the Multi-Average Power measurement.

  This command sets the windows, in which the Multi-Average Power measurement is performed, to ON or OFF. The Average Power is not measured in windows that are set to OFF.
- [Parameter] < bool > = { OFF | ON }
  OFF: Window display is OFF.
  ON: Window display is ON.
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":MAP:WIND:NUMB1 ON")
- [Relevant commands] [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]:POSition
  [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]:WIDTh

5.4.137 [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]:POSition

- [Command syntax] [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]:POSition < real >
- [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]:POSition?
- [Function description] Specifies the window display position in the Multi-Average Power measurement.

  This command specifies the time position in the window that is used in the Multi-Average Power measurement.
- [Parameter] < real > = Time (s/ms/µs/ns)
- [Query reply] NR3 (Real value: s)
- [Example] Call ibwrt (analyzer%, ":MAP:WIND:POS 400MS")
- [Relevant commands] [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]
  [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]:WIDTh

5.4.138 [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]:WIDTh

- [Command syntax] [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]:WIDTh < real >
  [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]:WIDTh?
- [Function description] Specifies the window display width in the Multi-Average Power measurement.

  This command specifies the width of the window that is used in the Multi-Average Power measurement. The Power in the measurement period specified here is calculated.
- [Parameter] < real > = Time (s/ms/µs/ns)
- [Query reply] NR3 (Real value: s)
- [Example] Call ibwrt (analyzer%, ":MAP:WIND:WIDT 200MS")
- [Relevant commands] [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]
  [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]:POSition

5.4.139 [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]:ACTive

- [Command syntax] [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]:ACTive
  [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]:ACTive?
- [Function description] Specifies the active window in the Multi-Average Power measurement.

  This command sets the window, which is specified in the Multi-Average Power measurement, as the active window. If the specified window is set to OFF, this command sets the window to ON.

  If the Power Ratio function is set to ON, this command specifies the reference measurement window.
- [Parameter] None
- [Query reply] NR1(Integer: window No.)
  1 - 10
- [Example] Call ibwrt (analyzer%, ":MAP:WIND:NUMB1:ACT")
  Call ibwrt (analyzer%, ":MAP:WIND:ACT?")
- [Relevant commands] [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]
  [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]:POSition
  [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]:WIDTh

5.4.140 [:SENSe<ch>]:MAPower:WINDow:RESet

- [Command syntax] [:SENSe<ch>]:MAPower:WINDow:RESet
- [Function description] Sets all windows excluding window No.1 to OFF

  This command sets all windows excluding window No.1 to OFF in the Multi-Average Power measurement.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":MAP:WIND:RES")
- [Relevant command] [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]

5.4.141 [:SENSe<ch>]:MAPower:WINDow:COUPling

- [Command syntax] [:SENSe<ch>]:MAPower:WINDow:COUPling <bool >
  [:SENSe<ch>]:MAPower:WINDow:COUPling?
- [Function description] Sets the window display, which is coupled with the Average Power in the Multi-Average Power measurement, to ON or OFF.

  This command displays the window coupled with the Average Power (Trace) if the window is set to ON in the Multi-Average Power measurement.
- [Parameter] < bool > = { OFF | ON }
  OFF: Coupling display is OFF
  ON: Coupling display is ON
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":MAP:WIND:COUP ON")
- [Relevant commands] [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]
  [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]:POSition
  [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]:WIDTh

5.4.142 [:SENSe<ch>]:MAPower:PRATio

- [Command syntax] [:SENSe<ch>]:MAPower:PRATio <bool>
  [:SENSe<ch>]:MAPower:PRATio?
- [Function description] Sets the Power Ratio measurement to ON or OFF in the Multi-Average Power measurement.

  This command performs the Power Ratio measurement in the Multi-Average Power measurement.
  This command calculates the level difference between the average power in the active window and the average power in other windows that are set to ON.
- [Parameter] < bool > = { OFF | ON }
  ON: The Power Ratio measurement is set to ON
  OFF: The Power Ratio measurement is set to OFF
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":MAP:PRAT ON")
- [Relevant command] [:SENSe<ch>]:MAPower:WINDow[:NUMBer<win>]:ACTive

5.4.143 [:SENSe<ch>]:MAPower:AVERage:COUNt

- [Command syntax] [:SENSe<ch>]:MAPower:AVERage:COUNt < int >
  [:SENSe<ch>]:MAPower:AVERage:COUNt?
- [Function description] Sets the number of times averaging is performed in the Multi-Average Power measurement.

  This command sets the number of times averaging is performed in the Multi-Average Power measurement.

  The sweep repeats for the number of times set in Average Count. When the results have been calculated from the measured values, the AVERaging and MEASuring bits in the standard operation status register are set.
- [Parameter] < int > = Number of times averaging is performed
- [Query reply] NR1(Integer)
- [Example] Call ibwrt (analyzer%, ":MAP:AVER:COUN 5")
- [Relevant commands] [:SENSe<ch>]:MAPower:AVERage[:STATe]
  [:SENSe<ch>]:MAPower:AVERage:MODE

5.4.144 [:SENSe<ch>]:MAPower:AVERage[:STATe]

- [Command syntax] [:SENSe<ch>]:MAPower:AVERage[:STATe] < bool >
  [:SENSe<ch>]:MAPower:AVERage[:STATe]?
- [Function description] Sets the averaging calculation mode to ON or OFF in the Multi-Average Power measurement.

  This command sets the mode, in which the averaging is performed, to ON or OFF in the Multi-Average Power measurement. When the average mode is switched to ON or OFF, the Multi-Average Power calculations that were performed up to then are reset.
- [Parameter] < bool > = { OFF | ON }
  ON: Averaging calculation mode
  OFF: Ordinary calculation mode
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":MAP:AVER:COUN 15")
  Call ibwrt (analyzer%, ":MAP:AVER:STAT ON")
- [Relevant commands] [:SENSe<ch>]:MAPower:AVERage:COUNt
  [:SENSe<ch>]:MAPower:AVERage:MODE

5.4.145 [:SENSe<ch>]:MAPower:AVERage:MODE

- [Command syntax] [:SENSe<ch>]:MAPower:AVERage:MODE < type >
  [:SENSe<ch>]:MAPower:AVERage:MODE?
- [Function description] Specifies the calculation type of the averaging calculation mode in the Multi-Average Power measurement.

  This command specifies the calculation type after the number set in Averaging Count has been reached in the Multi-Average Power measurement.

  There are two types of averaging calculations that can be performed after the number set in Averaging Count is reached. In the "Continuous" calculation, the averaging calculation continues by using the moving average method. In the "Repeat" calculation, the processing count is reset and averaging calculation is performed the number of times specified in Averaging Count.
- [Parameter] < type > = { CONTinuous | REPeat }
  CONTinuous: Moving average is performed after the number of times set in Averaging Count is reached.
  REPeat: The averaging is performed repeatedly.
- [Query reply] { CONT | REP }
- [Example] Call ibwrt (analyzer%, ":MAP:AVER:MODE CONT")
- [Relevant commands] [:SENSe<ch>]:MAPower:AVERage:COUNt
  [:SENSe<ch>]:MAPower:AVERage[:STATe]

5.4.146 [:SENSe<ch>]:MAPower:DATA:MODE

- [Command syntax] [:SENSe<ch>]:MAPower:DATA:MODE < type >
  [:SENSe<ch>]:MAPower:DATA:MODE?
- [Function description] Specifies the setting mode for the measurement parameter in the Multi-Average Power measurement.

  This command specifies the setting modes of the parameters used in the Multi-Average Power measurement. There are two modes as shown below:

  Default: Measures in the preset parameter state.

  Manual: Measures in the parameter state before entering the Multi-Average Power measurement mode.

  For more information on the measurement parameters used here, refer to the function description of the command [:SENSe<ch>]: MAPower:DATA:SAVE, which saves the parameter values used in default mode.
- [Parameter] < type > = { DEFault | MANual }

  DEFault: Performs a measurement by using the parameter values saved by the [:SENSe<ch>]:MAPower:DATA:SAVE command.

  MANual: Performs a measurement by using the setting value before entering the Multi-Average Power measurement mode.
- [Query reply] { DEF | MAN }
- [Example] Call ibwrt (analyzer%, ":MAP:DATA:MODE DEF")
- [Relevant command] [:SENSe<ch>]:MAPower:DATA:SAVE

5.4.147 [:SENSe<ch>]:MAPower:DATA:SAVE

- [Command syntax] [:SENSe<ch>]:MAPower:DATA:SAVE
- [Function description] Saves the function for the measurement parameters in the Multi-Average Power measurement.

  This command saves the current values of the parameters, which are used in the Multi-Average Power measurement, as the values used in default mode.

  The parameters saved by this command are shown below:
  - ON/OFF status of the Power Ratio measurement
  - ON/OFF status and the window position or width of each window
  - Active window No.
  - ON/OFF status of Couple to Power
  - RBW value
  - VBW value
  - Sweep time
  - Span frequency
  - ON/OFF status of Averaging and the number of times averaging is performed
  - Averaging repeat mode
  - Trace status
  - Trace detector
  - Trigger and trigger delay value
  - Gated sweep
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":MAP:DATA:SAVE")
- [Relevant command] [:SENSe<ch>]:MAPower:DATA:MODE

5.5 Configure Commands

This section describes the Configure subsystem.

In the Configure subsystem, the commands used for entry into various measurement modes are defined.

Command	Function	Reference Page
:CONFigure<ch>		
:CPOWer<screen>	Sets the Channel Power measurement mode	5-132
:APOWer<screen>	Sets the Average Power measurement mode	5-133
:OBW<screen>	Sets the OBW measurement mode	5-133
:MCACp	Sets the Multi Carrier ACP measurement mode	5-134
:ACP	Sets the ACP measurement mode	5-134
:SPURious	Sets the Spurious measurement mode	5-135
:SEMask	Sets the Spectrum Emission Mask measurement mode	5-135
:IM	Sets the IM measurement mode	5-136
:HARMonics	Sets the Harmonic measurement mode	5-136
:CCDF	Sets the CCDF measurement mode	5-137
:MAPower	Sets the Multi-Average Power measurement mode	5-137
:NORMal	Terminates each measurement mode	5-138

5.5.1 :CONFigure<ch>:CPOWer<screen>

- [Command syntax] :CONFigure<ch>:CPOWer<screen>
- [Function description] Sets the Channel Power measurement mode

  This command sets the measurement mode to the Channel Power measurement mode. If any other measurement mode is used, the measurement mode automatically switches to the Channel Power measurement mode.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CONF:CPOW")
- [Relevant commands] :FETCh<ch>:CPOWer<screen>?
  :MEASure<ch>:CPOWer<screen>?
  :READ<ch>:CPOWer<screen>?

5.5.2 :CONFigure<ch>:APOWer<screen>

- [Command syntax] :CONFigure<ch>:APOWer<screen>
- [Function description] Sets the Average Power measurement mode

  This command sets the measurement mode to the Average Power measurement mode. If any other measurement mode is used, the measurement mode automatically switches to the Average Power measurement mode.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CONF:APOW")
- [Relevant commands] :FETCh<ch>:APOWer<screen>?
  :MEASure<ch>:APOWer<screen>?
  :READ<ch>:APOWer<screen>?

5.5.3 :CONFigure<ch>:OBW<screen>

- [Command syntax] :CONFigure<ch>:OBW<screen>
- [Function description] Sets the OBW measurement mode

  This command sets the measurement mode to the Occupied Bandwidth (OBW) measurement mode.

  If any other measurement mode is used, the measurement mode automatically switches to the OBW measurement mode.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CONF:OBW")
- [Relevant commands] :FETCh<ch>:OBW<screen>?
  :MEASure<ch>:OBW<screen>?
  :READ<ch>:OBW<screen>?

5.5.4 :CONFigure<ch>:MCACp

- [Command syntax] :CONFigure<ch>:MCACp
- [Function description] Sets the Multi Carrier ACP measurement mode

  This command sets the measurement mode to the Multi Carrier ACP measurement mode.

  If any other measurement mode is used, the measurement mode automatically switches to the Multi Carrier ACP measurement mode.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CONF:MCAC")
- [Relevant commands] :FETCh<ch>: MCACp[:NUMBer{1|2|3|4|5|6}]?

  :MEASure<ch>: MCACp[:NUMBer{1|2|3|4|5|6}]?

  :READ<ch>: MCACp[:NUMBer{1|2|3|4|5|6}]?

5.5.5 :CONFigure<ch>:ACP

- [Command syntax] :CONFigure<ch>:ACP
- [Function description] Sets the ACP measurement mode

  This command sets the measurement mode to the Adjacent Channel Leakage Power (ACP) measurement mode.

  If any other measurement mode is used, the measurement mode automatically switches to the ACP measurement mode.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CONF:ACP")
- [Relevant commands] :FETCh<ch>: ACP[:NUMBer{1|2|3|4|5}]?

  :MEASure<ch>: ACP[:NUMBer{1|2|3|4|5}]?

  :READ<ch>: ACP[:NUMBer{1|2|3|4|5}]?

5.5.6 :CONFigure<ch>:SPURious

- [Command syntax] :CONFigure<ch>:SPURious
- [Function description] Sets the Spurious measurement mode

  This command sets the measurement mode to the Spurious measurement mode.

  If any other measurement mode is used, the measurement mode automatically switches to the Spurious measurement mode.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CONF:SPUR")
- [Relevant commands] :FETCh<ch>:SPURious[:NUMBer{1|2|3|...|14|15}]?
  :MEASure<ch>:SPURious[:NUMBer{1|2|3|...|14|15}]?
  :READ<ch>:SPURious[:NUMBer{1|2|3|...|14|15}]?

5.5.7 :CONFigure<ch>:SEMask

- [Command syntax] :CONFigure<ch>:SEMask
- [Function description] Sets the Spectrum Emission Mask measurement mode

  This command sets the measurement mode to the Spectrum Emission Mask measurement mode.

  If any other measurement mode is used, the measurement mode automatically switches to the Spectrum Emission Mask measurement mode.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CONF:SEM")
- [Relevant commands] :FETCh<ch>:SEMask[:NUMBer{1|2|3|4|5}]?
  :MEASure<ch>:SEMask[:NUMBer{1|2|3|4|5}]?
  :READ<ch>:SEMask[:NUMBer{1|2|3|4|5}]?

5.5.8 :CONFigure<ch>:IM

- [Command syntax] :CONFigure<ch>:IM
- [Function description] Sets the IM measurement mode

  This command sets the measurement mode to the Intermodulation Distortion (IM) measurement mode.

  If any other measurement mode is used, the measurement mode automatically switches to the Intermodulation Distortion measurement mode.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CONF:IM")
- [Relevant commands] :FETCh<ch>:IM[:NUMBer{1|3|5|7|9}]?
  :MEASure<ch>:IM[:NUMBer{1|3|5|7|9}]?
  :READ<ch>:IM[:NUMBer{1|3|5|7|9}]?

5.5.9 :CONFigure<ch>:HARMonics

- [Command syntax] :CONFigure<ch>:HARMonics
- [Function description] Sets the harmonic measurement mode

  This command sets the measurement mode to the Harmonic measurement mode.

  If any other measurement mode is used, the measurement mode automatically switches to the Harmonic measurement mode.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CONF:HARM")
- [Relevant commands] :FETCh<ch>:HARMonics[:NUMBer{2|3|4|5|6|7|8|9|10}]?
  :MEASure<ch>:HARMonics[:NUMBer{2|3|4|5|6|7|8|9|10}]?
  :READ<ch>:HARMonics[:NUMBer{2|3|4|5|6|7|8|9|10}]?

5.5.10 :CONFigure<ch>:CCDF

- [Command syntax] :CONFigure<ch>:CCDF
- [Function description] Sets the CCDF measurement mode

  This command sets the measurement mode to the CCDF measurement mode.

  If any other measurement mode is used, the measurement mode automatically switches to the CCDF measurement mode.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CONF:CCDF")
- [Relevant commands] :FETCh<ch>:CCDF[:NUMBer{1|2|3|4|5|6}]?

  :MEASure<ch>:CCDF[:NUMBer{1|2|3|4|5|6}]?

  :READ<ch>:CCDF[:NUMBer{1|2|3|4|5|6}]?

5.5.11 :CONFigure<ch>:MAPower

- [Command syntax] :CONFigure<ch>:MAPower
- [Function description] Sets the Multi-Average Power measurement mode.

  This command sets the measurement mode to the Multi-Average Power measurement mode.

  If any other measurement mode is used, the measurement mode automatically switches to the Multi-Average Power measurement mode.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CONF:MAP")
- [Relevant commands] :FETCh<ch>:MAPower[:NUMBer<win>]?

  :MEASure<ch>: MAPower[:NUMBer<win>]?

  :READ<ch>:MAPower[:NUMBer<win>]?

  :FETCh<ch>:MAPower:RMS[:NUMBer<win>]?

  :MEASure<ch>: MAPower:RMS[:NUMBer<win>]?

  :READ<ch>:MAPower:RMS[:NUMBer<win>]?

5.5.12 :CONFigure<ch>:NORMal

- [Command syntax] :CONFigure<ch>:NORMal
- [Function description] Terminates each measurement mode

  This command terminates the current measurement mode and sets the normal SA mode.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CONF:NORM")
- [Relevant commands]

5.6 Measure /Read /Fetch Command

This section describes the Measure, Read, and Fetch subsystems.

This section explains the Measure command as an example.The explanation applies to the Read and Fetch commands by replacing the MEASure command header part with READ or FETCh.

MEMO: ***There is no difference in the reply format of the Measure, Read, and Fetch commands. Differences between these commands: When measurement is required to be performed, the Measure or Read commands are used, and when the result data is simply read, the Fetch command is used. Both the Measure command and the Read command perform measurement.However, initialization when entering the measurement mode differs depending on the measurement. The difference is explained in the Function Description item. The same operation applies if no special explanations are given. If the Fetch command is issued without entering the corresponding measurement mode, a Query error occurs.***

<table>
<tr><th>Command</th><th>Function</th><th>Reference Page</th></tr>
<tr><td>:MEASure<ch></td><td></td><td></td></tr>
<tr><td>:CPOWer<screen>?</td><td>Channel Power measurement execution and measurement result (Trace) reading</td><td>5-143</td></tr>
<tr><td>:PDENsity?</td><td>Channel Power measurement execution and average power density (Trace) reading</td><td>5-143</td></tr>
<tr><td>:RMS?</td><td>Channel Power measurement execution and measurement result (RMS) reading</td><td>5-144</td></tr>
<tr><td>:PDENsity?</td><td>Channel Power measurement execution and average power density (RMS) reading</td><td>5-144</td></tr>
<tr><td>:APOWer<screen>?</td><td>Average Power measurement execution and measurement result (Trace) reading</td><td>5-145</td></tr>
<tr><td>:PDENsity?</td><td>Average Power measurement execution and average power density (Trace) reading</td><td>5-145</td></tr>
<tr><td>:RMS?</td><td>Average Power measurement execution and measurement result (RMS) reading</td><td>5-146</td></tr>
<tr><td>:PDENsity?</td><td>Average Power measurement execution and average power density (RMS) reading</td><td>5-146</td></tr>
<tr><td>:MEASure<ch></td><td></td><td></td></tr>
<tr><td>:ACP</td><td></td><td></td></tr>
<tr><td>[:NUMBer{1|2|3|4|5}?</td><td>ACP measurement execution and all measurement results reading</td><td>5-149</td></tr>
<tr><td>:RPOWer?</td><td>ACP measurement execution and reference power measurement result reading</td><td>5-150</td></tr>
<tr><td>:UPPer</td><td></td><td></td></tr>
<tr><td>[:NUMBer{1|2|3|45}]?</td><td>ACP measurement execution and reading of all measurement results of the specified channels on the Upper side</td><td>5-151</td></tr>
<tr><td>:LOWer</td><td></td><td></td></tr>
<tr><td>[:NUMBer{1|2|3|4|5}]?</td><td>ACP measurement execution and reading of all measurement results of the specified channels on the Lower side</td><td>5-152</td></tr>
</table>

Command	Function	Reference Page
:MEASure<ch> :MCACp [:NUMBer{1|2|3|4|5|6}]?	Multi Carrier ACP measurement execution and measurement result reading	5-153
:CPOWer [:NUMBer{1|2|...|9|10}]?	Multi Carrier ACP measurement execution and carrier power value reading	5-154
:MEASure<ch> :OBW<screen>?	OBW measurement execution and all measurement results reading	5-147
:OBW?	OBW measurement execution and measurement result reading (only OBW value)	5-147
:FCENter?	OBW measurement execution and measurement result reading (only OBW center frequency)	5-148
:MEASure<ch> :SPURious [:NUMBer{1|2...|14|15}]?	Spurious measurement execution and all measurement results reading	5-155
:MEASure<ch> :SEMask [:NUMBer{1|2|3|4|5}]?	Spectrum Emission Mask measurement execution and measurement result reading	5-157
:RPOWer?	Spectrum Emission Mask measurement execution and reference power result reading	5-158
:FAIL?	Spectrum Emission Mask measurement execution and overall Pass/Fail result reading	5-159

Command	Function	Reference Page
:MEASure<ch>		
:IM		
[:NUMBer{1|3|5|7|9}]?	IM measurement execution and measurement result reading	5-159
:REFerence?	IM measurement execution and reference frequency data reading	5-161
:DELTa?	IM measurement execution and reading of the frequency difference between two signals	5-161
:UPPer		
[:NUMBer{1|3|5|7|9}]?	IM measurement execution and reading of the specified-order modulation distortion measurement result	5-163
:LOWer		
[:NUMBer{1|3|5|7|9}]?	IM measurement execution and reading of the specified-order modulation distortion measurement result	5-164
:IP3?	IM measurement execution and 3rd order intercept point value reading	5-162
:IPOint		
[:NUMBer{1|3|5|7|9}]?	IM measurement execution and intercept point value reading	5-163
:MEASure<ch>		
:HARMonics		
[:NUMBer{2|3|...|9|10}]?	Harmonic measurement execution and all measurement results reading	5-166
:FUNDamental?	Harmonic measurement execution and fundamental wave measurement result reading	5-167
:MEASure		
:CCDF		
[:NUMBer{1|2|3|4|5|6}]?	CCDF measurement execution and the measurement result reading	5-168
:PFACtor?	CCDF measurement execution and Peak Factor reading	5-169
:APOWer?	CCDF measurement execution and Average Power reading	5-169
:PRATio		
[:NUMBer{1|2|3|4|5|6}]?	CCDF measurement execution and the power ratio reading	5-170

Command	Function	Reference Page
:MEASure<ch>		
:MAPower		
[:NUMBer{1|2...|9|10}]?	Multi-Average Power measurement execution and the measurement result (Trace) reading	5-171
:PDENsity		
[:NUMBer{1|2...|9|10}]?	Multi-Average Power measurement execution and the average power density (Trace) reading	5-172
:PRATio		
[:NUMBer{1|2...|9|10}]?	Multi-Average Power measurement execution and Power Ratio (Trace) reading	5-173
:RMS		
[:NUMBer{1|2...|9|10}]?	Multi-Average Power measurement execution and the measurement result (RMS) reading	5-174
:PDENsity		
[:NUMBer{1|2...|9|10}]?	Multi Average Power measurement execution and the average power density (RMS) reading	5-175
:PRATio		
[:NUMBer{1|2...|9|10}]?	Multi-Average Power measurement execution and Power Ratio (RMS) reading	5-176

5.6.1 :MEASure<ch>:CPOWer<screen>?

- [Command syntax] :MEASure<ch>:CPOWer<screen>?
- [Function description] Channel Power measurement execution and measurement result (Trace) reading

  Executes Channel Power measurement and returns the measurement result after completion of the measurement.
- [Parameter] None
- [Query reply] NR3 (Real value Channel Power value: Unit dBm)
- [Example]
  ```
  Result$ = Space$(1024)
  Call ibwrt (analyzer%, ":MEAS:CPOW?")
  Call ibrd (analyzer%, Result$)
  ```
- [Relevant commands] :UNIT<ch>:POWer<screen>
  :READ<ch>:CPOWer<screen>?
  :FETCh<ch>:CPOWer<screen>?

5.6.2 :MEASure<ch>:CPOWer<screen>:PDENsity?

- [Command syntax] :MEASure<ch>:CPOWer<screen>:PDENsity?
- [Function description] Channel Power measurement execution and average power density (Trace) reading

  Channel Power measurement execution and average power density (Trace) reading
- [Parameter] None
- [Query reply] NR3 (Real value: Unit dBm/Hz or dBμV/√Hz)
- [Example]
  ```
  Result$ = Space$(1024)
  Call ibwrt (analyzer%, ":MEAS:CPOW:PDEN?")
  Call ibrd (analyzer%, Result$)
  ```
- [Relevant commands] :UNIT<ch>:POWer<screen>
  :READ<ch>:CPOWer<screen>:PDENsity?
  :FETCh<ch>:CPOWer<screen>:PDENsity?

5.6.3 :MEASure<ch>:CPOWer<screen>:RMS?

- [Command syntax] :MEASure<ch>:CPOWer<screen>:RMS?
- [Function description] Channel Power measurement execution and measurement result (RMS) reading

  Executes Channel Power measurement and returns the measurement result after completion of the measurement.
- [Parameter] None
- [Query reply] NR3 (Real value Channel Power value: Unit dBm)
- [Example] Result$ = Space$(1024)

  Call ibwrt (analyzer%, ":MEAS:CPOW:RMS?")

  Call ibrd (analyzer%, Result$)
- [Relevant commands] :UNIT<ch>:POWer<screen>

  :READ<ch>:CPOWer<screen>:RMS?

  :FETCh<ch>:CPOWer<screen>:RMS?

5.6.4 :MEASure<ch>:CPOWer<screen>:RMS:PDENsity?

- [Command syntax] :MEASure<ch>:CPOWer<screen>:RMS:PDENsity?
- [Function description] Channel Power measurement execution and average power density (RMS) reading

  Executes Channel Power measurement and returns the average power density after completion of the measurement.
- [Parameter] None
- [Query reply] NR3 (Real value: Unit dBm/Hz or dBμV/√Hz)
- [Example] Result$ = Space$(1024)

  Call ibwrt (analyzer%, ":MEAS:CPOW:RMS:PDEN?")

  Call ibrd (analyzer%, Result$)
- [Relevant commands] :UNIT<ch>:POWer<screen>

  :READ<ch>:CPOWer<screen>:RMS:PDENsity?

  :FETCh<ch>:CPOWer<screen>:RMS:PDENsity?

5.6.5 :MEASure<ch>:APOWer<screen>?

- [Command syntax] :MEASure<ch>:APOWer<screen>?
- [Function description] Average Power measurement execution and measurement result (Trace) reading

  Executes Average Power measurement and returns the measurement result after completion of the measurement.
- [Parameter] None
- [Query reply] NR3 (Real value Average Power value: Unit dBm)
- [Example] Result$ = Space$(1024)
  Call ibwrt (analyzer%, ":MEAS:APOW?")
  Call ibrd (analyzer%, Result$)
- [Relevant commands] :UNIT<ch>:POWer<screen>
  :READ<ch>:APOWer<screen>?
  :FETCh<ch>:APOWer<screen>?

5.6.6 :MEASure<ch>:APOWer<screen>:PDENsity?

- [Command syntax] :MEASure<ch>:APOWer<screen>:PDENsity?
- [Function description] Average Power measurement execution and average power density (Trace) reading

  Executes Average Power measurement and returns the average power density after completion of the measurement.
- [Parameter] None
- [Query reply] NR3 (Real value: Unit dBm/Hz or dBμV/√Hz)
- [Example] Result$ = Space$(1024)
  Call ibwrt (analyzer%, ":MEAS:APOW:PDEN?")
  Call ibrd (analyzer%, Result$)
- [Relevant commands] :UNIT<ch>:POWer<screen>
  :READ<ch>:APOWer<screen>:PDENsity?
  :FETCh<ch>:APOWer<screen>:PDENsity?

5.6.7 :MEASure<ch>:APOWer<screen>:RMS?

- [Command syntax] :MEASure<ch>:APOWer<screen>:RMS?
- [Function description] Average Power measurement execution and measurement result (RMS) reading

  Executes Average Power measurement and returns the measurement result after completion of the measurement.
- [Parameter] None
- [Query reply] NR3 (Real value Average Power value: Unit dBm)
- [Example] Result$ = Space$(1024)
  Call ibwrt (analyzer%, ":MEAS:APOW:RMS?")
  Call ibrd (analyzer%, Result$)
- [Relevant commands] :UNIT<ch>:POWer<screen>
  :READ<ch>:APOWer<screen>:RMS?
  :FETCh<ch>:APOWer<screen>:RMS?

5.6.8 :MEASure<ch>:APOWer<screen>:RMS:PDENsity?

- [Command syntax] :MEASure<ch>:APOWer<screen>:RMS:PDENsity?
- [Function description] Average Power measurement execution and average power density (RMS) reading

  Executes Average Power measurement and returns the average power density after completion of the measurement.
- [Parameter] None
- [Query reply] NR3 (Real value: Unit dBm/Hz or dBμV/√Hz)
- [Example] Result$ = Space$(1024)
  Call ibwrt (analyzer%, ":MEAS:APOW:RMS:PDEN?")
  Call ibrd (analyzer%, Result$)
- [Relevant commands] :UNIT<ch>:POWer<screen>
  :READ<ch>:APOWer<screen>:RMS:PDENsity?
  :FETCh<ch>:APOWer<screen>:RMS:PDENsity?

5.6.9 :MEASure<ch>:OBW<screen>?

- [Command syntax] :MEASure<ch>:OBW<screen>?
- [Function description] OBW measurement execution and all measurement results reading

  Executes Occupied Bandwidth (OBW) measurement and returns the measurement result after completion of the measurement.
- [Parameter] None
- [Query reply] NR3, NR3
  (Real value OBW center frequency: Unit Hz, Real value OBW: Unit Hz)
- [Example] Result$ = Space$(1024)
  Call ibwrt (analyzer%, ":MEAS:OBW?")
  Call ibrd (analyzer%, Result$)
- [Relevant commands] :MEASure<ch>:OBW<screen>:OBW?
  :MEASure<ch>:OBW<screen>:FCENter?
  :READ<ch>:OBW<screen>?
  :FETCh<ch>:OBW<screen>?

5.6.10 :MEASure<ch>:OBW<screen>:OBW?

- [Command syntax] :MEASure<ch>:OBW<screen>:OBW?
- [Function description] OBW measurement execution and measurement result reading (only OBW value)

  Executes Occupied Bandwidth (OBW) measurement and returns the measurement result of the OBW value after completion of the measurement.
- [Parameter] None
- [Query reply] NR3 (Real value OBW: Unit Hz)
- [Example] Result$ = Space$(1024)
  Call ibwrt (analyzer%, ":MEAS:OBW:OBW?")
  Call ibrd (analyzer%, Result$)
- [Relevant commands] :MEASure<ch>:OBW<screen>?
  :MEASure<ch>:OBW<screen>:FCENter?
  :READ<ch>:OBW<screen>:OBW?
  :FETCh<ch>:OBW<screen>:OBW?

5.6.11 :MEASure<ch>:OBW<screen>:FCENter?

- [Command syntax] :MEASure<ch>:OBW<screen>:FCENTer?
- [Function description] OBW measurement execution and measurement result reading (only OBW center frequency)

  Executes Occupied Bandwidth (OBW) measurement and returns the measurement result of the center frequency value of the occupied band-width (OBW center frequency) after completion of the measurement.
- [Parameter] None
- [Query reply] NR3 (Real value OBW center frequency: Unit Hz)
- [Example] Result$ = Space$(1024)
  Call ibwrt (analyzer%, ":MEAS:OBW:FCEN?")
  Call ibrd (analyzer%, Result$)
- [Relevant commands] :MEASure<ch>:OBW<screen>?
  :MEASure<ch>:OBW<screen>:OBW?
  :READ<ch>:OBW<screen>:FCENter?
  :FETCh<ch>:OBW<screen>:FCENter?

5.6.12 :MEASure<ch>:ACP[:NUMBer{1|2|3|4|5}]?

- [Command syntax] :MEASure<ch>:ACP[:NUMBer{1|2|3|4|5}]?
- [Function description] ACP measurement execution and all measurement results reading

  Executes adjacent channel leakage power (ACP) measurement and returns the measurement result after completion of the measurement.

  With regard to the measurement result, when the NUMBer header is omitted, the reference power value and the data for the number of channels set before measurement are output in the order from the Lower side to the Upper side by each channel.

  When the adjacent channel number to be output is specified by the NUMBer header, the reference power value and the ACP value for the specified adjacent channel number are output in the order from the Lower side to the Upper side.
- [Parameter] None
- [Query reply] When the NUMBer header is omitted:

  NR3, NR3, NR3[, [NR3, NR3], ..., [NR3, NR3]]

  Output order: Real value reference power: Unit dBm,
  Real value lower level(1): Unit dB,
  Real value upper level(1): Unit dB
  [,[Real value lower level(2): Unit dB,
  Real value upper level(2): Unit dB],
  ...,
  Real value lower level(n): Unit dB,
  Real value upper level(n): Unit dB]]

  n: Number of channels to be measured set before ACP measurement (up to 5 groups)

  When the NUMBer header is specified:

  NR3, NR3, NR3

  Output order: Real value reference power: Unit dBm,
  Real value lower level(m): Unit dB,
  Real value upper level(m): Unit dB,

  m: The number representing the specified adjacent channel
- [Example]

```
Result$ = Space$(1024)
Call ibwrt (analyzer%, ":MEAS:ACP?")
Call ibrd(analyzer%, Result$)
```

- [Relevant commands] :UNIT<ch>:POWer<screen>
  :MEASure<ch>:ACP:LOWer[:NUMBer{1|2|3|4|5}]?
  :MEASure<ch>:ACP:UPPer[:NUMBer{1|2|3|4|5}]?
  :MEASure<ch>:ACP:RPOWer?
  :READ<ch>:ACP[:NUMBer{1|2|3|4|5}]?
  :FETCh<ch>:ACP[:NUMBer{1|2|3|4|5}]?

5.6.13 :MEASure<ch>:ACP:RPOWer?

- [Command syntax] :MEASure<ch>:ACP:RPOWer?
- [Function description] ACP measurement execution and reference power measurement result reading

  Executes adjacent channel leakage power (ACP) measurement and returns the reference power value after completion of the measurement.
- [Parameter] None
- [Query reply] NR3 (Real value reference power: Unit dBm)
- [Example] Result$ = Space$(1024)
  Call ibwrt (analyzer%, ":MEAS:ACP:RPOW?")
  Call ibrd(analyzer%, Result$)
- [Relevant commands] :UNIT<ch>:POWer<screen>
  :MEASure<ch>:ACP[:NUMBer{1|2|3|4|5}]?
  :MEASure<ch>:ACP:LOWer[:NUMBer{1|2|3|4|5}]?
  :MEASure<ch>:ACP:UPPer[:NUMBer{1|2|3|4|5}]?
  :READ<ch>:ACP:RPOWer?
  :FETCh<ch>:ACP:RPOWer?

5.6.14 :MEASure<ch>:ACP:UPPer[:NUMBer{1|2|3|4|5}]?

- [Command syntax] :MEASure<ch>:ACP:UPPer[:NUMBer{1|2|3|4|5}]?
- [Function description] ACP measurement execution and reading of all measurements results of the specified channels on the Upper side

  Executes adjacent channel leakage power (ACP) measurement and returns the leakage power value of the specified channels on the Upper side (the higher-frequency side than the carrier frequency) after completion of the measurement.

  When the NUMBer header is omitted, the data from the target channels, which are set before measurement, is output.

  When the target adjacent channel numbers are specified in the NUMBer header, the upper side leakage power values in the specified adjacent channels are returned.
- [Parameter] None
- [Query reply] When the NUMBer header is omitted:

  NR3[, NR3,..., NR3] (Real value Upper Channel: Unit dB)

  Output order: Real value upper level(1): Unit dB
  [,Real value upper level(2): Unit dB
  ,...
  ,Real value upper level(n): Unit dB]

  n: Number of channels to be measured set before ACP measurement (up to 5 groups)

  When the NUMBer header is specified:

  NR3 (Real value Upper Channel level{1|2|3|4|5}: Unit dB)
- [Example]

  ```
  Result$ = Space$(1024)
  Call ibwrt (analyzer%, ":MEAS:ACP:UPP:NUMB1?")
  Call ibrd(analyzer%, Result$)
  ```
- [Relevant commands]

  :MEASure<ch>:ACP[:NUMBer{1|2|3|4|5}]?
  :MEASure<ch>:ACP:LOWer[:NUMBer{1|2|3|4|5}]?
  :MEASure<ch>:ACP:RPOWer?
  :READ<ch>:ACP:UPPer[:NUMBer{1|2|3|4|5}]?
  :FETCh<ch>:ACP:UPPer[:NUMBer{1|2|3|4|5}]?

5.6.15 :MEASure<ch>:ACP:LOWer[:NUMBer{1|2|3|4|5}]?

- [Command syntax] :MEASure<ch>:ACP:LOWer[:NUMBer{1|2|3|4|5}]?
- [Function description] ACP measurement execution and reading of all measurement results of the specified channels on the Lower side

  Executes adjacent channel leakage power (ACP) measurement and returns the leakage power value of the specified channels on the Lower side (the lower-frequency side than the carrier frequency) after completion of the measurement.

  When the NUMBer header is omitted, the data from the target channels, which are set before measurement, is output.

  When the target adjacent channel numbers are specified in the NUMBer header, the lower side leakage power values in the specified adjacent channels are returned.
- [Parameter] None
- [Query reply] When the NUMBer header is omitted:

  NR3[, NR3,..., NR3] (Real values Lower Channel: Unit dB)

  Output order: Real value lower level(1): Unit dB
  [,Real value lower level(2): Unit dB
  ,...
  ,Real value lower level(n): Unit dB]

  n: Number of channels to be measured set before ACP measurement (up to 5 groups)

  When the NUMBer header is specified:

  NR3 (Real value Lower Channel level{1|2|3|4|5}: Unit dB)
- [Example]

```
Result$ = Space$(1024)
Call ibwrt (analyzer%, ":MEAS:ACP:LOW:NUMB1?")
Call ibrd(analyzer%, Result$)
```

- [Relevant commands]

  :MEASure<ch>:ACP[:NUMBer{1|2|3|4|5}?

  :MEASure<ch>:ACP:UPPer[:NUMBer{1|2|3|4|5}?

  :MEASure<ch>:ACP:RPOWer?

  :READ<ch>:ACP:LOWer[:NUMBer{1|2|3|4|5}]?

  :FETCh<ch>:ACP:LOWer[:NUMBer{1|2|3|4|5}]?

5.6.16 :MEASure<ch>:MCACp[:NUMBer{1|2|3|4|5|6}]?

- [Command syntax] :MEASure<ch>:MCACp[:NUMBer{1|2|3|4|5|6}]?
- [Function description] Multi Carrier ACP measurement execution and all measurement results reading

  Executes Multi Carrier ACP measurement and reads all measurement results after completion of the measurement.

  With regard to the result data to be read, the reference power value at a specified carrier frequency, the leakage power value at a specified adjacent channel frequency, and the Pass/Fail result data comprise one set, and up to 6 data sets are output.

  When the NUMBer header is used, only the result data of the adjacent channel with the specified number is output.
- [Parameter] None
- [Query reply] When the NUMBer header is omitted:

  NR3,NR3, NR1[, [NR3, NR3, NR1], ..., [NR3, NR3, NR1]]

  Output order: Reference power(1): Unit dBm,
  ACP level(1): Unit dB,
  Pass/Fail(1): 0/1
  [,[Reference power(2),
  ACP level(2) ,
  Pass/Fail(2)],
  ...,
  [Reference power(n),
  [ACP level(n),
  Pass/Fail:(n)]]

  n: Number of channels to be measured set before multi carrier power measurement (up to 6 groups)

  When the NUMBer header is specified:

  Output order: Reference power(m): Unit dBm,
  ACP level(m): Unit dB,
  Pass/Fail(m): 0/1,

  m: Specified adjacent channel number
- [Example]

```
Result$ = Space$(1024)
Call ibwrt (analyzer%, ":MEAS:MCAC:NUMB1?")
Call ibrd(analyzer%, Result$)
```

- [Relevant commands] :UNIT<ch>:POWer<screen>
  :MEASure<ch>:MCACp:CPOWer[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?
  :READ<ch>:MCACp[:NUMBer{1|2|3|4|5|6}]?
  :FETCh<ch>:MCACp[:NUMBer{1|2|3|4|5|6}]?

5.6.17 :MEASure<ch>:MCACp:CPOWer[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

- [Command syntax] :MEASure<ch>:MCACp:CPOWer[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?
- [Function description] Multi Carrier ACP measurement execution and specified carrier power value reading

  Executes Multi Carrier ACP measurement and returns the carrier power value at the specified frequency after completion of the measurement.

  When the NUMBer header is omitted, the carrier power measurement results for the number of carriers set are output.

  When a carrier is specified by the NUMBer header, the carrier power measurement result of the specified carrier number is output.
- [Parameter] None
- [Query reply] When the NUMBer header is omitted:

  NR3 [, NR3, ..., NR3]
  (All real values Carrier Power: Unit dBm)

  Output order:
  Carrier Power(1) : Unit dBm
  [,Carrier Power(2) : Unit dBm
  ,...
  ,Carrier Power(n): Unit dBm]

  n: Number of carrier signals set before measurement (up to 10)

  When specified by the NUMBer header:
  NR3 (Real value Carrier Power value: Unit dBm)
- [Example] Result$ = Space$(1024)
  Call ibwrt (analyzer%, ":MEAS:MCAC:CPOW:NUMB2?")
  Call ibrd(analyzer%, Result$)
- [Relevant commands] :UNIT<ch>:POWer<screen>
  :MEASure<ch>:MCACp[:NUMBer{1|2|3|4|5|6}]?
  :READ<ch>:MCACp:CPOWer[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?
  :FETCh<ch>:MCACp:CPOWer[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

5.6.18 :MEASure<ch>:SPURious[:NUMBer{1|2|3|...|14|15}]?

- [Command syntax] :MEASure<ch>:SPURious[:NUMBer{1|2|3|...|14|15}?
- [Function description] Spurious measurement execution and all measurement results reading

  Executes Spurious measurement and outputs the measurement result data after completion of the measurement. In Spurious measurement, the frequency, level, and Pass/Fail judgment value comprise one set. In the measurement area of up to 15 groups, up to 10 data sets are output for one group. The number of data sets to be output in each measurement area varies depending on the measurement. However, at least one data set having the maximum value in each area is returned.

  Besides parameters set for the table, it is necessary to set Peak Delta Y, used for peak point judgment, appropriately as a parameter required for Spurious search.

  When the NUMBer header is omitted, all measurement results data are output.

  When the NUMBer header is specified, the result data of measurement performed based on the measurement condition in the table corresponding to the specified number is output.
- [Parameter] None
- [Query reply] When the NUMBer header is omitted:

  NR3, NR3, NR1 [, [NR3, NR3, NR1], [NR3, NR3, NR1], ...,
  [NR3, NR3, NR1]]

  Output order: Freq(11): Unit Hz,
  Level(11): Unit dBm,
  P/F(11):0/1
  [,[Freq(12), Level(12), P/F(12)],
  ...,
  [Freq(21), Level(21), P/F(21)],
  ...,
  [Freq(nm), Level(nm), P/F(nm)]]

  n: Measurement area number in the Spurious table: Up to 15

  m: Number of data items detected as spurious in one measurement area: Up to 10

  When the NUMBer header is specified:

  NR3, NR3, NR1 [, [NR3, NR3, NR1], ...,
  [NR3, NR3, NR1]]

5.6.18 :MEASure<ch>:SPURious[:NUMBer{1|2|3|...|14|15}]?

Output order: Freq(n1): Unit Hz,
Level(n1): Unit dBm,
P/F(n1): 0/1
[,[Freq(n2), Level(n2), P/F(n2)],
...,
[Freq(nm), Level(nm), P/F(nm)]]

n: Measurement area number in the Spurious table: 1 to 15

m: Number of data items detected as spurious in one measurement area: Up to 10

- [Example]

```
Result$ = Space$(1024)
Call ibwrt (analyzer%, ":MEAS:SPUR:NUMB1?")
Call ibrd(analyzer%, Result$)
```

- [Relevant commands]

:UNIT<ch>:POWer<screen>
:CALCulate<ch>:MARKer<screen>:MAXimum:DELTa
:READ<ch>:SPURious[:NUMBer{1|2|3|...|14|15}]?
:FETCh<ch>:SPURious[:NUMBer{1|2|3|...|14|15}]?

5.6.19 :MEASure<ch>:SEMask[:NUMBer{1|2|3|4|5}]?

- [Command syntax] :MEASure<ch>:SEMask[:NUMBer{1|2|3|4|5}]?
- [Function description] Spectrum Emission Mask measurement execution and measurement result reading

  Executes Spectrum Emission Mask measurement and outputs the measurement result data after completion of the measurement. In Spectrum Emission Mask measurement, measurement areas of up to 5 groups, above and below the center carrier frequency, wait for the measurement results at one time. The maximum-level frequency data in each measurement area, level value, and mask judgment result value are returned.

  When the NUMBer header is omitted, all measurement results data are output.

  When the NUMBer header is specified, the measurement result data in the area specified by the number is output.
- [Parameter] None
- [Query reply] When the NUMBer header is omitted:

  NR3, NR3, NR3, NR1, NR3, NR3, NR3, NR1 , [NR3, NR3, NR3, NR1, NR3, NR3, NR3, NR1], ...,

  [NR3, NR3, NE3, NR1, NR3, NR3, NR3, NR1]

  Output order: Lower Freq(1): Unit Hz,
  Lower Level Abs(1): Unit dBm,
  Lower Level Rel(1): Unit dB,
  Lower P/F(1): 0/1,
  Upper Freq(1): Unit Hz,
  Upper Level Abs(1): Unit dBm,
  Upper Level Rel(1): Unit dB,
  Upper P/F(1) : 0/1
  [,[Lower Freq(2), Lower Level Abs(2),
  Lower Level Rel(2), Lower P/F(2), Upper Freq(2),
  Upper Level Abs(2), Upper Level Rel(2), Upper P/F(2)],
  ...,
  [Lower Freq(n), Lower Level Abs(n),
  Lower Level Rel(n), Lower P/F(n), Upper Freq(n),
  Upper level Abs(n), Upper Level Rel(n), Upper P/F(n)]

  n: Number of measurement areas defined: Up to 5

When the NUMBer header is specified:
NR3, NR3, NR3, NR1, NR3, NR3, NR3, NR1

Output order: Lower Freq(n): Unit Hz,
Lower Level Abs(n): Unit dBm,
Lower Level Rel(n): Unit dB
Lower P/F(n): 0/1,
Upper Freq(n): Unit Hz,
Upper Level Abs(n): Unit dBm,
Upper Level Rel(n): Unit dB
Upper P/F(n) : 0/1

n: Number of measurement areas defined: 1 to 5

- [Example] Result$ = Space$(1024)
  Call ibwrt (analyzer%, ":MEAS:SEM?")
  Call ibrd(analyzer%, Result$)
- [Relevant commands] :UNIT<ch>:POWer<screen>
  :READ<ch>:SEMask[:NUMBer{1|2|3|4|5}]?
  :FETCh<ch>:SEMask[:NUMBer{1|2|3|4|5}]?

5.6.20 :MEASure<ch>:SEMask:RPOWer?

- [Command syntax] :MEASure<ch>:SEMask:RPOWer?
- [Function description] Spectrum Emission Mask measurement execution and measurement result reading

  Executes Spectrum Emission Mask measurement and returns the reference power value after completion of the measurement.
- [Parameter] None
- [Query reply] NR3 (Real value reference power: Unit dBm)
- [Example] Result$ = Space$(1024)
  Call ibwrt (analyzer%, ":MEAS:SEM:RPOW?")
  Call ibrd (analyzer%, Result$)
- [Relevant commands] :UNIT<ch>:POWer<screen>
  :READ<ch>:SEMask:RPOWer?
  :FETCh<ch>:SEMask:RPOWer?

5.6.21 :MEASure<ch>:SEMask:FAIL?

- [Command syntax] :MEASure<ch>:SEMask:FAIL?
- [Function description] Spectrum Emission Mask measurement execution and overall Pass/Fail result reading

  Executes Spectrum Emission Mask measurement and returns the overall Pass/Fail result after completion of the measurement.
- [Parameter] None
- [Query reply] { PASS | FAIL }
- [Example]
  ```
  Result$ = Space$(1024)
  Call ibwrt (analyzer%, ":MEAS:SEM:FAIL?")
  Call ibrd (analyzer%, Result$)
  ```
- [Relevant commands] :READ<ch>:SEMask:FAIL?
  :FETCh<ch>:SEMask:FAIL?

5.6.22 :MEASure<ch>:IM[:NUMBer{1|3|5|7|9}]?

- [Command syntax] :MEASure<ch>:IM[:NUMBer{1|3|5|7|9}]?
- [Function description] IM measurement execution and measurement result reading

  Executes Intermodulation Distortion (IM) measurement and outputs the measurement result data after completion of the measurement.

  In Intermodulation Distortion measurement, measurements for the specified number of orders are performed, and the frequency and level of the reference signal, the frequency difference between two signals, and the results for the orders are output.

  When the NUMBer header is omitted, all measurement results data are output.

  When the NUMBer header is specified, the measurement result data on the specified order is output.
- [Parameter] None
- [Query reply] When the NUMBer header is omitted:

  NR3, NR3, NR3, NR3, NR3, NR1, NR3, NR1[, [NR3, NR1, NR3, NR1], ..., [NR3, NR1, NR3, NR1]]

Output order: Reference freq: Unit Hz,
Reference level: Unit dBm,
Delta freq: Unit Hz,
3rd order Intercept point: Unit dBm,
Fundamental wave Lower-side level: Unit dB,
-1: Fixed value,
Fundamental wave Upper-side level: Unit dB,
-1: Fixed value
[,[3rd order distortion Lower-side level: Unit dB,
3rd order distortion Lower-side P/F: 0/1,
3rd order distortion Upper-side level: Unit dB,
3rd order distortion Upper-side P/F: 0/1],
...,
[nth order distortion Lower-side level: Unit dB,
nth order distortion Lower-side P/F: 0/1,
nth order distortion Upper-side level: Unit dB,
nth order distortion Upper-side P/F: 0/1]]

n: Set orders (orders 3/5/7/9): Up to 4

When the NUMBer header is specified:
NR3, NR3, NR3, NR3, NR3, NR1, NR3, NR1

Output order: Reference freq: Unit Hz,
Reference level: Unit dBm,
Delta freq: Unit Hz,
3rd order Intercept point: Unit dBm,
nth order distortion Lower-side level: Unit dB,
nth order distortion Lower-side P/F: 0/1/-1,
nth order distortion Upper-side level: Unit dB,
nth order distortion Upper-side P/F: 0/1/-1

n: Specified orders (orders 1/3/5/7/9)

- [Example]

```
Result$ = Space$(1024)
Call ibwrt (analyzer%, ":MEAS:IM?")
Call ibrd(analyzer%, Result$)
```

- [Relevant commands]

:UNIT<ch>:POWer<screen>
:READ<ch>:IM[:NUMBer{1|3|5|7|9}]?
:FETCh<ch>:IM[:NUMBer{1|3|5|7|9}]?

5.6.23 :MEASure<ch>:IM:REFerence?

- [Command syntax] :MEASure<ch>:IM:REFerence?
- [Function description] IM measurement execution and reference frequency data reading

  Executes Intermodulation Distortion (IM) measurement and outputs the reference signal frequency and level value after completion of the measurement.
- [Parameter] None
- [Query reply] NR3, NR3 (Reference freq: Unit Hz, Reference level: Unit dBm)
- [Example] Result$ = Space$(1024)

  Call ibwrt (analyzer%, ":MEAS:IM:REF?")

  Call ibrd(analyzer%, Result$)
- [Relevant commands] :UNIT<ch>:POWer<screen>

  :READ<ch>:IM:REFerence?

  :FETCh<ch>:IM:REFerence?

5.6.24 :MEASure<ch>:IM:DELTa?

- [Command syntax] :MEASure<ch>:IM:DELTa?
- [Function description] IM measurement execution and reading of the frequency difference between two signals

  Executes Intermodulation Distortion (IM) measurement and outputs the frequency difference between the reference signal and the second signal input after completion of the measurement.
- [Parameter] None
- [Query reply] NR3 (Delta freq: Unit Hz)
- [Example] Result$ = Space$(1024)

  Call ibwrt (analyzer%, ":MEAS:IM:DELT?")

  Call ibrd(analyzer%, Result$)
- [Relevant commands] :UNIT<ch>:POWer<screen>

  :READ<ch>:IM:DELTa?

  :FETCh<ch>:IM:DELTa?

5.6.25 :MEASure<ch>:IM:IP3?

- [Command syntax] :MEASure<ch>:IM:IP3?
- [Function description] IM measurement execution and 3rd order intercept point value reading

  Executes Intermodulation Distortion (IM) measurement and outputs the 3rd order intercept point calculation result after completion of the measurement.
- [Parameter] None
- [Query reply] NR3 (3rd order Intercept point value: Unit dBm)
- [Example] Result$ = Space$(1024)

  Call ibwrt (analyzer%, ":MEAS:IM:IP3?")

  Call ibrd(analyzer%, Result$)
- [Relevant commands] :UNIT<ch>:POWer<screen>

  :READ<ch>:IM:IP3?

  :FETCh<ch>:IM:IP3?

5.6.26 :MEASure<ch>:IM:IPOint[:NUMBer{3|5|7|9}]?

- [Command syntax] :MEASure<ch>:IM:IPOint[:NUMBer{3|5|7|9}]?
- [Function description] IM measurement execution and intercept point value reading

  Executes Intermodulation Distortion (IM) measurement and outputs the intercept point calculation result after completion of the measurement.

  When the NUMBer header is omitted, all intercept point values are output according to the specified distortion order.

  When the NUMBer header is specified, the intercept point value of the specified distortion order is output.
- [Parameter] None
- [Query reply] When the NUMBer header is omitted:

  NR3[, NR3,..., NR3] (Intercept point value: Unit dBm)

  Output order: Real value 3rd intercept point value: Unit dBm,
  [Real value 5th intercept point value: Unit dBm,
  Real value 7th intercept point value: Unit dBm,
  Real value 9th intercept point value: Unit dBm]

  When the NUMBer header is specified:

  NR3 (Real value intercept point value{3|5|7|9}: Unit dBm)
- [Example]

```
Result$ = Space$(1024)
Call ibwrt (analyzer%, ":MEAS:IM:IPO?")
Call ibrd(analyzer%, Result$)
```

- [Relevant commands] :UNIT<ch>:POWer<screen>

  :READ<ch>:IM:IPOint[:NUMBer{3|5|7|9}]?

  :FETCh<ch>:IM:IPOint[:NUMBer{3|5|7|9}]?

5.6.27 :MEASure<ch>:IM:UPPer[:NUMBer{1|3|5|7|9}]?

- [Command syntax] :MEASure<ch>:IM:UPPer[:NUMBer{1|3|5|7|9}]?
- [Function description] IM measurement execution and reading of the specified-order modulation distortion measurement result

  Executes Intermodulation Distortion (IM) measurement and outputs the specified-order modulation distortion measurement value on the upper frequency side (Upper) after completion of the measurement.

  When the NUMBer header is omitted, the results data on all orders on the upper frequency side are output.

  When the NUMBer header is specified, the result data on the specified order is output.
- [Parameter] None
- [Query reply] When the NUMBer header is omitted:

  NR3, NR1, NR3, NR1 [, [NR3, NR1], ..., [NR3, NR1]]

  Output order: Fundamental wave Upper-side level: Unit dB,
  -1: Fixed value,
  3rd order distortion Upper-side level: Unit dB,
  3rd order distortion Upper-side P/F: 0/1
  [,[5th order distortion Upper-side level: Unit dB,
  5th order distortion Upper-side P/F: 0/1],
  ...,
  [nth order distortion Upper-side level: Unit dB,
  nth order distortion Upper-side P/F: 0/1]]

  n: Set orders (orders 3/5/7/9): Up to 4

  When the NUMBer header is specified:

  NR3, NR1

  Output order: nth order distortion Upper-side level: Unit dB,
  nth order distortion Upper-side P/F: 0/1/-1

  n: Specified orders (orders 1/3/5/7/9)
- [Example]

```
Result$ = Space$(1024)
Call ibwrt (analyzer%, ":MEAS:IM:UPP:NUMB3?")
Call ibrd(analyzer%, Result$)
```

- [Relevant commands] :READ<ch>:IM:UPPer[:NUMBer{1|3|5|7|9}]?

  :FETCh<ch>:IM:UPPer[:NUMBer{1|3|5|7|9}]?

5.6.28 :MEASure<ch>:IM:LOWer[:NUMBer{1|3|5|7|9}]?

- [Command syntax] :MEASure<ch>:IM:LOWer[:NUMBer{1|3|5|7|9}]?
- [Function description] IM measurement execution and reading of the specified-order modulation distortion measurement result

  Executes Intermodulation Distortion (IM) measurement and outputs the specified-order modulation distortion measurement value on the lower frequency side (Lower) after completion of the measurement.

  When the NUMBer header is omitted, the results data on all orders on the lower frequency side are output.

  When the NUMBer header is specified, the result data on the specified order is output.
- [Parameter] None
- [Query reply] When the NUMBer header is omitted:

  NR3, NR1, NR3, NR1 [, [NR3, NR1], ..., [NR3, NR1]]

  Output order: Fundamental wave Lower-side level: Unit dB,
  -1: Fixed value,
  3rd order distortion Lower-side level: Unit dB,
  3rd order distortion Lower-side P/F: 0/1
  [,[5th order distortion Lower-side level: Unit dB,
  5th order distortion Lower-side P/F: 0/1],
  ...,
  [nth order distortion Lower-side level: Unit dB,
  nth order distortion Lower-side P/: 0/1]]

  n: Set orders (orders 3/5/7/9): Up to 4

  When the NUMBer header is specified:

  NR3, NR1

  Output order: nth order distortion Lower-side level Unit dB,
  nth order distortion Lower-side P/F: 0/1/-1

  n: Specified orders (orders 1/3/5/7/9)
- [Example]

```
Result$ = Space$(1024)
Call ibwrt (analyzer%, ":MEAS:IM:LOW:NUMB3?")
Call ibrd(analyzer%, Result$)
```

- [Relevant commands] :READ<ch>:IM:LOWer[:NUMBer{1|3|5|7|9}]?
  :FETCh<ch>:IM:LOWer[:NUMBer{1|3|5|7|9}]?

5.6.29 :MEASure<ch>:HARMonics[:NUMBer{2|3|4|5|6|7|8|9|10}]?

- [Command syntax] :MEASure<ch>:HARMonics[:NUMBer{2|3|4|5|6|7|8|9|10}]?
- [Function description] Harmonic measurement execution and all measurement results reading

  Executes Harmonic measurement and outputs the measurement result data after completion of the measurement.

  With regard to the measurement result, the frequency, the absolute level, and the relative level comprise one set, and the data for the number of harmonic orders set are output.

  When the NUMBer header is omitted, all measurement results are output.

  When the NUMBer header is specified, the harmonic measurement result for the specified order is output.
- [Parameter] None
- [Query reply] When the NUMBer header is omitted:

  NR3, NR3, NR3, NR3, NR3[, [NR3, NR3, NR3], ..., [NR3, NR3, NR3]]

  Output order: Fundamental wave frequency: Unit Hz,
  Fundamental wave absolute level: Unit dBm,
  2nd harmonic frequency: Unit Hz,
  2nd harmonic absolute level: Unit dBm,
  2nd harmonic relative level: Unit dBc
  [,[3rd harmonic frequency: Unit Hz,
  3rd harmonic absolute level: Unit dBm,
  3rd harmonic relative level: Unit dBc],
  ...,
  [nth harmonic frequency: Unit Hz,
  nth harmonic absolute level: Unit dBm,
  nth harmonic relative level: Unit dBc]]

  n: Set harmonic order: Up to 10th

  When the NUMBer header is specified:

  NR3, NR3, NR3, NR3, NR3

  Output order: Fundamental wave frequency: Unit Hz,
  Fundamental wave absolute level: Unit dBm,
  nth harmonic frequency: Unit Hz,
  nth harmonic absolute level: Unit dBm,
  nth harmonic relative level: Unit dBc

  n: Specified harmonic order: 2 to 10

- [Example] Result$ = Space$(1024)
  Call ibwrt (analyzer%, ":MEAS:HARM:NUMB3?")
  Call ibrd(analyzer%, Result$)
- [Relevant commands] :UNIT<ch>:POWer<screen>
  :READ<ch>:HARMonics[:NUMBer{2|3|4|5|6|7|8|9|10}]?
  :FETCh<ch>:HARMonics[:NUMBer{2|3|4|5|6|7|8|9|10}]?

5.6.30 :MEASure<ch>:HARMonics:FUNDamental?

- [Command syntax] :MEASure<ch>:HARMonics:FUNDamental?
- [Function description] Harmonic measurement execution and fundamental wave measurement result reading

  Executes Harmonic measurement and outputs the measurement result data on the fundamental wave after completion of the measurement.
- [Parameter] None
- [Query reply] NR3, NR3 (Fundamental wave frequency: Unit Hz, Fundamental wave absolute level:Unit dBm)
- [Example] Result$ = Space$(1024)
  Call ibwrt (analyzer%, ":MEAS:HARM:FUND?")
  Call ibrd(analyzer%, Result$)
- [Relevant commands] :UNIT<ch>:POWer<screen>
  :READ<ch>:HARMonics:FUNDamental?
  :FETCh<ch>:HARMonics:FUNDamental?

5.6.31 :MEASure<ch>:CCDF[:NUMBer{1|2|3|4|5|6}]?

- [Command syntax] :MEASure<ch>:CCDF[:NUMBer{1|2|3|4|5|6}]?
- [Function description] CCDF measurement execution and measurement result reading

  Executes CCDF measurement and outputs Peak Factor, Average Power, and the power ratio after completion of the measurement.

  When the NUMBer header is omitted, all measurement results data are output.

  When the NUMBer header is specified, the measurement result data of the power ratio specified by NUMBer is output.

  NUMBer1: Power ratio of 10.0 %
  NUMBer2: Power ratio of 1.0 %
  NUMBer3: Power ratio of 0.1 %
  NUMBer4: Power ratio of 0.01 %
  NUMBer5: Power ratio of 0.001 %
  NUMBer6: Power ratio of 0.0001 %
- [Parameter] None
- [Query reply] When the NUMBer header is omitted:
  NR3, NR3, NR3, NR3, NR3, NR3, NR3, NR3

  Output order: Peak Factor: Unit dB,
  Average Power: Unit dBm,
  Power ratio of 10.0 %: Unit dB,
  Power ratio of 1.0 %: Unit dB,
  Power ratio of 0.1 %: Unit dB,
  Power ratio of 0.01 %: Unit dB,
  Power ratio of 0.001 %: Unit dB,
  Power ratio of 0.0001 %: Unit dB

  When the NUMBer header is specified:
  NR3, NR3, NR3

  Output order: Peak Factor: Unit dB,
  Average Power: Unit dBm,
  Specified power ratio: Unit dB
- [Example]

```
Result$ = Space$(1024)
Call ibwrt (analyzer%, ":MEAS:CCDF?")
Call ibrd(analyzer%, Result$)
```

- [Relevant commands] :UNIT<ch>:POWer<screen>
  :READ<ch>:CCDF[:NUMBer{1|2|3|4|5|6}]?
  :FETCh<ch>:CCDF[:NUMBer{1|2|3|4|5|6}]?

5.6.32 :MEASure<ch>:CCDF:PFACtor?

- [Command syntax] :MEASure<ch>:CCDF:PFACtor?
- [Function description] CCDF measurement execution and Peak Factor reading

  Executes CCDF measurement and outputs Peak Factor after completion of the measurement.
- [Parameter] None
- [Query reply] NR3 (Real value Peak Factor value: Unit dB)
- [Example] Result$ = Space$(1024)
  Call ibwrt (analyzer%, ":MEAS:CCDF:PFAC?")
  Call ibrd(analyzer%, Result$)
- [Relevant commands] :READ<ch>:CCDF:PFACtor?
  :FETCh<ch>:CCDF:PFACtor?

5.6.33 :MEASure<ch>:CCDF:APOWer?

- [Command syntax] :MEASure<ch>:CCDF:APOWer?
- [Function description] CCDF measurement execution and Average Power reading

  Executes CCDF measurement and outputs Average Power after completion of the measurement.
- [Parameter] None
- [Query reply] NR3 (Real value Average Power value: Unit dBm)
- [Example] Result$ = Space$(1024)
  Call ibwrt (analyzer%, ":MEAS:CCDF:APOW?")
  Call ibrd(analyzer%, Result$)
- [Relevant commands] :UNIT<ch>:POWer<screen>
  :READ<ch>:CCDF:APOWer?
  :FETCh<ch>:CCDF:APOWer?

5.6.34 :MEASure<ch>:CCDF:PRATio[:NUMBer{1|2|3|4|5|6}]?

- [Command syntax] :MEASure<ch>:CCDF:PRATio[:NUMBer{1|2|3|4|5|6}]?
- [Function description] CCDF measurement execution and power ratio reading

  Executes CCDF measurement and outputs the power ratio after completion of the measurement.

  When the NUMBer header is omitted, all measurement results data are output.

  When the NUMBer header is specified, the measurement result data of the power ratio specified by NUMBer is output.

  NUMBer1: Power ratio of 10.0 %
  NUMBer2: Power ratio of 1.0 %
  NUMBer3: Power ratio of 0.1 %
  NUMBer4: Power ratio of 0.01 %
  NUMBer5: Power ratio of 0.001 %
  NUMBer6: Power ratio of 0.0001 %
- [Parameter] None
- [Query reply] When the NUMBer header is omitted:
  NR3, NR3, NR3, NR3, NR3, NR3

  Output order: Power ratio of 10.0 %: Unit dB,
  Power ratio of 1.0 %: Unit dB,
  Power ratio of 0.1 %: Unit dB,
  Power ratio of 0.01 %: Unit dB,
  Power ratio of 0.001 %: Unit dB,
  Power ratio of 0.0001 %: Unit dB

  When the NUMBer header is specified:
  NR3

  Output order: Specified power ratio: Unit dB
- [Example] Result$ = Space$(1024)
  Call ibwrt (analyzer%, ":MEAS:CCDF:PRAT?")
  Call ibrd(analyzer%, Result$)
- [Relevant commands] :READ<ch>:CCDF:PRATio[:NUMBer{1|2|3|4|5|6}]?
  :FETCh<ch>:CCDF:PRATio[:NUMBer{1|2|3|4|5|6}]?

5.6.35 :MEASure<ch>:MAPower[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

- [Command syntax] :MEASure<ch>:MAPower[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

- [Function description] Multi-Average Power measurement execution and the measurement result (Trace) reading

  Executes the Multi-Average Power measurement and returns the measurement result after the measurement is complete.

- [Parameter] None

- [Query reply] When the NUMBer header is omitted:

  NR3[,NR3,...,NR3]
  (Real value that indicates Average Power: Unit dBm)

  Output order:Average Power of all windows that are set to ON

  When the NUMBer header is specified:

  NR3
  (Real number that indicates Average Power of the specified window: Unit dBm)

- [Example] Result$ = Space$(1024)

  Call ibwrt (analyzer%, ":MEAS:MAP?")

  Call ibrd (analyzer%, Result$)

- [Relevant commands] :UNIT<ch>:POWer<screen>

  :READ<ch>:MAPower[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

  :FETCh<ch>:MAPower[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

5.6.36 :MEASure<ch>:MAPower:PDENsity[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

- [Command syntax] :MEASure<ch>:MAPower:PDENsity[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?
- [Function description] Multi-Average Power measurement execution and the average power density (Trace) reading

  Executes the Multi-Average Power measurement and returns the average power density after the measurement is complete.
- [Parameter] None
- [Query reply] When the NUMBer header is omitted:

  NR3[,NR3,...,NR3]
  (Real value that indicates the average power density: Unit dBm/Hz or dBμV/√Hz)

  Output order: Average power density of all windows that are set to ON

  When the NUMBer header is specified:

  NR3
  (Real value that indicates the average power density of the specified window: Unit dBm/Hz or dBμV/√Hz)
- [Example] Result$ = Space$(1024)

  Call ibwrt (analyzer%, :MEAS:MAP:PDEN?)

  Call ibrd (analyzer%, Result$)
- [Relevant commands] :UNIT<ch>:POWer<screen>

  :READ<ch>:MAPower:PDENsity[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

  :FETCh<ch>:MAPower:PDENsity[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

5.6.37 :MEASure<ch>:MAPower:PRATio[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

- [Command syntax] :MEASure<ch>:MAPower:PRATio[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?
- [Function description] Multi Average Power measurement execution and the Power Ratio (Trace) reading

  Executes the Multi Average Power measurement and returns the measurement result after the measurement is complete.

  When executing the Power Ratio measurement, set the Power Ratio to ON.
- [Parameter] None
- [Query reply] When the NUMBer header is omitted:

  NR3[,NR3,...,NR3](Real value that indicates Power Ratio: Unit dB)

  Output order: Power Ratio of all windows that are set to ON

  When the NUMBer header is specified:

  NR3

  (Real value that indicates Power Ratio of the specified window: Unit dB)
- [Example] Result$ = Space$(1024)

  Call ibwrt (analyzer%, :MEAS:MAP:PRAT?)

  Call ibrd (analyzer%, Result$)
- [Relevant commands] :READ<ch>:MAPower:PRATio[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

  :FETCh<ch>:MAPower:PRATio[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

5.6.38 :MEASure<ch>:MAPower:RMS[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

- [Command syntax] :MEASure<ch>:MAPower:RMS[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?
- [Function description] Multi-Average Power measurement execution and the measurement result (RMS) reading

  Executes the Multi Average Power measurement and returns the measurement result after the measurement is complete.
- [Parameter] None
- [Query reply] When the NUMBer header is omitted:

  NR3[,NR3,...,NR3](Real value that indicates Average Power: Unit dBm)

  Output order: Average Power of all windows that are set to ON

  When the NUMBer header is specified:

  NR3

  (Real value that indicates Average Power of the specified window: Unit dBm)
- [Example] Result$ = Space$(1024)

  Call ibwrt (analyzer%, :MEAS:MAP:RMS?)

  Call ibrd (analyzer%, Result$)
- [Relevant commands] :UNIT<ch>:POWer<screen>

  :READ<ch>:MAPower:RMS[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

  :FETCh<ch>:MAPower:RMS[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

5.6.39 :MEASure<ch>:MAPower:RMS:PDENsity[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

- [Command syntax] :MEASure<ch>:MAPower:RMS:PDENsity[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?
- [Function description] Multi-Average Power measurement execution and the average power density (RMS) reading

  Executes the Multi-Average Power measurement and returns the average power density after the measurement is complete.
- [Parameter] None
- [Query reply] When the NUMBer header is omitted:

  NR3[,NR3,...,NR3]
  (Real value that indicates the average power density: Unit dBm/Hz or dBμV/√Hz)

  Output order: Average power density of all windows that are set to ON

  When the NUMBer header is specified:

  NR3
  (Real value that indicates the average power density of the specified window: Unit dBm/Hz or dBμV/√Hz)
- [Example] Result$ = Space$(1024)

  Call ibwrt (analyzer%, :MEAS:MAP:RMS:PDEN?)

  Call ibrd (analyzer%, Result$)
- [Relevant commands] :UNIT<ch>:POWer<screen>

  :READ<ch>:MAPower:RMS:PDENsity[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

  :FETCh<ch>:MAPower:RMS:PDENsity[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

5.6.40 :MEASure<ch>:MAPower:RMS:PRATio[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

- [Command syntax] :MEASure<ch>:MAPower:RMS:PRATio[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?
- [Function description] Multi-Average Power measurement execution and the Power Ratio (RMS) reading

  Executes the Multi-Average Power measurement and returns the measurement result after the measurement is complete.

  When executing the Power Ratio measurement, set the Power Ratio to ON.
- [Parameter] None
- [Query reply] When the NUMBer header is omitted:

  NR3[,NR3,...,NR3] (Real value that indicates Power Ratio: Unit dB)

  Output order: Power Ratio of all windows that are set to ON

  When the NUMBer header is specified:

  NR3

  (Real value that indicates Power Ratio of the specified window: Unit dB)
- [Example] Result$ = Space$(1024)

  Call ibwrt (analyzer%, :MEAS:MAP:RMS:PRAT?)

  Call ibrd (analyzer%, Result$)
- [Relevant commands] :READ<ch>:MAPower:RMS:PRATio[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

  :FETCh<ch>:MAPower:RMS:PRATio[:NUMBer{1|2|3|4|5|6|7|8|9|10}]?

5.7 Initiate Commands

This section describes the Initiate subsystem.

In the Initiate subsystem, the commands used for sweep control in the SA mode are defined.

Command	Function	Reference Page
:INITiate<ch>		
:CONTinuous	Continuous sweep mode ON/OFF	5-177
[:IMMediate]	Starting a sweep or measurement	5-178
:RESTart	Resetting and restarting a sweep	5-178
:ABORt	Aborting a sweep	5-178
:TS	Resetting and restarting a sweep, and suspending after the completion of the sweep	5-179

5.7.1 :INITiate<ch>:CONTinuous

- [Command syntax] :INITiate<ch>:CONTinuous < bool >
  :INITiate<ch>:CONTinuous?
- [Function description] Continuous sweep mode ON/OFF

  Sets the sweep mode. There are two sweep modes, as listed below:

  Continuous sweep mode:
  Sweeps run continuously.

  Single sweep mode:
  After the completion of a sweep, the sweep is terminated and the next sweep is not started.
- [Parameter] < bool > = { OFF | ON }
  OFF: Sets the Single sweep mode
  ON: Sets the continuous sweep mode
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":INIT:CONT ON")
- [Relevant commands] :INITiate<ch>[:IMMediate]
  :INITiate<ch>:RESTart

5.7.2 :INITiate<ch>[:IMMediate]

- [Command syntax] :INITiate<ch>[:IMMediate]
- [Function description] Starts a sweep or measurement

  Starts a sweep immediately. In addition, by sending this command after entry into each measurement mode, the respective measurements are started.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":INIT:IMM")
- [Relevant commands] :INITiate<ch>:RESTart
  :INITiate<ch>:ABORt

5.7.3 :INITiate<ch>:RESTart

- [Command syntax] :INITiate<ch>:RESTart
- [Function description] Resets and restarts a sweep

  After resetting the current sweep, a new sweep is started.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":INIT:REST")
- [Relevant commands] :INITiate<ch>[:IMMediate]
  :INITiate<ch>:ABORt

5.7.4 :INITiate<ch>:ABORt

- [Command syntax] :INITiate<ch>:ABORt
- [Function description] Aborts a sweep

  Aborts the current sweep.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":INIT:ABOR")
- [Relevant commands] :INITiate<ch>:RESTart

5.7.5 :INITiate<ch>:TS

- [Command syntax] :INITiate<ch>:TS
- [Function description] Resets and restarts a sweep, and makes a suspension after the completion of the sweep

  After resetting the current sweep, a new sweep is started.

  Unlike the INITiate<ch>:RESTart command, this command does not accept the next command until the completion of the current sweep.The next sweep execution waits until the current sweep is completed. This function allows programming of functions that require full completion of a sweep to get a correct result (e.g., frequency counter function) without caring about the acquisition timing of the measurement result.

  When a sweep is started and completed using this command, the single sweep condition is applied and the sweep is suspended.

  To return to the continuous sweep condition, use the :INITiate<ch>:CONTinuous command.
- [Parameter] None
- [Query reply] None
- [Example]

```
FreqCOUNT$ = Space$(40)
'-------------- Prepare counter measurement ------------------
Call ibwrt (analyzer%, ":CALC:MARK:FCO ON")
'-------------- Start "take sweep mode" -----------------------
Call ibwrt (analyzer%, ":INIT:TS")
'------- Read the result of counter meas. after sweeping ---
Call ibwrt (analyzer%, ":CALC:MARK:FCO:FREQ?")
Call ibrd (analzer%, FreqCOUNT$)
```

- [Relevant commands] :INITiate<ch>:RESTart
  :INITiate<ch>:CONTinuous

5.8 Trigger Commands

This section describes the Trigger subsystem.

In the Trigger subsystem, the commands related to the trigger functions for sweep and measurement are defined.

Command	Function	Reference Page
:TRIGger<ch> [:SEQuence<screen>]		
:SOURce	Setting the trigger	5-180
:SLOPe	Setting the trigger polarity of each trigger source	5-181
:LEVel		
:VIDeo	Setting the trigger level when using a Video trigger	5-181
:EXTernal	Setting the trigger level when using an EXT2 (external input terminal 2) trigger	5-182
:IF	Setting the trigger level when using an IF trigger	5-182
:DELay	Setting the trigger delay value	5-183

5.8.1 :TRIGger<ch>[:SEQuence<screen>]:SOURce

- [Command syntax] :TRIGger<ch>[:SEQuence<screen>]:SOURce < type >
  :TRIGger<ch>[:SEQuence<screen>]:SOURce?
- [Function description] Sets the trigger

  Sets the trigger source.

  If there is no option for using the Link trigger, the Link trigger cannot be set.
- [Parameter] < type > = { IMMediate | IF | VIDeo | EXT1 | EXT2 | LINE | LINK }
  IMMediate: Free-run mode without trigger setting
  IF: IF trigger
  VIDeo: Video trigger
  EXT1: EXT1 input signal trigger
  EXT2: EXT2 input signal trigger
  LINE: Power line trigger
  LINK: Link trigger
- [Query reply] { IMM | IF | VID | EXT1 | EXT2 | LINE | LINK }
- [Example] Call ibwrt (analyzer%, ":TRIG:SEQ:SOUR IF")
- [Relevant commands] :TRIGger<ch>[:SEQuence<screen>]:SLOPe

5.8.2 :TRIGger<ch>[:SEQuence<screen>]:SLOPe

- [Command syntax] :TRIGger<ch>[:SEQuence<screen>]:SLOPe < type >
  :TRIGger<ch>[:SEQuence<screen>]:SLOPe?
- [Function description] Sets the trigger polarity of each trigger source

  Sets the trigger polarity of the set trigger sources. The trigger polarity can be set for the following five types of trigger sources:
  - Video trigger
  - IF trigger
  - Ext1 trigger
  - Ext2 trigger
  - Link trigger
- [Parameter] < type > = { NEGative | POSitive }
  NEGative: Falling or negative polarity
  POSitive: Rising or positive polarity
- [Query reply] { NEG | POS }
- [Example] Call ibwr (analyzer%, ":TRIG:SEQ:SLOP POS")
- [Relevant commands] :TRIGger<ch>[:SEQuence<screen>]:SOURce

5.8.3 :TRIGger<ch>[:SEQuence<screen>]:LEVel:VIDeo

- [Command syntax] :TRIGger<ch>[:SEQuence<screen>]:LEVel:VIDeo < real >
  :TRIGger<ch>[:SEQuence<screen>]:LEVel:VIDeo?
- [Function description] Sets the trigger level when using a Video trigger

  Sets the trigger level value when using a Video trigger.
- [Parameter] < real > = Trigger level value (dBm)
- [Query reply] NR3 (Real value:unit dBm)
- [Example] '------------ Set trigger level to -50dbm ------------------
  Call ibwr (analyzer%, ":TRIG:SEQ:LEV:VID -50DBM")
- [Relevant commands] :TRIGger<ch>[:SEQuence<screen>]:SOURce
  :TRIGger<ch>[:SEQuence<screen>]:SLOPe
  :UNIT<ch>:POWer<screen>

5.8.4 :TRIGger<ch>[:SEQuence<screen>]:LEVel:EXTernal

- [Command syntax] :TRIGger<ch>[:SEQuence<screen>]:LEVel:EXTernal < real >
  :TRIGger<ch>[:SEQuence<screen>]:LEVel:EXTernal?
- [Function description] Sets the trigger level when using an EXT2 (external input terminal 2) trigger

  Sets the trigger level value when using an EXT2 trigger with a voltage value.
- [Parameter] < real > = Voltage value
  Setting range: 0 - 5 V
- [Query reply] NR3 (Real value: Unit V)
- [Example] '------------ Set trigger level to 4.5 V ---------------------
  Call ibwr (analzyzer%, ":TRIG:SEQ:LEV:EXT 4.5")
- [Relevant commands] :TRIGger<ch>[:SEQuence<screen>]:SOURce
  :TRIGger<ch>[:SEQuence<screen>]:SLOPe

5.8.5 :TRIGger<ch>[:SEQuence<screen>]:LEVel:IF

- [Command syntax] :TRIGger<ch>[:SEQuence<screen>]:LEVel:IF < real >
  :TRIGger<ch>[:SEQuence<screen>]:LEVel:IF?
- [Function description] Sets the trigger level when using an IF trigger

  Sets the trigger level value when using an IF trigger in percentage terms.
- [Parameter] < real > = A percentage value for the IF input scale
  Setting range: 0 - 100%
- [Query reply] NR3 (Real value: Unit %)
- [Example] '----------- Set trigger level to 55% -------------------
  Call ibwrt (analyzer%, ":TRIG:SEQ:LEV:IF 55PCT")
- [Relevant commands] :TRIGger<ch>[:SEQuence<screen>]:SOURce
  :TRIGger<ch>[:SEQuence<screen>]:SLOPe

5.8.6 :TRIGger<ch>[:SEQuence<screen>]:DELay

- [Command syntax] :TRIGger<ch>[:SEQuence<screen>]:DELay < real >
  :TRIGger<ch>[:SEQuence<screen>]:DELay?
- [Function description] Sets the trigger delay value

  Sets the trigger delay time when using a trigger mode except for the free-run trigger and power line trigger. The delay time is measured from the trigger point.
- [Parameter] < real > = Trigger delay time (s/ms/μs/ns)
- [Query reply] NR3 (Real value: Unit s)
- [Example] '------------- Set trigger delay time to 200ms ----------
  Call ibwrt (analyzer%, ":TRIG:SEQ:DEL 200MS")
- [Relevant commands] :TRIGger<ch>[:SEQuence<screen>]:SOURce
  :TRIGger<ch>[:SEQuence<screen>]:SLOPe

5.9 Display Commands

This section describes the Display subsystem.

In the Display subsystem, the commands related to the display of settings of screen display scales and annotations are defined.

Command	Function	Reference Page
:DISPlay		
:ANNotation		
:CURSor	Sets the display of the XY cursor information data to ON or OFF	5-186
:DATE	Sets the date display ON/OFF	5-186
:FORMat	Sets the date display type	5-187
:DLINe	Sets the display of the information data for the Display Line ON/OFF	5-187
:LOGO	Sets the display of the ADVANTEST logo ON/OFF	5-188
:RLINe	Sets the display of the information data for Reference Line ON/OFF	5-188
:TITLe	Sets the screen title	5-189
:WINDow	Sets the display of the information data for the measuring window ON/OFF	5-189
:DISPlay<ch>		
[:WINDow:<screen>]		
:ACTive	Specifies the active screen when the waveform zoom function is turned on	5-190
:TRACe		
[:NUMBer{1|2|3|4}]		
:ACTive	Selects the active trace	5-191
:MODE	Sets the display mode of the specified trace	5-192
:NCORrection		
:STATe	Sets the trace normalize function ON/OFF	5-193
: STORe	Stores the reference waveform data to be used for the trace normalize function	5-194
[:NUMBer{1|2}]		
:STORe	Stores the waveform data of trace 1 or 2	5-195
:AANalog		
:STATe	Sets the quasi analog display mode ON/OFF	5-195

Command	Function	Reference Page
:DISPlay<ch>		
[:WINDow]		
:TRACe		
:SPLit	Sets the two screen display mode ON/OFF	5-194
:X		
[:SCALe]		
:ZOOM		
:MODE	Selects the waveform zoom function and releases the zoom function	5-196
:FREQuency		
:CENTer	Specifies the zoom frequency when the waveform zoom function is turned on	5-197
:SPAN	Specifies the zoom width when the waveform zoom function is turned on	5-198
:TIME		
:DELay	Specifies the zoom time position when the waveform zoom function is turned on	5-198
:WIDTh	Specifies the zoom time width when the waveform zoom function is turned on	5-199
:DISPlay<ch>		
[:WINDow<screen>]		
:TRACe		
:Y		
[:SCALe]		
:RLEVel	Sets the reference level	5-199
:OFFSet	Sets the Offset value to the reference level value	5-200
:STATe	Sets the Offset value to the reference level value ON/OFF	5-201
:PDIVision	Sets the value of one division when the log scale is displayed	5-202
:SPACing	Sets the type of the vertical scale	5-202
:DISPlay<ch>		
[:WINDow]		
:TRACe		
:CCDF		
:STATe	Sets the reference waveform display in the CCDF measurement to ON or OFF	5-203
:GAUSsian		
:STATe	Sets the ideal gaussian noise in the CCDF measurement to ON or OFF	5-203
:X		
[:SCALe]		
:CCDF	Sets the maximum horizontal axis value for the waveform display in the CCDF measurement	5-204

5.9.1 :DISPlay:ANNotation:CURSor

- [Command syntax] :DISPlay:ANNotation:CURSor < bool >
  :DISPlay:ANNotation:CURSor?
- [Function description] Sets the display of the XY cursor information data to ON or OFF.

  Switches ON and OFF the display of the information data such as a XY cursor position that is displayed while the XY cursor function is set to ON.
- [Parameter] < bool > = { OFF | ON }
  OFF: Hides the display of the XY cursor information.
  ON: Displays the XY cursor information.
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":DISP:ANN:CURS ON")
- [Relevant commands] :CALCulate<ch>:CURSor<screen>:STATe

5.9.2 :DISPlay:ANNotation:DATE

- [Command syntax] :DISPlay:ANNotation:DATE < bool >
  :DISPlay:ANNotation:DATE?
- [Function description] Sets the date display ON/OFF

  Sets the date display on the screen of this instrument ON/OFF.
- [Parameter] < bool > = { OFF | ON }
  OFF: Sets the date display off
  ON: Sets the date display on
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":DISP:ANN:DATE ON")
- [Relevant command] :DISPlay:ANNotation:DATE:FORMat

5.9.3 :DISPlay:ANNotation:DATE:FORMat

- [Command syntax] :DISPlay:ANNotation:DATE:FORMat < type >
  :DISPlay:ANNotation:DATE:FORMat?
- [Function description] Sets the date display type

  Sets the type of the date to be displayed on the screen of this instrument. A format can be selected from the following three types: Month/Day/ Year, Day/Month/Year, and Year/Month/Day.
- [Parameter] < type > = { MDY | DMY | YMD }
  MDY: Selects the Month/Day/Year format
  DMY: Selects the Day/Month/Year format
  YMD: Selects the Year/Month/Day format
- [Query reply] { MDY | DMY | YMD }
- [Example] Call ibwrt (analyzer%, ":DISP:ANN:DATE:FORM MDY")
- [Relevant command] :DISPlay:ANNotation:DATE

5.9.4 :DISPlay:ANNotation:DLINe

- [Command syntax] :DISPlay:ANNotation:DLINe < bool >
  :DISPlay:ANNotation:DLINe?
- [Function description] Sets the display of the information data for Display Line ON/OFF

  Switches ON and OFF the display of the Display Line position information to be displayed when the Display Line function is turned on.
- [Parameter] < bool > = { OFF | ON }
  OFF: Turns off the display of the Display Line information
  ON: Turns on the display of the Display Line information
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":DISP:ANN:DLIN OFF")
- [Relevant commands] :CALCulate:DLINe<screen>:STATe

5.9.5 :DISPlay:ANNotation:LOGO

- [Command syntax] :DISPlay:ANNotation:LOGO < bool >
  :DISPlay:ANNotation:LOGO?
- [Function description] Sets the display of the ADVANTEST logo ON/OFF

  Switches the ADVANTEST logo which is displayed on the screen of this instrument ON/OFF.
- [Parameter] < bool > = { OFF | ON }
  OFF: Turns off the logo display
  ON: Turns on the logo display
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":DISP:ANN:LOGO OFF")
- [Relevant commands]

5.9.6 :DISPlay:ANNotation:RLINe

- [Command syntax] :DISPlay:ANNotation:RLINe < bool >
  :DISPlay:ANNotation:RLINe?
- [Function description] Sets the display of the information data for Reference Line ON/OFF

  Switches ON/OFF of the Reference Line position data, etc. when the Reference Line function is on.
- [Parameter] < bool > = { OFF | ON }
  OFF: Turns off the display of the Reference Line data
  ON: Turns on the display of the Reference Line data
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":DISP:ANN:RLIN ON")
- [Relevant commands] :CALCulate<ch>:RLINe<screen>:STATe

5.9.7 :DISPlay:ANNotation:TITLe

- [Command syntax] :DISPlay:ANNotation:TITLe < str >
  :DISPlay:ANNotation:TITLe?
- [Function description] Sets the screen title

  Specifies with a character string the screen title to be displayed on the screen of this instrument.

  A character string up to 32 characters in length can be specified.

  Alphanumeric characters except the special characters shown below can be used for the character string of the title.
- [Parameter] <str>=" character string "
- [Query reply] " character string "
- [Example] '-------------- Set the display title to "ADVANTEST CO."---------------
  Call ibwrt (analyzer%, ":DISP:ANN:TITL ""ADVANTEST CO.""")
- [Relevant commands]

5.9.8 :DISPlay:ANNotation:WINDow

- [Command syntax] :DISPlay:ANNotation:WINDow < bool >
  :DISPlay:ANNotation:WINDow?
- [Function description] Sets the display of the information data for the measuring window ON/OFF

  Switches ON and OFF the display of the measuring window position information to be displayed when the measuring window function is turned on.
- [Parameter] < bool > = { OFF | ON }
  OFF: Turns off the display of the measuring window information
  ON: Turns on the display of the measuring window information
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":DISP:ANN:WIND ON")
- [Relevant commands] :CALCulate<ch>:WINDow<screen>:STATe

5.9.9 :DISPlay<ch>[:WINDow<screen>]:ACTive

- [Command syntax] :DISPlay<ch>[:WINDow<screen>]:ACTive
  :DISPlay<ch>[:WINDow<screen>]:ACTive?
- [Function description] Specifies the active screen when the waveform zoom function is turned on

  Generally, when the zoom function is turned on, two screens are displayed; the upper screen displays the original waveform and the lower screen displays the zoomed waveform. Set the active screen by using this command if setting a frequency without specifying the target screen.
- [Parameter] None
- [Query reply] { 1 | 2 }
  1: Currently the upper screen is active.
  2: Currently the lower screen is active.
- [Example]

```
'-------Appear 2 screens on display by zoom function ----------
Call ibwrt (analyzer%, ":DISP:TRAC:X:ZOOM:MODE ZMFF")
'---------- Select active screen to the upper screen -------------
Call ibwrt (analyzer%, ":DISP:WIND1:ACT")
'---------- Set the center freq. to the upper screen --------------
Call ibwrt (analyzer%, ":FREQ:CENT 2GHZ")
```

- [Relevant commands] :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:MODE
  :DISPlay<ch>[:WINDow]:TRACe:SPLit

5.9.10 :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}]:ACTive

- [Command syntax] :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}]:ACTive

  :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}]:ACTive?
- [Function description] Selects the active trace

  Out of four traces on the specified screen, selects the target trace to be operated when the trace number which is in the commands related to the trace after this command is not specified.
- [Parameter] None
- [Query reply] { 1 | 2 | 3 | 4 }
- [Example]

```
'----------- Activate the trace no. to 2 -------------------------------
Call ibwrt (analyzer%, ":DISP:WIND1:TRAC:NUMB2:ACT")
'----------- Set trace mode to MAX hold for trace no.2 ---------------
Call ibwrt (analyzer%, ":DISP:WIND1:TRAC:MODE MAXH")
```

- [Relevant commands] :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}]:MODE

  :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}]:NCORrection:STATe

  :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}]:NCORrection:STORe

5.9.11 :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}]:MODE

- [Command syntax] :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}]:MODE<type >

  :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}]:MODE?

- [Function description] Sets the display mode of the specified trace

  Sets the display mode of the specified trace. Select a display mode from the six types listed below. When the trace No. is omitted, the mode of the trace which is active at the time is set.

  WRITE: Writes a waveform each time a sweep is performed. Only the latest waveform data is used.

  VIEW: Stops updating the waveform, and displays the waveform specified when VIEW is specified.

  BLANK: Deletes the waveform.

  MAX HOLD: Displays the maximum value in the waveform data at each point (frequency/time) previously updated each time a sweep is performed.

  MIN HOLD: Displays the minimum value in the waveform data at each point (frequency/time) previously updated each time a sweep is performed.

  AVERAGE: According to the operation method specified with the average type, calculates the average value from the waveform data of every sweep, and displays the result as the waveform data. There are three average types: Video, RMS, and Volts.

- [Parameter] < type> = { WRITe | VIEW | BLANk | MAXHold | MINHold | AVERage }

  WRITe: Sets the WRITE mode.

  VIEW: Sets the VIEW mode.

  BLANk: Sets the BLANK mode.

  MAXHold: Sets the MAX HOLD mode.

  MINHold: Sets the MIN HOLD mode.

  AVERage: Sets the AVERAGE mode.

- [Query reply] { WRIT | VIEW | BLAN | MAXH | MINH | AVER }

- [Example]

```
'---- Set trace mode to max hold mode for trace no.1,
'-----and set it to average mode for trace no.2
Call ibwrt (analyzer%, ":DISP:WIND1:TRAC:NUMB1
:MODE MAXH")
Call ibwrt (analyzer%, ":DISP:WIND1:TRAC:NUMB2:MODE AVER")
```

- [Relevant commands] :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}]:ACTive

5.9.12 :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}] :NCORrection:STATe

- [Command syntax]

  :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}] :NCORrection:STATe < bool >

  :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}] :NCORrection:STATe?

- [Function description]

  Sets the trace normalize function ON/OFF

  Sets ON or OFF the normalize function for the trace on the specified screen.

  When the normalize function is turned on,
  Using the :DISPlay<ch>[:WINDow<screen>] :TRACe[:NUMBer{1|2|3|4}]:NCORrection:STORe command, the difference of the current sweep waveform is calculated based on the waveform data saved in the internal memory of this instrument, and displayed as a waveform.

  When the trace No. is omitted, the mode of the trace which is active at the time is set.

- [Parameter]

  < bool > = { OFF | ON }

  OFF: Turns off the trace normalize function.

  ON: Turns on the trace normalize function.

- [Query reply]

  { OFF | ON }

- [Example]

  Call ibwrt (analyzer%, ":DISP:TRAC:NUMB1:NCOR:STAT ON")

- [Relevant commands]

  :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}] :ACTive

  :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}] :NCORrection:STORe

5.9.13 :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}]:NCORrection :STORe

- [Command syntax] :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}] :NCORrection:STORe
- [Function description] Stores the reference waveform data to be used for the trace normalize function

  Stores the reference waveform data to be used for the trace on the specified screen when the normalize function is turned on.

  The waveform data to be stored is the waveform data of the specified trace which is taken in this instrument when this command is issued. When the trace No. is omitted, the mode of the trace which is active at the time is set.
- [Parameter] None
- [Query reply] None
- [Example] '-- Save the current data for normaraize operation and then set it to on ---

  Call ibwrt (analyzer%, ":DISP:TRAC:NUMB1:NCOR:STOR")

  Call ibwrt (analyzer%, ":DISP:TRAC:NUMB1:NCOR:STAT ON")
- [Relevant commands] :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}] :ACTive

  :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}] :NCORrection:STATe

5.9.14 :DISPlay<ch>[:WINDow]:TRACe:SPLit

- [Command syntax] :DISPlay<ch>[:WINDow]:TRACe:SPLit < bool >

  :DISPlay<ch>[:WINDow]:TRACe:SPLit?
- [Function description] Sets the two screen display mode ON/OFF

  Sets the screen which is split up and down. Each screen can be set independently.
- [Parameter] < bool > = { OFF | ON }

  OFF: Changes to the single-screen display

  ON: Changes to the two-screen display
- [Query reply] { OFF | ON }
- [Example] '------------ Change the screen mode to 2 screen mode --------

  Call ibwrt (analyzer%, ":DISP:TRAC:SPL ON")
- [Relevant commands]

5.9.15 :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2}]:STORe

- [Command syntax] :DISPlay<ch>[:WINDow<screen>]:TRACe[NUMBer{1|2}]:STORe
- [Function description] Stores the waveform data of trace 1 or 2

  Moves the present waveform data of trace 1 or 2 to the waveform data of trace 3 or 4 on the specified screen and stores.

  When the data is stored in trace 3 or 4, at the same time its waveform display mode is automatically changed to VIEW mode. The trace number of the data destination is fixed. The data of trace 1 is stored in trace 3, and the data of trace 2 is stored in trace 4.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":DISP:TRAC:NUMB1:STOR")
- [Relevant commands]

5.9.16 :DISPlay<ch>[:WINDow<screen>]:TRACe:AANalog:STATe

- [Command syntax] :DISPlay<ch>[:WINDow<screen>]:TRACe:AANalog:STATe < bool >
  :DISPlay<ch>[:WINDow<screen>]:TRACe:AANalog:STATe?
- [Function description] Sets ON/OFF of the quasi analog trace mode

  Sets the quasi analog display mode ON/OFF.
- [Parameter] < bool > = { OFF | ON }
  OFF: Cancels the quasi analog display mode
  ON: Sets the quasi analog display mode
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":DISP:TRAC:AAN:STAT ON")
- [Relevant commands] [:SENSe<ch>]:AANalog:SAMPle:COUNt

5.9.17 :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:MODE

- [Command syntax]
  :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:MODE < type >
  :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:MODE?
- [Function description]
  Selects the waveform zoom function and releases the zoom function

  Sets the screen zoom function ON/OFF. When the zoom function is turned on, the two-screen display mode is automatically specified. The upper screen displays the original waveform and the lower screen displays the zoomed waveform of the waveform in the zoomed area on the upper screen.

  There are three zoom function modes: the frequency vs. frequency axis, the time vs. time axis, and the frequency vs. time axis. The setting state of the lower screen is changed according to the specified mode.
- [Parameter]
  < type > = { OFF | ZMFF | ZMTT | ZMFT }
  OFF: Cancels the zoom state
  ZMFF: Changes to the zoom mode (frequency vs. frequency axis)
  ZMTT: Change to the zoom mode (time vs. time axis)
  ZMFT: Changes to the zoom mode (frequency vs. time axis)
- [Query reply]
  { OFF | ZMFF | ZMTT | ZMFT }
  OFF: The zoom mode is not specified
  ZMFF: The zoom mode is specified (frequency vs. frequency axis)
  ZMTT: The zoom mode is specified (time vs. time axis)
  ZMFT: The zoom mode is specified (frequency vs. time axis)
- [Example]
  Call ibwrt (analyzer%, ":DISP:TRAC:X:ZOOM:MODE ZMFF")
- [Relevant commands]
  :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:FREQuency :CENTer
  :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:FREQuency :SPAN
  :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:TIME:DELay
  :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:TIME:WIDTh

5.9.18 :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:FREQuency:CENTer

- [Command syntax] :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:FREQuency :CENTer < real >

  :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:FREQuency :CENTer?
- [Function description] Specifies the zoom frequency when the waveform zoom function is turned on

  Specify the center frequency position to be zoomed using the zoom function. When the zoom function is turned on, the center frequency of the lower screen is set by sending this command. At this frequency as the center, the frequency span is zoomed into the zoom frequency span specified by the DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM: FREQuency:SPAN command. This command is valid for the zoom function for the frequency vs. frequency axis and the frequency vs. time axis.
- [Parameter] <real> = Frequency (GHz/MHz/kHz/Hz)
- [Query reply] NR3 (Real value: Unit Hz)
- [Example] Call ibwrt (analyzer%, ":DISP:TRAC:X:ZOOM:MODE ZMFF")

  '------------- Change the zoom freq. to 1GHz ---------------------------------

  Call ibwrt (analyzer%,":DISP:TRAC:X:ZOOM:FREQ:CENT 1GHZ")
- [Relevant commands] :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:MODE

  :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:FREQuency :SPAN

5.9.19 :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:FREQuency:SPAN

- [Command syntax] :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:FREQuency :SPAN < real >

  :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:FREQuency :SPAN?
- [Function description] Specifies the zoom width when the waveform zoom function is turned on

  Based on the zoom frequency specified using the zoom function, specifies the frequency span to be zoomed.

  This command is valid only for the zoom function for the frequency vs. frequency axis.
- [Parameter] <real> = Frequency (GHz/MHz/kHz/Hz)
- [Query reply] NR3 (Real value: Unit Hz)
- [Example] Call ibwrt (analyzer%,":DISP:TRAC:X:ZOOM:FREQ:SPAN 10MHZ")
- [Relevant commands] :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:MODE

  :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:FREQuency :CENTer

5.9.20 :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:TIME:DELay

- [Command syntax] :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:TIME :DELay < real >

  :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:TIME:DELay?
- [Function description] Specifies the zoom time position when the waveform zoom function is turned on

  Specifies the time position as the start position on the time axis to be zoomed using the zoom function.

  When the zoom function is turned on, sets the position on the time axis on the lower screen by sending this command.

  At this time position as the start position, the time width is zoomed into the zoom time width specified by the DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:TIME:WIDTh command.
- [Parameter] < real > = time (s/ms/μs/ns)
- [Query reply] NR3 (Real value: Unit s)
- [Example] Call ibwrt (analyzer%, ":DISP:TRAC:X:ZOOM:TIME:DEL 10MS")
- [Relevant commands] :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:MODE

  :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:TIME:WIDTh

5.9.21 :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:TIME:WIDTh

- [Command syntax] :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:TIME :WIDTh < real >
  :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:TIME:WIDTh?
- [Function description] Specifies the zoom width when the waveform zoom function is turned on

  Specifies the time width to be zoomed by setting the zoom start time specified using the zoom function as the start point.
  This command is valid only for the zoom function for the time vs. time axis.
- [Parameter] < real > = time (s/ms/μs)
- [Query reply] NR3 (Real value: Unit s)
- [Example] Call ibwrt (analyzer%, ":DISP:TRAC:X:ZOOM:TIME:WIDT 100MS")
- [Relevant commands] :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:MODE
  :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:ZOOM:TIME:DELay

5.9.22 :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:RLEVel

- [Command syntax] :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe] :RLEVel < real >
  :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:RLEVel?
- [Function description] Sets the reference level

  Sets the reference level.
- [Parameter] < real > = Reference level value (dBm)
  The value of the reference level depends on the unit of the vertical axis.
- [Query reply] NR3 (real number value in dBm)
- [Example] Call ibwrt (analyzer%, ":DISP:TRAC:Y:RLEV 10DBM")
- [Relevant commands] :UNIT<ch>:POWer<screen>

5.9.23 :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:RLEVel:OFFSet

- [Command syntax] :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:RLEVel :OFFSet < real >

  :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:RLEVel :OFFSet?
- [Function description] Sets Offset value to reference level value

  Sets the Offset value to the reference level value.

  Once the Offset value becomes valid by the :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:RLEVel :OFFSet:STATe command, the Offset value is added to the set reference level value.

  Displayed reference level = Real reference level value + Offset value

  The Offset value affects all absolute level values such as the marker level value, etc.

  When the Offset value is already valid, for the reference level value specified using the reference level setting command, the real reference level of this instrument is set based on the calculation below:

  Actually set reference level = Value set using the command - Offset value
- [Parameter] < real > = Offset level value (dB)
- [Query reply] NR3 (Real value: Unit dB)

  Without depending on the display unit, a dB value is always returned.
- [Example] Call ibwrt (analyzer%, ":DISP:TRAC:Y:RLEV:OFFS 15DB")
- [Relevant commands] :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:RLEVel

  :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:RLEVel :OFFSet:STATe

5.9.24 :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:RLEVel:OFFSet :STATe

- [Command syntax]
  :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:RLEVel :OFFSet:STATe < bool >
  :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:RLEVel :OFFSet:STATe?
- [Function description]
  Sets the setting of the Offset value to the reference level value ON/OFF

  Sets whether the Offset value is valid or invalid for the reference level value.

  By setting the Offset value on using this command,
  the Offset value specified by
  the :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:RLEVel :OFFSet command becomes valid.

  The Offset value is applied to the level values used for all absolute values of this instrument, and it also affects the value read by the marker.

  By setting the Offset value off, the specified Offset value becomes invalid and it no longer affects any level value.
- [Parameter]
  < bool > = { OFF | ON }
  OFF: Invalidates the Offset level value
  ON: Validates the Offset level value
- [Query reply]
  { OFF | ON }
- [Example]
  '---- Set reference level offset and then activate it ------------------
  Call ibwrt (analyzer%, ":DISP:TRAC:Y:RLEV:OFFS 15DB")
  Call ibwrt (analyzer%, ":DISP:TRAC:Y:RLEV:OFFS:STAT ON")
- [Relevant commands]
  :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:RLEVel
  :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:RLEVel :OFFSet

5.9.25 :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:PDIVision

- [Command syntax] :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:PDIVision < real >

  :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:PDIVision?
- [Function description] Sets the value of one division when the log scale is displayed

  When the display is switched to the log scale display, sets the display unit for one division on the scale. The range can be set between 20 dB/div and 0.1 dB/div.

  For the range from 20 dB/div to 1 dB/div, the resolution is set to 1 dB, and for the range from 1 dB/div to 0.1 dB/div, the resolution is set to 0.1 dB.
- [Parameter] < real > = dB value (dB) per 1 div

  Setting range: 20 - 0.1
- [Query reply] NR3 (Real value: Unit dB)
- [Example] Call ibwrt (analyzer%, ":DISP:TRAC:Y:SPAC LOG")

  Call ibwrt (analyzer%, ":DISP:TRAC:Y:PDIV 15DB")
- [Relevant commands] :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:SPACing

5.9.26 :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:SPACing

- [Command syntax] :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:SPACing < type >

  :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:SPACing?
- [Function description] Sets the vertical scale type

  Sets the type of the vertical scale (log scale or linear scale).

  When the log scale is selected, all scales on the vertical axis are decided according to the value (dB/div) set by the :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:PDIVision command. When the linear scale is selected, the display unit system for the vertical axis is automatically switched to the VOLT unit system.
- [Parameter] < type > = { LOGarithmic | LINear }

  LOGarithmic: Sets the log scale

  LINear: Sets the linear scale
- [Query reply] { LOG | LIN }
- [Example] Call ibwrt (analyzer%, ":DISP:TRAC:Y:SPAC LOG")
- [Relevant commands] :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:PDIVision

  :UNIT<ch>:POWer<screen>

5.9.27 :DISPlay<ch>[:WINDow]:TRACe:CCDF:STATe

- [Command syntax] :DISPlay<ch>[:WINDow]:TRACe:CCDF:STATe <bool>
  :DISPlay<ch>[:WINDow]:TRACe:CCDF:STATe?
- [Function description] Sets the reference waveform display in the CCDF measurement to ON or OFF

  This command sets the reference waveform display in the CCDF measurement to ON or OFF.
  If ON is selected, the currently displayed CCDF waveform is acquired and displayed as the reference waveform.
  If OFF is selected, the reference waveform is deleted.
- [Parameter] <bool> = {OFF | ON}
  OFF: Waveform deletion
  ON: Waveform display
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":DISP:TRAC:CCDF:STAT ON")
- [Relevant commands] :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:CCDF

5.9.28 :DISPlay<ch>[:WINDow]:TRACe:CCDF:GAUSsian:STATe

- [Command syntax] :DISPlay<ch>[:WINDow]:TRACe:CCDF:GAUSsian:STATe <bool>
  :DISPlay<ch>[:WINDow]:TRACe:CCDF:GAUSsian:STATe?
- [Function description] Sets the ideal gaussian noise in the CCDF measurement to ON or OFF

  This command sets the ideal gaussian noise in the CCDF measurement to ON or OFF.
  If ON is selected, the ideal gaussian noise is displayed.
  If OFF is selected, the ideal gaussian noise is deleted.
- [Parameter] <bool> = {OFF | ON}
  OFF: Waveform deletion
  ON: Waveform display
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":DISP:TRAC:CCDF:GAUS:STAT ON")
- [Relevant commands] :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:CCDF

5.9.29 :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:CCDF

- [Command syntax] :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:CCDF <real>
  :DISPlay<ch>[:WINDow]:TRACe:X[:SCALe]:CCDF?
- [Function description] Sets the maximum horizontal axis value for the waveform display in the CCDF measurement

  This command sets the horizontal axis scale for the waveform display in the CCDF measurement.
- [Parameter] < real > = maximum horizontal axis value for the waveform display (dB)
  Setting range: 1 dB - 100 dB
- [Query reply] NR3 (Real value: Unit dB)
- [Example] Call ibwrt (analyzer%, ":DISP:TRAC:X:CCDF 40DB")
- [Relevant commands] :DISPlay<ch>[:WINDow]:TRACe:CCDF:STATe
  :DISPlay<ch>[:WINDow]:TRACe:CCDF:GAUSsian:STATe

5.10 Trace Command

This section describes the Trace subsystem.

In the Trace subsystem, the commands used to output the Trace waveform data in the SA mode are defined.

Command	Function	Reference Page
:TRACe<ch>		
[:DATA<screen>]?	Outputs the trace data	5-205

5.10.1 :TRACe<ch>[:DATA<screen>]?

- [Command syntax] :TRACe<ch>[:DATA<screen>]? < type >
- [Function description] Outputs the trace data

  Outputs the trace data of the specified trace. The format of the data to be output is defined by the FORMat:TRACe[:DATA] command and the FORMat:BORDer command.

  The data to be output is the level data at each frequency point (when the frequency axis is displayed) or each time point (when the time axis is displayed).
- [Parameter] < type > = { TRACE1 | TRACE2 | TRACE3 | TRACE4 }
- [Query reply] < block > or < ASCII group>
- [Example] Result$ = Space$(50000)

```
'---------- Set data output mode to ASCII ----------------
Call ibwrt (analyzer%, ":FORM:TRAC ASC")

Call ibwrt (analyzer%, ":TRAC:DATA? TRACE1")
Call ibrd (analyzer%, Result$)
```

- [Relevant commands] :FORMat:TRACe[:DATA]

  :FORMat:BORDer

5.11 Format Commands

This section describes the Format subsystem.

In the Format subsystem, the commands used to specify the data format in which the Trace waveform data is output in the SA mode are defined.

Command	Function	Reference Page
:FORMat		
:BORDer	Sets the output byte order of the trace data	5-206
:TRACe		
[:DATA]	Sets the output format of the trace data	5-207

5.11.1 :FORMat:BORDer

- [Command syntax] :FORMat:BORDer < type >
  :FORMat:BORDer?
- [Function description] Sets the output byte order of the trace data

  Sets the byte order when REAL format is selected as the output format for the trace data.

  There are two types of formats available: NORMal and SWAPped.

  For example, suppose the original data is a data row of "12345678" in hexadecimal format.

  When NORMal is specified, the data is output in the order of "12345678."

  When SWAPped is specified, the data is output in the order of "21436587."
- [Parameter] < type > = { NORMal | SWAPped }
  NORMal: Outputs high-order bytes first and then low-order bytes
  SWAPped: Outputs low-order bytes first and then high-order bytes
- [Query reply] { NORM | SWAP }
- [Example]

```
'------- Select trace format to REAL 32bits type ----------
Call ibwrt (analyzer%, ":FORM:TRAC REAL,32")
Call ibwrt (analyzer%, ":FORM:BORD SWAP")
'------- Start reading the trace --------------
Call ibwrt (analyzer%, ":TRAC:DATA? TRACE2")
```

- [Relevant commands] :TRACe<ch>[:DATA<screen>]?
  :FORMat:TRACe[:DATA]

5.11.2 :FORMat:TRACe[:DATA]

- [Command syntax] :FORMat:TRACe[:DATA] < type > , < int >
  :FORMat:TRACe[:DATA]?
- [Function description] Sets the output format of the trace data

  Sets the output format in which the trace data is output. There are two types of formats available: Binary block format and ASCII format. Their respective output formats are as follows:

  Binary block

  Data example:

  #48008xxxxxxxxyyyyyyyyzzzzzzzz.......

  Description:

  #4: Indicates 4-digit data.

  8008: Indicates data with a length of 4 digits

  xxxxxxxx: The level data at the first frequency point or time point (IEEE 64-bit double-precision floating-point format, 8 bytes/data length)

  ASCII:

  Data example:

  1.2345678e+1,-1.3456789e+1,...

  Description:

  -1.2345678e+1

  The level data at the first frequency point or time point (data in NR3 format in ASCII)

  Repetition of data items in the number of data items first delimited by a comma (,) is performed.
- [Parameter] < type > = { REAL | ASCii }

  REAL: Select Binary block format.

  ASCii: Select ASCII format.

  < int > = { 32 | 64 } | { 8 | 9 | 10 | ... | 21 | 22 }

  {32 | 64 } : When REAL is selected

  { 8 | 9 | 10 | ... | 21 | 22 }: When ASCii is selected

- [Query reply] { REAL | ASC } , {{ 32 | 64 } | { 8 | 9 | 10 | ... | 21 | 22 } }
  {32 | 64 } : When REAL is selected
  { 8 | 9 | 10 | ... | 21 | 22 }: When ASCii is selected
- [Example] Call ibwrt (analyzer%, ":FORM:TRAC ASC")
- [Relevant commands] :TRACe<ch>[:DATA<screen>]?
  :FORMat:BORDer

MEMO: The output format of trace data is determined by combining the :FORMat:BORDer command and the :FORMat:TRACe[:DATA] command as follows:

FORMat:TRACe[:DATA]	FORMat:BORDer	
	NORMal	SWAPped
ASCii	ASCII	
REAL, 32	IEEE 32bit binary	IEEE 32bit binary Changing the byte order
REAL, 64	IEEE 64bit binary	IEEE 64bit binary Changing the byte order

5.12 Calibration Commands

This section describes the Calibration subsystem.

In the Calibration subsystem, the commands related to calibration of this instrument, such as the execution of calibration and execution mode of calibration, are defined.

Command	Function	Reference Page
:CALibration		
:SANalyzer	Execution of calibration using the external CAL signal (including RF ATT)	5-209
:ATTenuation		
:NONE	Execution of calibration using the internal CAL signal (excluding RF ATT)	5-210

5.12.1 :CALibration:SANalyzer

- [Command syntax] :CALibration:SANalyzer
- [Function description] Executes calibration using the external CAL signal

  Executes internal calibration (including RF attenuators) using the CAL signal output to the external terminal of this instrument. When calibration operation is terminated, the CALIbration bit in the standard operation status register is set.
- [Parameter] None
- [Query reply] None
- [Example]

```
Call ibwrt (analyzer%, ":STAT:OPER:ENAB 1")' Enable the cal bit
Call ibwrt (analyzer%, ":CAL:SAN")            ' Start calibration

Esr$ = Space$(20)
Do                                             ' Loop until end of cal.
  Call ibwrt(analyzer%, ":STAT:OPER:EVEN?")' Read stat reg.
  Call ibrd(analyzer%, Esr$)
Loop Undil ((VAL(Esr$) AND 1) > 0)
```

- [Relevant commands]

5.12.2 :CALibration:SANalyzer:ATTenuation:NONE

- [Command syntax] :CALibration:SANalyzer:ATTenuation:NONE
- [Function description] Executes calibration using the internal CAL signal

  Executes internal calibration (excluding calibration for RF attenuators) using the calibration signal provided in this instrument. When calibration operation is terminated, the CALIbration bit in the standard operation status register is set.
- [Parameter] None
- [Query reply] None
- [Example]

```
Call ibwrt (analyzer%, ":STAT:OPER:ENAB 1")' Enable the cal bit
Call ibwrt (analyzer%, ":CAL:SAN:ATT:NONE") ' Start calibration

Esr$ = Space$(20)
Do                                          ' Loop until end of cal.
  Call ibwrt(analyzer%, ":STAT:OPER:EVEN?")' Read stat reg.
  Call ibrd(analyzer%, Esr$)
Loop Undil ((VAL(Esr$) AND 1) > 0)
```

- [Relevant commands]

5.13 Mass Memory Commands

This section describes the Mass Memory subsystem.

In the Mass Memory subsystem, the commands related to storing and loading of the settings of this instrument are defined.

Command	Function	Reference Page
:MMEMory		
:STORe		
:STATe	Execution of the Save function used to save the settings of this instrument	5-212
:STANdard		
:STATe	Executes the SAVE function for the Standard information	5-212
:LOAD		
:STATe	Execution of the LOAD function used to load the settings of this instrument	5-213
:STANdard		
:STATe	Executes the LOAD function for the Standard information	5-213
:SELect		
:ITEM		
:SETup	Setting of Save conditions of the setting parameters	5-214
:TRACe	Setting of Save conditions of the trace data	5-214
:NCORrection	Setting of Save conditions of the correction data for the Normalize function	5-215
:LIMit	Setting of Save conditions of the limit line data for the limit line function	5-215
:CORRection	Setting of Save conditions of the level correction data for the input level correction function	5-216
:SPURious	Setting of Save conditions of the setting parameters for the Spurious measurement function	5-216
:SEMask	Setting of Save conditions of the setting parameters for the Spectrum Emission Mask measurement function	5-217

5.13.1 :MMEMory:STORe:STATe

- [Command syntax] :MMEMory:STORe:STATe < int >
- [Function description] Execution of the Save function used to save the settings of this instrument

  Saves the settings of this instrument with the specified file name.

  It is necessary to determine whether the settings are to be saved or not by using the ON/OFF function of the Save switch in advance. Specifies the file number to save the relevant parameter and data according to the states of the switches. The file named ["FILE" + file number] is saved.

  The Save file is stored in the internal device.
- [Parameter] < int > = File number

  File No.: 0 - 9999
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":MMEM:STOR:STAT 1")
- [Relevant commands] :MMEMory:LOAD:STATe

  Commands related to :MMEMory:SELect:ITEM

5.13.2 :MMEMory:STORe:STANdard:STATe

- [Command syntax] :MMEMory:STORe:STANdard:STATe < int >
- [Function description] Executes the SAVE function for the Standard information.

  Saves the Standard information with the specified file number.

  The destination of the Save file is the internal device.
- [Parameter] < int > = File number

  File No.: 0 - 9999
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":MMEM:STOR:STAN:STAT 1")
- [Relevant commands] :MMEMory:LOAD:STANdard:STATe

  :SYSTem:SELect:STANdard

5.13.3 :MMEMory:LOAD:STATe

- [Command syntax] :MMEMory:LOAD:STATe < int >
- [Function description] Execution of the LOAD function used to load the settings of this instrument

  Executes the LOAD function to load the saved settings of this instrument from the file in which they are saved.

  The same as the Save function, it is possible to load only from the internal device.
- [Parameter] < int > = File number

  File No.: 0 - 9999
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":MMEM:LOAD:STAT 2")

  Call ibwrt (analyzer%, "*WAI") ' Wait until end of loading
- [Relevant commands] :MMEMory:STORe:STATe

5.13.4 :MMEMory:LOAD:STANdard:STATe

- [Command syntax] :MMEMory:LOAD:STANdard:STATe < int >
- [Function description] Executes the LOAD function for the Standard information.

  Executes the Load function from the file which saved the Standard information. The load function can, as with the Save function, only be executed from the internal device.
- [Parameter] < int > = File number

  File No.: 0 - 9999
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":MMEM:LOAD:STAN:STAT 2")

  Call ibwrt (analyzer%, "*WAI") ' Wait until end of loading
- [Relevant commands] :MMEMory:STORe:STANdard:STATe

  :SYSTem:SELect:STANdard

5.13.5 :MMEMory:SELect:ITEM:SETup

- [Command syntax] :MMEMory:SELect:ITEM:SETup < bool >
  :MMEMory:SELect:ITEM:SETup?
- [Function description] Setting of Save conditions of the setting parameters

  Sets whether or not to save the setting parameters when the Save function is executed. When Save of a setting parameter is turned off by using this command, the parameter is not saved.

  The setting parameters are set to be saved by default.
- [Parameter] < bool > = { OFF | ON }
  OFF: Setting parameters are not saved
  ON: Setting parameters are saved
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":MMEM:SEL:ITEM:SET ON")
- [Relevant commands] :MMEMory:STORe:STATe

5.13.6 :MMEMory:SELect:ITEM:TRACe

- [Command syntax] :MMEMory:SELect:ITEM:TRACe < bool >
  :MMEMory:SELect:ITEM:TRACe?
- [Function description] Setting of Save conditions of the trace data

  Sets whether or not to save four sets of trace data when the Save function is executed. When Save of a setting parameter is turned off by using this command, the parameter is not saved.

  The data is not set to be saved by default.
- [Parameter] < bool > = { OFF | ON }
  OFF: Trace data is not saved
  ON: Trace data is saved
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":MMEM:SEL:ITEM:TRAC ON")
  Call ibwrt (analyzer%, ":MMEM:SEL:ITEM:TRAC OFF")
- [Relevant commands] :MMEMory:STORe:STATe

5.13.7 :MMEMory:SELect:ITEM:NCORrection

- [Command syntax] :MMEMory:SELect:ITEM:NCORrection < bool >
  :MMEMory:SELect:ITEM:NCORrection?
- [Function description] Setting of Save conditions of the correction data for the Normalize function

  Sets whether or not to save four sets of correction data for the Normalize function when the Save function is executed. When Save of a setting parameter is turned off by using this command, the parameter is not saved.

  The data is not set to be saved by default.
- [Parameter] < bool > = { OFF | ON }
  OFF: Normalize correction data is not saved
  ON: Normalize correction data is saved
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":MMEM:SEL:ITEM:NCOR ON")
  Call ibwrt (analyzer%, ":MMEM:SEL:ITEM:NCOR OFF")
- [Relevant commands] :MMEMory:STORe:STATe

5.13.8 :MMEMory:SELect:ITEM:LIMit

- [Command syntax] :MMEMory:SELect:ITEM:LIMit < bool >
  :MMEMory:SELect:ITEM:LIMit?
- [Function description] Setting of Save conditions of the limit line data for the limit line function

  Sets whether or not to save two sets of limit line data for the limit line function when the Save function is executed. When Save of a setting parameter is turned off by using this command, the parameter is not saved.

  The data is not set to be saved by default.
- [Parameter] < bool > = { OFF | ON }
  OFF: Limit line data is not saved
  ON: Limit line data is saved
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":MMEM:SEL:ITEM:LIM ON")
- [Relevant commands] :MMEMory:STORe:STATe

5.13.9 :MMEMory:SELect:ITEM:CORRection

- [Command syntax] :MMEMory:SELect:ITEM:CORRection < bool >
  :MMEMory:SELect:ITEM:CORRection?
- [Function description] Setting of Save conditions of the level correction data for the input level correction function

  Sets whether or not to save the level correction data for the input level correction function when the Save function is executed. When Save of a setting parameter is turned off by using this command, the parameter is not saved.

  The data is not set to be saved by default.
- [Parameter] < bool > = { OFF | ON }
  OFF: Input level correction data is not saved
  ON: Input level correction data is saved
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":MMEM:SEL:ITEM:CORR ON")
- [Relevant commands] :MMEMory:STORe:STATe

5.13.10 :MMEMory:SELect:ITEM:SPURious

- [Command syntax] :MMEMory:SELect:ITEM:SPURious < bool >
  :MMEMory:SELect:ITEM:SPURious?
- [Function description] Setting of Save conditions of the setting parameters for the Spurious measurement function

  Sets whether or not to save the setting parameters for the Spurious measurement function when the Save function is executed. When Save of a setting parameter is turned off by using this command, the parameter is not saved.

  The setting parameters are not set to be saved by default.
- [Parameter] < bool > = { OFF | ON }
  OFF: Spurious measurement setting parameters are not saved
  ON: Spurious measurement setting parameters are saved
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":MMEM:SEL:ITEM:SPUR ON")
- [Relevant commands] :MMEMory:STORe:STATe

5.13.11 :MMEMory:SELect:ITEM:SEMask

• [Command syntax]	:MMEMory:SELect:ITEM:SEMask < bool > :MMEMory:SELect:ITEM:SEMask?
• [Function description]	Setting of Save conditions of the setting parameters for the Spectrum Emission Mask measurement function Sets whether or not to save the setting parameters for the Spectrum Emission Mask measurement function when the Save function is executed. When Save of a setting parameter is turned off by using this command, the parameter is not saved. The setting parameters are not set to be saved by default.
• [Parameter]	< bool > = { OFF | ON } OFF: Spectrum Emission Mask measurement setting parameters are not saved ON: Spectrum Emission Mask measurement setting parameters are saved
• [Query reply]	{ OFF | ON }
• [Example]	Call ibwrt (analyzer%, ":MMEM:SEL:ITEM:SEM ON")
• [Relevant commands]	:MMEMory:STORe:STATe

5.14 Calculate Commands

This section describes the Calculate subsystem.

The Calculate subsystem defines those commands that are related to analysis by marker functions or limit line functions.

MEMO: ***The following notations are used for convenience within the Calculate subsystem.***
<mkr>:
Written in the command header and indicates the active marker number of the command.
The marker number ranges from 1 to 10.
An equivalent notation is {1|2|3|4|5|6|7|8|9|10}.
<area>:
Written in the command header and indicates the active area number of the command.
The area number ranges from 1 to 10.
An equivalent notation is {1|2|3|4|5|6|7|8|9|10}.

Command	Function	Reference Page
:CALCulate<ch>		
:MARKer<screen>		
[:NUMBer<mkr>]		
:ACTive	Specifies an operation target marker (active marker) among the multimarkers	5-225
:FUNCtion		
[:STATe]	Sets ON or OFF marker functions	5-226
[:NUMBer<mkr>]		
[:STATe]	Sets ON or OFF the specified multimarker	5-226
:X	Specifies a frequency position and a time position of the specified multimarker	5-227
:ABSolute?	Reads the absolute values (frequency and time) of the specified multimarker	5-228
:Y?	Reads the level value of the specified multimarker	5-229
:ABSolute?	Reads the absolute level value of the specified multimarker	5-228
:MAXimum		
[:PEAK]	Searches the maximum peak point using the specified multimarker	5-230
:NEXT	Searches the next peak using the specified multimarker	5-231
:LEFT	Searches the next peak in the left direction using the specified multimarker	5-232
:RIGHt	Searches the next peak in the right direction using the specified multimarker	5-234

Command	Function	Reference Page
:CALCulate<ch>		
:MARKer<screen>		
[:NUMBer<mkr>]		
:MINimum		
[:PEAK]	Searches the minimum peak using the specified multimarker	5-236
:NEXT	Searches the next minimum peak using the specified multi-marker	5-237
:TRACe	Moves the specified marker to the specified trace	5-238
:RESet	Sets all the markers except marker No. 1 off	5-239
:LIST		
[:STATe]	Displays the marker list of the markers displayed	5-239
:MAXimum		
:LIST	Searches peak points and displays a marker list	5-240
:CONTinuous	Sets ON or OFF the continuous peak point search mode	5-240
:DELTa	Specifies a deviation for peak point judgment at the time of peak point search	5-241
:STEP	Sets a marker step size	5-242
:AUTO	Sets a marker step size mode	5-242
:SEARch		
:X		
:MODE	Sets a peak search range specification mode on the horizontal axis	5-243
:POSition	Specifies the reference position of the peak search range on the horizontal axis	5-244
:WIDTh	Specifies a search width from the reference position of the peak search range on the horizontal axis	5-245
:COUPling	Sets a move mode of the peak search range on the horizontal axis	5-246
:Y		
:MODE	Sets a peak search range specification mode on the vertical axis	5-247
:DLINe	Specifies the peak search range with Display Line used as the reference	5-248
:LUPPer	Specifies the peak search range with Limit Line1 used as the reference	5-249
:LLOWer	Specifies the peak search range with Limit Line2 used as the reference	5-250

Command	Function	Reference Page
:CALCulate<ch>		
:MARKer<screen>		
:MINNer<area>	Sets ON or OFF the display of the marker frame for the Multi Inner Peak Search function	5-251
:MAXimum		
:PEAK	Executes peak search in all marker frames by the Multi Inner Peak Search function	5-252
:LIST?	Reads the peak value in all marker frames by the Multi Inner Peak Search function	5-253
:X		
:POSition	Specifies a position on the horizontal axis in the marker frame by the Multi Inner Peak Search function	5-254
:WIDTh	Specifies the width of the marker frame on the horizontal axis by the Multi Inner Peak Search function	5-255
:Y	Sets the specified marker frame and the vertical axis search range mode of the Multi Inner Peak Search function	5-256
:LOWer	Specifies the lower position of the vertical axis of the marker frame by the Multi Inner Peak Search function	5-257
:UPPer	Specifies the upper position of the vertical axis of the marker frame by the Multi Inner Peak Search function	5-258
:CALCulate<ch>		
:MARKer<screen>		
[:NUMBer<mkr>]		
:SET		
:CENTer	Sets the marker frequency as the center frequency	5-258
:RLEVel	Sets the marker level value as the reference level	5-259
:CENTer		
:STEP	Sets the marker frequency as the center frequency step size	5-259
:MARKer		
:STEP	Sets the marker frequency as the marker frequency step size	5-260
:MAXimum		
:SET		
:CENTer	Sets the marker frequency as the center frequency after peak search is performed	5-260
:RLEVel	Sets the marker level value as the reference level after peak search is performed	5-261
:ROBJect	Specifies a reference for the relative value display of the marker	5-266

Command	Function	Reference Page
:CALCulate<ch>		
:DELTamarker<screen>		
[:NUMBer<mkr>]		
:SET		
:CENTer	Sets the delta marker frequency as the center frequency	5-261
:STEP	Sets the delta marker frequency as the center frequency step size	5-262
:SPAN	Sets the delta marker frequency as the span frequency	5-262
:MARKer		
:STEP	Sets the delta marker frequency as the marker step size	5-263
[:STATe]	Sets ON or OFF the display of the delta marker	5-263
:FIXed		
[:STATe]	Sets ON or OFF the display of the fixed delta marker	5-264
:MAXimum		
[:PEAK]	Searches the peak and sets a fixed marker	5-264
:INVerse		
[:STATe]	Sets ON or OFF the display of the (1/delta) marker	5-265
:X?	Reads the frequency value of the delta marker	5-265
:Y?	Reads the level value of the delta marker	5-266
:CALCulate<ch>		
:MARKer<screen>		
[:NUMBer<mkr>]		
:FCOunt		
[:STATe]	Sets ON or OFF the frequency counter function	5-267
:FREQuency?	Reads the results of measurement in frequency counter function	5-268
:CALCulate<ch>		
:MARKer<screen>		
[:NUMBer<mkr>]		
:STRack		
[:STATe]	Sets ON or OFF the Signal Tracking function	5-269

Command	Function	Reference Page
:CALCulate<ch> :MARKer<screen> [:NUMBer<mkr>] :FUNCtion :XDBDown		
:LEVel	Sets the Down width of the X dB Down function	5-272
:MODE	Selects a display mode after execution of the X dB Down function	5-274
:RMARker [:STATe]	Sets ON or OFF the display of the reference marker when the X dB Down function is executed	5-276
:CONTinuous [:STATe]	Sets ON or OFF the continuous X dB Down function	5-275
:XDBDown	Executes the X dB Down function	5-269
:LEFT	Executes the X dB Down Left function	5-270
:RIGHt	Executes the X dB Down Right function	5-271
:PEAK	Executes the X dB Down function after peak search	5-273
:CALCulate<ch> :MARKer<screen> :FUNCtion :NOISe	Reads the results in the Noise/Hz measurement.	5-279
:BWIDth	Sets the noise measurement bandwidth used in the Noise/Hz measurement.	5-277
:CALCulate<ch> :MARKer<screen> [:NUMBer<mkr>] :FUNCtion :STATe	Sets ON or OFF the Noise/Hz function	5-277
:MODE	Selects an operation mode for Noise/Hz function	5-278

Command	Function	Reference Page
:CALCulate<ch>		
:MARKer<screen>		
[:NUMBer<mkr>]		
:FUNCtion		
:AM?	Reads the results of %AM measurement	5-279
:STATe	Sets ON or OFF %AM measurement	5-280
:CALCulate<ch>		
:DLINe<screen>	Sets the display position of the display line	5-280
:STATe	Sets ON or OFF the display of the display line	5-281
:RLINe<screen>	Sets the display position of the reference line	5-281
:STATe	Sets ON or OFF the display of the reference line	5-282
:CALCulate<ch>		
:LIMit<screen>		
:AUTO	Automatically adjusts the level position of the limit line	5-282
:CONTrol		
:X		
:DOMain	Selects the domain of the limit line to be used	5-283
:MODE	Selects horizontal axis data attributes of the limit line to be used	5-285
:REFerence	Specifies the reference position when the horizontal axis relative value attributes of the limit line to be used are selected	5-286
:USER	Sets the user-defined reference position when the horizontal axis relative value attributes of the limit line to be used are selected	5-287
:OFFSet	Sets the offset value when the horizontal axis relative value attributes of the limit line to be used are selected	5-288
:Y		
:MODE	Selects vertical axis data attributes of the limit line to be used	5-289
:REFerence	Specifies the reference position when the vertical axis relative value attributes of the limit line to be used are selected	5-290
:USER	Sets the user-defined reference position when the vertical axis relative value attributes of the limit line to be used are selected	5-291
:OFFSet	Sets the offset value when the vertical axis relative value attributes of the limit line to be used are selected	5-291

Command	Function	Reference Page
:CALCulate<ch>		
:LIMit<screen>		
:FAIL?	Reads the Pass/Fail judgment by the limit line	5-292
:STATe	Sets ON or OFF the Pass/Fail judgment by the limit line	5-298
{:UPPer | :LOWer}		
:COPY	Copies the data for limit line 1 to limit line 2 or Copies the data for limit line 2 to limit line 1	5-293
:DATA	Inputs data to limit line 1 or 2	5-294
:DELete	Deletes the data from Limit Line Table 1 or 2	5-295
:PASS	Sets the judgment condition for Pass/Fail judgment by Limit Line Table 1 or 2	5-296
:STATe	Sets ON or OFF the display of limit line 1 or 2	5-297
:CALCulate<ch>		
:WINDow<screen>		
:POSition	Sets the position of the Measuring Window	5-299
:WIDTh	Sets the width of the Measuring Window	5-300
:STATe	Sets ON or OFF the display of the Measuring Window	5-301
:CURSor<screen>		
:ANCHor	Sets ON or OFF the anchor function of the X and Y cursors	5-301
:X	Sets the display position of the X cursor when X and Y cursors are displayed	5-302
:Y	Sets the display position of the Y cursor when X and Y cursors are displayed	5-302
:STATe	Sets ON or OFF the display of the X and Y cursors	5-303

5.14.1 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive

- [Command syntax]
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive?
- [Function description]
  Specifies an operation target marker (active marker) among the multimarkers

  This command is used to specify the active marker among the ten multimarkers. The marker numbers that can be specified range from 1 to 10. If any marker that has already been specified is on, that marker is the active marker. If the specified marker is off, that marker is set on so that it will be the active marker.

  This command can determine an active marker for any other marker function command that omits the marker number of the active marker.
- [Parameter]
  None
- [Query reply]
  NR1 (Integer: marker No.)
  1 - 10
- [Example]
  '-------------- Activate marker No.1 and 2 -------------------
  Call ibwrt (analyzer%, ":CALC:MARK:NUMB1 ON")
  Call ibwrt (analyzer%, ":CALC:MARK:NUMB2 ON")
  '-------------- Move active marker to No.1 -------------------
  Call ibwrt (analyzer%, ":CALC:MARK:NUMB1:ACT")

  '-------------- Set the freq. of marker No.1 to 1GHz --------
  Call ibwrt (analyzer%, ":CALC:MARK:X 1GHZ")
- [Relevant commands]
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>][:STATe]
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:X
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:Y?
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MAXimum[:PEAK]
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MAXimum:NEXT
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MAXimum:LEFT
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MAXimum:RIGHt
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MINimum[:PEAK]
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MINimum:NEXT

5.14.2 :CALCulate<ch>:MARKer<screen>:FUNCtion[:STATe]

- [Command syntax] :CALCulate<ch>:MARKer<screen>:FUNCtion[:STATe] < bool >
  :CALCulate<ch>:MARKer<screen>:FUNCtion[:STATe]?
- [Function description] Sets ON or OFF marker functions

  This command is used to set ON or OFF the marker functions. The setting of marker ON displays marker No.1. The setting of marker OFF turns off all markers on display.
- [Parameter] < bool > = { OFF | ON }
  OFF: Display of all markers is off
  ON: Display of Marker No.1 is on
- [Query reply] { OFF | ON }
  OFF: All markers are in the off state
  ON: Multimarkers are or the delta marker is in the on state
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:FUNC ON")
- [Relevant commands] :CALCulate<ch>:DELTamarker<screen>[:STATe]
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>][:STATe]
  :CALCulate<ch>:MARKer<screen>:MAXimum:LIST

5.14.3 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>][:STATe]

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>][:STATe] <bool>
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>][:STATe]?
- [Function description] Sets ON or OFF the specified multimarker

  This command is used to set ON or OFF the specified marker among the ten multimarkers. This command is effective for the specified marker only.
- [Parameter] < bool > = { OFF | ON }
  OFF: Display of the specified marker is off
  ON: Display of the specified marker is on
- [Query reply] { OFF | ON }
- [Example] '-------------- Activate marker No.1 and 2 ------------------
  Call ibwrt (analyzer%, ":CALC:MARK:NUMB1 ON")
  Call ibwrt (analyzer%, ":CALC:MARK:NUMB2 ON")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:X

5.14.4 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:X

- [Command syntax]
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:X < real >
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:X?

- [Function description]
  Specifies a frequency position and a time position of the specified multi-marker

  This command is used to specify a frequency position and a time position of the active marker among the ten multimarkers. The marker numbers that can be specified range from 1 to 10. If a marker number is omitted, the marker that is active at that time point is assumed.

  If any marker that has already been specified is on, that marker is the active marker. If the specified marker is off, that marker is set on so that it will be the active marker.

  When this command is sent as a query command, an absolute value is returned if the delta marker function is off, while a relative value is returned if the delta marker function is on.

- [Parameter]
  < real > = Frequency (GHz/MHz/kHz/Hz) or time (s/ms/μs/ns)
  Frequency span: Specify a frequency.
  Zero span: Specify time.

- [Query reply]
  NR3 (real number value: frequency or time)
  When the delta marker function is OFF:
  Frequency span in Hz
  Zero span in s
  When the delta marker function is ON:
  Frequency span: relative frequency from the delta marker (Hz)
  Zero span: relative time from the delta marker (s)

- [Example]

```
'-------------- Activate marker No.1 and 2 ------------------
Call ibwrt (analyzer%, ":CALC:MARK:NUMB1 ON")
Call ibwrt (analyzer%, ":CALC:MARK:NUMB2 ON")

'-------------- Set the freq. of marker No.1 to 1GHz --------
Call ibwrt (analyzer%, ":CALC:MARK:NUMB1:X 1GHZ")
```

- [Relevant commands]
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>][:STATe]
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive
  :CALCulate<ch>:DELTamarker<screen>[:STATe]
  :CALCulate<ch>:DELTamarker<screen>:FIXed[:STATe]
  :CALCulate<ch>:DELTamarker<screen>:FIXed:MAXimum[:PEAK]

5.14.5 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:X:ABSolute?

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:X:ABSolute?
- [Function description] Reads the absolute values (frequency and time) of the specified multi-marker

  This command is used to read the absolute values (frequency or time) of the specified marker. This command is used also to read the absolute values of the frequency or time of the specified marker when the delta marker function is on.
- [Parameter] None
- [Query reply] NR3 (real number value of frequency in Hz or time in s)
- [Example] Mkr$ = Space$(100)

  Call ibwrt (analyzer%, ":CALC:MARK:NUMB2:X:ABS?")

  Call ibrd(analyzer%, Mkr$)
- [Relevant commands] :CALCulate<ch>:DELTamarker<screen>[:STATe]

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:Y:ABSolute?

5.14.6 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:Y:ABSolute?

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:Y:ABSolute?
- [Function description] Reads the absolute level values of the specified multimarker

  This command is used to read the absolute level values of the specified marker.

  This command is used also to read the absolute value of the level value of the specified marker when the delta marker function is on.
- [Parameter] None
- [Query reply] NR3 (real number value of level in dBm)
- [Example] MkrFreq$ = Space$(100)

  Call ibwrt (analyzer%, ":CALC:MARK:NUMB2:Y:ABS?")

  Call ibrd (analyzer%, MkrFreq$)
- [Relevant commands] :CALCulate<ch>:DELTamarker<screen>[:STATe]

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:X:ABSolute?

  :UNIT<ch>:POWer<screen>

5.14.7 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:Y?

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:Y?
- [Function description] Reads the level value of the specified multimarker

  This command is used to read the level value of the specified marker among the ten multimarkers. The marker numbers that can be specified range from 1 to 10.

  If the specified marker is on, the level value of the position of that marker is returned. The unit of the level value to be read follows the unit system specified by the unit specification command :UNIT<ch>:POWer<screen>.

  If a marker number is omitted, the level value of the marker that is active at that time is read.

  If the specified marker is off, a query error occurs and the message <NO DATA>, which indicates the value is indefinite, is returned.

  An absolute value is returned if the delta marker function is off, while a relative value is returned if the delta marker function is on.
- [Parameter] None
- [Query reply] NR3 (real number value of level)

  When the delta marker function is off:absolute value of level in dBm

  When the delta marker function is on:relative level value from the delta marker in dB
- [Example]

```
Mkr1$ = Space$(30)
'--------------- Activate marker No.1 and 2 -------------------
Call ibwrt (analyzer%, ":CALC:MARK:NUMB1 ON")
'--------------- Set the freq. of marker No.1 to 1GHz --------
Call ibwrt (analyzer%, ":CALC:MARK:NUMB1:X 1GHZ")

'--------------Read the level of marker No.1------------------
Call ibwrt(analyzer%, ":CALC:MARK:Y?")
Call ibrd(analyzer%, Mkr1$)
OutputMsgs " MKR #1 = "&Mkr1$
```

- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>][:STATe]

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:X

  :UNIT<ch>:POWer<screen>

  :CALCulate<ch>:DELTamarker<screen>[:STATe]

  :CALCulate<ch>:DELTamarker<screen>:FIXed[:STATe]

  :CALCulate<ch>:DELTamarker<screen>:FIXed:MAXimum[:PEAK]

5.14.8 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MAXimum[:PEAK]

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MAXimum[:PEAK]
- [Function description] Searches the maximum peak point using the specified multimarker

  This command is used to search the maximum peak point of the waveform at the time it is executed and places the specified marker on that point. If a marker number is omitted, the marker that is active at that time is placed on the peak point.

  Even if the specified marker is off, the command searches the maximum peak point, displays the marker, and places it on the peak point.

  An execution error occurs if this command is sent with the search range limited by the :CALCulate<ch>:MARKer<screen>:SEARch command group and no waveforms exist in that range. In such a case, the marker is not displayed or remains where it was before this command is executed.
- [Parameter] None
- [Query reply] None
- [Example]

```
'-------------- Activate marker No.1 and 2 ------------------
Call ibwrt (analyzer%, ":CALC:MARK:NUMB1 ON")
'------------------------- Search peak point --------------------
Call ibwrt (analyzer%, ":CALC:MARK:NUMB1:MAX")

'--------------Read the level of peak point ------------------
Call ibwrt(analyzer%, ":CALC:MARK:Y?")
Call ibrd(analyzer%, Mkr1$)
OutputMsgs " Peak Level = "&Mkr1$
```

- [Relevant commands]
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>][:STATe]
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive
  :CALCulate<ch>:MARKer<screen>:SEARch:X:MODE
  :CALCulate<ch>:MARKer<screen>:SEARch:X:POSition
  :CALCulate<ch>:MARKer<screen>:SEARch:X:WIDTh
  :CALCulate<ch>:MARKer<screen>:SEARch:X:COUPling
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:MODE
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:DLINe
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LUPPer
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LLOWer
  :CALCulate<ch>:MARKer<screen>:MAXimum:DELTa

5.14.9 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MAXimum:NEXT

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MAXimum:NEXT
- [Function description] Searches the next peak point using the specified multimarker

  This command is used to search the point next highest to the peak point searched last, and places the marker on that point. The marker numbers that can be specified range from 1 to 10. If a marker number is omitted, the marker that is active at that time is used. An execution error occurs if the command fails to find the next peak point. If the specified marker is off, that marker is set on so that it will be displayed at the searched position.

  An execution error occurs if this command is sent with the search range limited by the :CALCulate<ch>:MARKer<screen>:SEARch command group and no waveforms exist in that range. In such a case, the marker is not displayed or remains where it was before this command is executed.
- [Parameter] None
- [Query reply] None
- [Example]

```
Mkr1$ = Space$(30)
Mkr2$ = Space$(30)

Call ibwrt (analyzer%, ":CALC:MARK:NUMB1 ON")
'------------------------- Search peak point --------------------
Call ibwrt (analyzer%, ":CALC:MARK:NUMB1:MAX")
'--------------Read the level of peak point -------------------
Call ibwrt(analyzer%, ":CALC:MARK:Y?")
Call ibrd(analyzer%, Mkr1$)

'--------------------- Search next peak point ------------------
Call ibwrt (analyzer%, ":CALC:MARK:NUMB2:MAX:NEXT")
'-------------Read the level of next peak point ----------------
Call ibwrt(analyzer%, ":CALC:MARK:Y?")
Call ibrd(analyzer%, Mkr2$)

OutputMsgs " Peak Level      = "&Mkr1$
OutputMsgs " Next Peak Level = "&Mkr2$
```

- [Relevant commands]
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>][:STATe]
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:X
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MAXimum[:PEAK]
  :CALCulate<ch>:MARKer<screen>:SEARch:X:MODE

:CALCulate<ch>:MARKer<screen>:SEARch:X:POSition
:CALCulate<ch>:MARKer<screen>:SEARch:X:WIDTh
:CALCulate<ch>:MARKer<screen>:SEARch:X:COUPling
:CALCulate<ch>:MARKer<screen>:SEARch:Y:MODE
:CALCulate<ch>:MARKer<screen>:SEARch:Y:DLINe
:CALCulate<ch>:MARKer<screen>:SEARch:Y:LUPPer
:CALCulate<ch>:MARKer<screen>:SEARch:Y:LLOWer
:CALCulate<ch>:MARKer<screen>:MAXimum:DELTa

5.14.10 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MAXimum:LEFT

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>] :MAXimum:LEFT
- [Function description] Searches the next peak point in the left direction using the specified multimarker

  This command is used to search the point next highest to the peak point searched last in the direction of lower frequencies or in the direction of backward time using the specified marker among the ten multimarkers.

  The marker numbers that can be specified range from 1 to 10. If a marker number is omitted, the marker that is active at that time is used for searching. An execution error occurs if the command fails to find the peak point. An execution error occurs also if the specified marker is off.

  An execution error occurs if this command is sent with the search range limited by the :CALCulate<ch>:MARKer<screen>:SEARch command group and no waveforms exist in that range. In such a case, the marker is not displayed or remains where it was before this command is executed.
- [Parameter] None
- [Query reply] None
- [Example]

```
Mkr1$ = Space$(30)
Mkr2$ = Space$(30)

Call ibwrt (analyzer%, ":CALC:MARK:NUMB1 ON")
'------------------------- Search peak point --------------------
Call ibwrt (analyzer%, ":CALC:MARK:NUMB1:MAX")
'--------------Read the level of peak point ------------------
Call ibwrt(analyzer%, ":CALC:MARK:Y?")
Call ibrd(analyzer%, Mkr1$)

'--------------------- Search next peak point -----------------
Call ibwrt (analyzer%, ":CALC:MARK:NUMB1:MAX:LEFT")
'--------------Read the level of peak point ------------------
```

```
Call ibwrt(analyzer%, ":CALC:MARK:Y?")
Call ibrd(analyzer%, Mkr2$)

OutputMsgs " Peak Level     = ";&Mkr1$
OutputMsgs " Left Next Peak Level = "&Mkr2$
```

- [Relevant commands]

:CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>][:STATe]
:CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive
:CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MAXimum[:PEAK]
:CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MAXimum:RIGHt
:CALCulate<ch>:MARKer<screen>:SEARch:X:MODE
:CALCulate<ch>:MARKer<screen>:SEARch:X:POSition
:CALCulate<ch>:MARKer<screen>:SEARch:X:WIDTh
:CALCulate<ch>:MARKer<screen>:SEARch:X:COUPling
:CALCulate<ch>:MARKer<screen>:SEARch:Y:MODE
:CALCulate<ch>:MARKer<screen>:SEARch:Y:DLINe
:CALCulate<ch>:MARKer<screen>:SEARch:Y:LUPPer
:CALCulate<ch>:MARKer<screen>:SEARch:Y:LLOWer
:CALCulate<ch>:MARKer<screen>:MAXimum:DELTa

5.14.11 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MAXimum:RIGHt

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]
  :MAXimum:RIGHt
- [Function description] Searches the next peak point in the right direction using the specified multimarker

  This command is used to search the point next highest to the peak point searched last in the direction of higher frequencies or in the direction of forward time using the specified marker among the ten multimarkers.

  The marker numbers that can be specified range from 1 to 10. If a marker number is omitted, the marker that is active at that time is used for searching.

  An execution error occurs if the command fails to find the peak point. An execution error occurs also if the specified marker is off.

  An execution error occurs if this command is sent with the search range limited by the :CALCulate<ch>:MARKer<screen>:SEARch command group and no waveforms exist in that range. In such a case, the marker is not displayed or remains where it was before this command is executed.
- [Parameter] None
- [Query reply] None
- [Example]

```
Mkr1$ = Space$(30)
Mkr2$ = Space$(30)

Call ibwrt (analyzer%, ":CALC:MARK:NUMB1 ON")
'------------------------- Search peak point --------------------
Call ibwrt (analyzer%, ":CALC:MARK:NUMB1:MAX")
'--------------Read the level of peak point -------------------
Call ibwrt(analyzer%, ":CALC:MARK:Y?")
Call ibrd(analyzer%, Mkr1$)

'--------------------- Search next peak point ------------------
Call ibwrt (analyzer%, ":CALC:MARK:NUMB1:MAX:RIGH")
'--------------Read the level of peak point -------------------
Call ibwrt(analyzer%, ":CALC:MARK:Y?")
Call ibrd(analyzer%, Mkr2$)

OutputMsgs " Peak Level     = "&Mkr1$
OutputMsgs " Left Next Peak Level = "&Mkr2$
```

- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>][:STATe]
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]
  :MAXimum[:PEAK]

:CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]
:MAXimum:LEFT
:CALCulate<ch>:MARKer<screen>:SEARch:X:MODE
:CALCulate<ch>:MARKer<screen>:SEARch:X:POSition
:CALCulate<ch>:MARKer<screen>:SEARch:X:WIDTh
:CALCulate<ch>:MARKer<screen>:SEARch:X:COUPling
:CALCulate<ch>:MARKer<screen>:SEARch:Y:MODE
:CALCulate<ch>:MARKer<screen>:SEARch:Y:DLINe
:CALCulate<ch>:MARKer<screen>:SEARch:Y:LUPPer
:CALCulate<ch>:MARKer<screen>:SEARch:Y:LLOWer
:CALCulate<ch>:MARKer<screen>:MAXimum:DELTa

5.14.12 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MINimum[:PEAK]

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MINimum[:PEAK]
- [Function description] Searches the minimum peak using the specified multimarker

  This command is used to search the minimum peak point of the waveform at the time it is executed, using the specified marker among the ten multimarkers.

  If a marker number is omitted, the minimum peak point is searched using the marker that is active at that time.

  An execution error occurs if the specified marker is off.

  An execution error occurs if this command is sent with the search range limited by the :CALCulate<ch>:MARKer<screen>:SEARch command group and no waveforms exist in that range.In such a case, the marker is not displayed or remains where it was before this command is executed.
- [Parameter] None
- [Query reply] None
- [Example]

```
'-------------- Activate marker No.1 and 2 -------------------
Call ibwrt (analyzer%, ":CALC:MARK:NUMB1 ON")
'------------------------- Search peak point --------------------
Call ibwrt (analyzer%, ":CALC:MARK:NUMB1:MIN")

'-----------Read the level of minimum peak point -----------
Call ibwrt(analyzer%, ":CALC:MARK:Y?")
Call ibrd(analyzer%, Mkr1$)
OutputMsgs " Min. Peak Level = "&Mkr1$
```

- [Relevant commands]
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>][:STATe]
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive
  :CALCulate<ch>:MARKer<screen>:SEARch:X:MODE
  :CALCulate<ch>:MARKer<screen>:SEARch:X:POSition
  :CALCulate<ch>:MARKer<screen>:SEARch:X:WIDTh
  :CALCulate<ch>:MARKer<screen>:SEARch:X:COUPling
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:MODE
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:DLINe
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LUPPer
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LLOWer
  :CALCulate<ch>:MARKer<screen>:MAXimum:DELTa

5.14.13 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MINimum:NEXT

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MINimum:NEXT
- [Function description] Searches the next minimum peak point using the specified multimarker

  This command is used to search the point next lowest to the peak point searched last using the specified marker among the ten multimarkers.

  The marker numbers that can be specified range from 1 to 10. If a marker number is omitted, the marker that is active at that time is used for searching. An execution error occurs if the command fails to find the peak point. An execution error occurs also if the specified marker is off.

  An execution error occurs if this command is sent with the search range limited by the :CALCulate<ch>:MARKer<screen>:SEARch command group and no waveforms exist in that range.In such a case, the marker is not displayed or remains where it was before this command is executed.
- [Parameter] None
- [Query reply] None
- [Example]

```
Mkr1$ = Space$(30)
Mkr2$ = Space$(30)

Call ibwrt (analyzer%, ":CALC:MARK:NUMB1 ON")
'------------------------ Search peak point --------------------
Call ibwrt (analyzer%, ":CALC:MARK:NUMB1:MIN")
'-------------Read the level of min. peak point -------------
Call ibwrt(analyzer%, ":CALC:MARK:Y?")
Call ibrd(analyzer%, Mkr1$)

'------------------- Search next min. peak point -------------
Call ibwrt (analyzer%, ":CALC:MARK:NUMB1:MIN:NEXT")
'-----------Read the level of next min. peak point -----------
Call ibwrt(analyzer%, ":CALC:MARK:Y?")
Call ibrd(analyzer%, Mkr2$)

OutputMsgs " Min. Peak Level      = "&Mkr1$
OutputMsgs " Next Min. Peak Level = "&Mkr2$
```

- [Relevant commands]
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>][:STATe]
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MINimum[:PEAK]
  :CALCulate<ch>:MARKer<screen>:SEARch:X:MODE
  :CALCulate<ch>:MARKer<screen>:SEARch:X:POSition

:CALCulate<ch>:MARKer<screen>:SEARch:X:WIDTh
:CALCulate<ch>:MARKer<screen>:SEARch:X:COUPling
:CALCulate<ch>:MARKer<screen>:SEARch:Y:MODE
:CALCulate<ch>:MARKer<screen>:SEARch:Y:DLINe
:CALCulate<ch>:MARKer<screen>:SEARch:Y:LUPPer
:CALCulate<ch>:MARKer<screen>:SEARch:Y:LLOWer
:CALCulate<ch>:MARKer<screen>:MAXimum:DELTa

5.14.14 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:TRACe

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:TRACe < int >
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:TRACe?
- [Function description] Moves the specified marker to the specified trace

  This command is used to move the specified marker to the specified trace. The marker numbers that can be specified range from 1 to 10, and the trace numbers that can be specified range from 1 to 4.

  If a marker number is omitted, the marker that is active at that time is moved.

  An execution error occurs if the trace of the specified trace number is blank.

  If any marker that has already been specified is off, that marker is set on so that it will be displayed on the specified trace.
- [Parameter] < int > = trace number
  Setting range: 1 - 4
- [Query reply] NR1 (integer value: trace number)
- [Example]

```
Mkr1$ = space(30)

Call ibwrt (analyzer%, ":DISP:TRAC:NUMB1:MODE MAXH")
Call ibwrt (analyzer%, ":DISP:TRAC:NUMB2:MODE AVER")
'----- Put marker on the tarace 1 and set delt. marker -------
Call ibwrt (analyzer%, ":CALC:MARK:NUMB1:TRAC 1")
Call ibwrt (analyzer%, ":CALC:DELT ON")
'-----------Move normal marker to the trance no.2-----------
Call ibwrt(analyzer%, ":CALC:MARK:NUMB1:TRAC 2")

'---- Get the level difference between trace no.1 and 2 -----
Call ibwrt(analyzer%, ":CALC:MARK:Y?")
Call ibrd(analyzer%, Mkr1$)

OutputMsgs " Level difference = "&Mkr1$
```

- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>][:STATe]
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive
  :DISPlay<ch>[:WINDow<screen>]:TRACe[:NUMBer{1|2|3|4}]
  :MODE

5.14.15 :CALCulate<ch>:MARKer<screen>:RESet

- [Command syntax] :CALCulate<ch>:MARKer<screen>:RESet
- [Function description] Sets all the markers except marker No. 1 off

  This command is used to set all the markers except marker No. 1 off.
  Marker No.1 is displayed at the center of the horizontal axis.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:RES")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>][:STATe]

5.14.16 :CALCulate<ch>:MARKer<screen>:LIST[:STATe]

- [Command syntax] :CALCulate<ch>:MARKer<screen>:LIST[:STATe] < bool >
  :CALCulate<ch>:MARKer<screen>:LIST[:STATe]?
- [Function description] Displays the marker list of the markers displayed

  This command is used to display a list of the levels, frequencies, and time values of all the markers on display.
  The level and frequency values in the list are displayed in relative values from the delta marker if the delta marker function is on.
- [Parameter] < bool > = {OFF | ON }
  OFF: Sets the display of the marker list off
  ON: Sets the display of the marker list on
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:LIST ON")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>][:STATe]
  :CALCulate<ch>:DELTamarker<screen>[:STATe]
  :CALCulate<ch>:DELTamarker<screen>:FIXed[:STATe]
  :CALCulate<ch>:DELTamarker<screen>:FIXed:MAXimum[:PEAK]
  :CALCulate<ch>:MARKer<screen>:MAXimum:DELTa

5.14.17 :CALCulate<ch>:MARKer<screen>:MAXimum:LIST

- [Command syntax] :CALCulate<ch>:MARKer<screen>:MAXimum:LIST < type >
- [Function description] Searches peak points and displays a marker list

  This command is used to search peak points, to display markers in the ascending order of frequencies or in the descending order of levels, and also to display a marker list. A maximum of 10 points can be searched.
- [Parameter] < type> = {FREQuency | LEVel }

  FREQuency: Searches and displays peak points in the order of frequencies

  LEVel: Searches and displays peak points in the order of levels
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:MAX:LIST FREQ")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>][:STATe]

  :CALCulate<ch>:DELTamarker<screen>[:STATe]

  :CALCulate<ch>:DELTamarker<screen>:FIXed[:STATe]

  :CALCulate<ch>:DELTamarker<screen>:FIXed:MAXimum[:PEAK]

  :CALCulate<ch>:MARKer<screen>:MAXimum:DELTa

5.14.18 :CALCulate<ch>:MARKer<screen>:MAXimum:CONTinuous

- [Command syntax] :CALCulate<ch>:MARKer<screen>:MAXimum:CONTinuous < bool >
- [Function description] Sets ON or OFF the continuous peak point search mode

  This command is used to set ON or OFF the continuous peak search function that searches the peak point for each sweep and places the marker on that point.

  When this function is set on, the peak points are searched on the spectrum waveform after each sweep. The marker that is active when this command is sent is used for continuous peak point search.
- [Parameter] < bool > = { OFF | ON }

  OFF: Sets the continuous peak search function off

  ON: Sets the continuous peak search function on
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:NUMB1 ON")

  Call ibwrt (analyzer%, ":CALC:MARK:MAX:CONT ON")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>][:STATe]

5.14.19 :CALCulate<ch>:MARKer<screen>:MAXimum:DELTa

- [Command syntax] :CALCulate<ch>:MARKer<screen>:MAXimum:DELTa < real >
- [Function description] Specifies a deviation for peak point judgment at the time of peak point search

  This command is used to specify a deviation for peak point judgment when each peak search function is executed. The deviation provides a criterion for the judgment that a peak point is detected by comparison on the vertical axis (level).

  The specified displacement unit is always dB, independent of the vertical axis scales and UNIT.
- [Parameter] < real > = vertical deviation in dB

  Setting range: 0.1-100.0 dB (resolution: 0.1)
- [Query reply] NR3 (real number value in dB)
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:MAX:DELT 0.7")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MAXimum :LEFT

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MAXimum :RIGHt

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MINimum :NEXT

  :CALCulate<ch>:MARKer<screen>:MAXimum:LIST

5.14.20 :CALCulate<ch>:MARKer<screen>:STEP

- [Command syntax] :CALCulate<ch>:MARKer<screen>:STEP < real >
  :CALCulate<ch>:MARKer<screen>:STEP?
- [Function description] Sets a marker step size

  This command is used to set a marker step size.

  This command sends frequency data when frequency span is used or time data when zero span is used at the time this command is sent. When this command is sent, the marker step size mode becomes off automatically so that the marker will move by the specified step size.
- [Parameter] < real > = step frequency size (GHz/MHz/kHz/Hz)
  or
  step time size (s/ms/μs/ns)
- [Query reply] NR3 (real number value)
  Frequency span in Hz
  Zero span in s
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:STEP 10MHZ")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>:STEP:AUTO

5.14.21 :CALCulate<ch>:MARKer<screen>:STEP:AUTO

- [Command syntax] :CALCulate<ch>:MARKer<screen>:STEP:AUTO < bool >
  :CALCulate<ch>:MARKer<screen>:STEP:AUTO?
- [Function description] Sets a marker step size mode

  This command is used to select a marker step size setting mode. In AUTO mode, data 1/10 the set frequency span or time is used as the marker step size. In the MANUAL mode, any data set by the command :CALCulate<ch>:MARKer<screen>:STEP is used.
- [Parameter] < bool > = { OFF | ON }
  OFF: MANUAL mode
  ON: AUTO mode
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:STEP:AUTO OFF")
  Call ibwrt (analyzer%, ":CALC:MARK:STEP 5MHZ")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>:STEP

5.14.22 :CALCulate<ch>:MARKer<screen>:SEARch:X:MODE

- [Command syntax] :CALCulate<ch>:MARKer<screen>:SEARch:X:MODE < type >
  :CALCulate<ch>:MARKer<screen>:SEARch:X:MODE?
- [Function description] Sets a peak search range specification mode on the horizontal axis

  A search range (frequency or time) can be set for peak search by a marker. When a range is set, peak search can be done in the specified waveform range.

  Any of the three options of within the specified range, outside the specified range, or the entire screen can be selected.
- [Parameter] < type > = { ALL | INNer | OUTer }

  ALL: The area of the entire screen on the horizontal axis is searched.

  INNer: The area within the specified frequency or time width is searched

  OUTer: The area outside the specified frequency or time width is searched
- [Query reply] { ALL | INN | OUT }
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:SEAR:X:MODE INN")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>:SEARch:X:POSition
  :CALCulate<ch>:MARKer<screen>:SEARch:X:WIDTh
  :CALCulate<ch>:MARKer<screen>:SEARch:X:COUPling
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:MODE
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:DLINe
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LUPPer
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LLOWer

5.14.23 :CALCulate<ch>:MARKer<screen>:SEARch:X:POSition

- [Command syntax] :CALCulate<ch>:MARKer<screen>:SEARch:X:POSition < real >
  :CALCulate<ch>:MARKer<screen>:SEARch:X:POSition?
- [Function description] Specifies the reference position of the peak search range on the horizontal axis

  This command is used to specify the reference position of the peak search range in terms of frequency or time. The parameter to send depends on whether the frequency span or the zero span is set for the span state of this command.

  If frequency span is used, specify the center position of the peak search range in terms of absolute frequency.

  If zero span is used, specify the span start time for the peak search range to be specified.

  Specify the reference position by this command and specify the peak search range by the command

  :CALCulate<ch>:MARKer<screen>:SEARch:X:WIDTh,

  which determines the width of the peak search.
- [Parameter] < real > = Frequency (GHz/MHz/kHz/Hz) or
  < real > = Time (s/ms/μs/ns)
- [Query reply] NR3 (real number value)
  Frequency span in Hz
  Zero span in s
- [Example]

```
'----------------- For freq. span mode ------------------------------------
Call ibwrt (analyzer%, ":CALC:MARK:SEAR:X:POS 1.75GHZ")

'----------------- For zero span mode ------------------------------------
Call ibwrt (analyzer%, ":CALC:MARK:SEAR:X:POS 105MS")
```

- [Relevant commands] :CALCulate<ch>:MARKer<screen>:SEARch:X:MODE
  :CALCulate<ch>:MARKer<screen>:SEARch:X:WIDTh
  :CALCulate<ch>:MARKer<screen>:SEARch:X:COUPling
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:MODE
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:DLINe
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LUPPer
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LLOWer

5.14.24 :CALCulate<ch>:MARKer<screen>:SEARch:X:WIDTh

- [Command syntax] :CALCulate<ch>:MARKer<screen>:SEARch:X:WIDTh < real >
  :CALCulate<ch>:MARKer<screen>:SEARch:X:WIDTh?
- [Function description] Specifies a search width from the reference position of the peak search range on the horizontal axis

  This command is used to specify a search width from the reference position of the peak search range in terms of frequency or time. The parameter to send depends on whether the frequency span or the zero span is set for the span state of this command.

  If frequency span is used, specify the search frequency width centered on the reference position.
  If zero span is used, specify the search time width that starts at the reference position.
  Set the reference position by the
  :CALCulate<ch>:MARKer<screen>:SEARch:X:POSition command
- [Parameter] < real > = Frequency (GHz/MHz/kHz/Hz) or
  < real > = Time (s/ms/μs/ns)
- [Query reply] NR3 (real number value)
  Frequency span in Hz
  Zero span in s
- [Example]

```
'-------------------------- For freq. span mode ---------------------------
Call ibwrt (analyzer%, ":CALC:MARK:SEAR:X:POS 1.75GHZ")
Call ibwrt (analyzer%, ":CALC:MARK:SEAR:X:WIDT 200MHZ")

'-------------------------- For zero span mode ----------------------------
Call ibwrt (analyzer%, ":CALC:MARK:SEAR:X:POS 105MS")
Call ibwrt (analyzer%, ":CALC:MARK:SEAR:X:WIDT 150MS")
```

- [Relevant commands] :CALCulate<ch>:MARKer<screen>:SEARch:X:MODE
  :CALCulate<ch>:MARKer<screen>:SEARch:X:POSition
  :CALCulate<ch>:MARKer<screen>:SEARch:X:COUPling
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:MODE
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:DLINe
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LUPPer
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LLOWer

5.14.25 :CALCulate<ch>:MARKer<screen>:SEARch:X:COUPling

- [Command syntax] :CALCulate<ch>:MARKer<screen>:SEARch:X:COUPling < bool >
  :CALCulate<ch>:MARKer<screen>:SEARch:X:COUPling?
- [Function description] Sets a move mode of the peak search range on the horizontal axis

  This instrument has a function to limit the peak search range by the marker.

  The search range is set by this search range function in two modes. In one mode the range follows the frequency set after the range is set, and in the other mode the range is fixed on the screen regardless of the set value. This command is used to select a mode.

  In the mode where the set frequency is followed, the search range does not change by the frequency or time once set, even if the center frequency or span frequency is changed after the range is set. In the other mode, where the range does not change on the screen, the range maintains its position on the screen and the screen point width that are determined at the time of range setting.The position and the width of the range do not change, even if the frequency or time changes.
- [Parameter] < bool > = { OFF | ON }
  OFF: Fixes the search range on the screen
  ON: Fixes the search range to the set frequency or time
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:SEAR:X:COUP ON")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>:SEARch:X:MODE
  :CALCulate<ch>:MARKer<screen>:SEARch:X:POSition
  :CALCulate<ch>:MARKer<screen>:SEARch:X:WIDTh
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:MODE
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:DLINe
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LUPPer
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LLOWer

5.14.26 :CALCulate<ch>:MARKer<screen>:SEARch:Y:MODE

- [Command syntax] :CALCulate<ch>:MARKer<screen>:SEARch:Y:MODE < type >
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:MODE?
- [Function description] Sets a peak search range specification mode on the vertical axis

  A level range on the vertical axis can be set in addition to the peak search target range for peak search by a marker. When this function is used, peak search can be done in the specified level range.

  One of the following three ranges can be specified as the search range: the whole vertical axis level range on the screen, the range in relation to the display line as a reference, and the range in relation to the limit line as a reference.
- [Parameter] < type > = { ALL | DLINe | LLINe }
  - ALL: The search range is the entire level displayed on the screen on the vertical axis.
  - DLINe: The search range is defined with the Display Line used as the reference.
  - LLINe: The search range is defined with the Limit Line used as the reference.
- [Query reply] { ALL | DLIN | LLIN}
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:SEAR:Y:MODE DLIN")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>:SEARch:X:MODE
  :CALCulate<ch>:MARKer<screen>:SEARch:X:POSition
  :CALCulate<ch>:MARKer<screen>:SEARch:X:WIDTh
  :CALCulate<ch>:MARKer<screen>:SEARch:X:COUPling
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:DLINe
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LUPPer
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LLOWer

5.14.27 :CALCulate<ch>:MARKer<screen>:SEARch:Y:DLINe

- [Command syntax] :CALCulate<ch>:MARKer<screen>:SEARch:Y:DLINe < type >
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:DLINe?
- [Function description] Specifies the peak search range with Display Line used as the reference

  This command is used to specify whether the signals above the Display Line on the vertical axis level as the reference are searched or those below the Display Line are searched. The setting of this command is effective if the Display Line has been selected as the reference level by the :CALCulate<ch>:MARKer<screen>:SEARch:Y:MODE command.

  The level on the Display Line is searched.
- [Parameter] < type > = { ABOVe | BELow }
  ABOVe: Searches the signals above the Display Line.
  BELow: Searches the signals below the Display Line.
- [Query reply] { ABOV | BEL }
- [Example] '-------- Set search mode to the reference of display line -----------
  Call ibwrt (analyzer%, ":CALC:MARK:SEAR:Y:MODE DLIN")
  Call ibwrt (analyzer%, ":CALC:MARK:SEAR:Y:DLIN ABOV")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>:SEARch:X:MODE
  :CALCulate<ch>:MARKer<screen>:SEARch:X:WIDTh
  :CALCulate<ch>:MARKer<screen>:SEARch:X:COUPling
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:MODE
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LUPPer
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LLOWer

5.14.28 :CALCulate<ch>:MARKer<screen>:SEARch:Y:LUPPer

- [Command syntax] :CALCulate<ch>:MARKer<screen>:SEARch:Y:LUPPer < type >
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LUPPer?
- [Function description] Specifies the peak search range with Limit Line1 used as the reference

  This command is used to specify whether the signals above the Limit Line1 on the vertical axis level as the reference are searched or those below the Limit Line are searched.

  The setting of this command is effective if the Limit Line has been selected as the reference level by the command
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:MODE.

  The level on the Limit Line is searched.
- [Parameter] < type > = { ABOVe | BELow }
  ABOVe: Searches the signals above Limit Line1.
  BELow: Searches the signals below Limit Line1.
- [Query reply] { ABOV | BEL }
- [Example]

```
'-------- Set search mode to the reference of limit line -----------
Call ibwrt (analyzer%, ":CALC:MARK:SEAR:Y:MODE LLIN")
Call ibwrt (analyzer%, ":CALC:MARK:SEAR:Y:LUPP ABOV")
Call ibwrt (analyzer%, ":CALC:MARK:SEAR:Y:LLOW BEL")
```

- [Relevant commands] :CALCulate<ch>:MARKer<screen>:SEARch:X:MODE
  :CALCulate<ch>:MARKer<screen>:SEARch:X:POSition
  :CALCulate<ch>:MARKer<screen>:SEARch:X:COUPling
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:MODE
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:DLINe
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LLOWer

5.14.29 :CALCulate<ch>:MARKer<screen>:SEARch:Y:LLOWer

- [Command syntax]
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LLOWer < type >
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LLOWer?
- [Function description]
  Specifies the peak search target range with Limit Line2 used as the reference

  This command is used to specify whether the signals above the Limit Line2 on the vertical axis level as the reference are searched or those below the Limit Line are searched.

  The setting of this command is effective if the Limit Line has been selected as the reference level by the :CALCulate<ch>:MARKer<screen>:SEARch:Y:MODE command.

  The level on the Limit Line is searched.
- [Parameter]
  < type > = { ABOVe | BELow }
  ABOVe: Searches the signals above Limit Line2.
  BELow: Searches the signals below Limit Line2.
- [Query reply]
  { ABOV | BEL }
- [Example]
```
'-------- Set search mode to the reference of limit line -----------
Call ibwrt (analyzer%, ":CALC:MARK:SEAR:Y:MODE LLIN")
Call ibwrt (analyzer%, ":CALC:MARK:SEAR:Y:LUPP ABOV")
Call ibwrt (analyzer%, ":CALC:MARK:SEAR:Y:LLOW BEL")
```
- [Relevant commands]
  :CALCulate<ch>:MARKer<screen>:SEARch:X:MODE
  :CALCulate<ch>:MARKer<screen>:SEARch:X:POSition
  :CALCulate<ch>:MARKer<screen>:SEARch:X:COUPling
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:MODE
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:DLINe
  :CALCulate<ch>:MARKer<screen>:SEARch:Y:LUPPer

5.14.30 :CALCulate<ch>:MARKer<screen>:MINNer<area>

- [Command syntax] :CALCulate<ch>:MARKer<screen>:MINNer<area> < bool >
  :CALCulate<ch>:MARKer<screen>:MINNer<area>?
- [Function description] Sets ON or OFF the display of the marker frame for the Multi Inner Peak Search function

  This command is used to set ON or OFF the display of the marker frame used in the Multi Inner Peak Search function.

  When the Multi Inner Peak Search function execution command :CALCulate<ch>:MARKer<screen>:MINNer<area>:MAXimum:PEAK is executed when the marker frame is displayed, among ten markers, the markers associated with the numbers of the specified marker frames perform peak search within the frames. This allows the peaks in a maximum of ten separate areas to be determined at one time.

  The vertical axis (Y axis) and the horizontal axis (X axis) can be set within each marker frame. The marker is displayed on the peak point in the frame if a waveform exists in the frame. The marker is not displayed if no waveform exits in the frame.
- [Parameter] < bool > = { OFF | ON }
  OFF: Sets the display of the marker frame off
  ON: Sets the display of the marker frame on
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:MINN1 ON")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>:MINNer<area>:MAXimum:PEAK
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:X:POSition
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:X:WIDTh
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y:UPPer
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y:LOWer

5.14.31 :CALCulate<ch>:MARKer<screen>:MINNer<area>:MAXimum:PEAK

- [Command syntax] :CALCulate<ch>:MARKer<screen>:MINNer<area>:MAXimum :PEAK
- [Function description] Executes peak search in all marker frames by the Multi Inner Peak Search function

  This command is used to search the peak value in all the set marker frames at one time.

  When this command is executed, among ten markers, the markers associated with the numbers of the specified marker frames perform peak search within the frames. This allows the peaks in a maximum of ten separate areas to be determined at one time.

  The vertical axis (Y axis) and the horizontal axis (X axis) can be set within each marker frame. The marker is displayed on the peak point in the frame if a waveform exists in the frame. The marker is not displayed if no waveform exits in the frame. If any number from 1 to 10 is set for the marker frame number in the command header, peak search operation is performed in all marker frames.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:MINN2:MAX:PEAK")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>:MINNer<area>:MAXimum:LIST?

  :CALCulate<ch>:MARKer<screen>:MINNer<area>

  :CALCulate<ch>:MARKer<screen>:MINNer<area>:X:POSition

  :CALCulate<ch>:MARKer<screen>:MINNer<area>:X:WIDTh

  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y

  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y:UPPer

  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y:LOWer

5.14.32 :CALCulate<ch>:MARKer<screen>:MINNer<area>:MAXimum:LIST?

- [Command syntax] :CALCulate<ch>:MARKer<screen>:MINNer<area>:MAXimum:LIST?
- [Function description] Reads the peak value in all marker frames by the Multi Inner Peak Search function

  This command is used to read the peak value searched for in the set marker frames. Data for the marker frames is output when this command is executed.
- [Parameter] None
- [Query reply] [NR1, NR3, NR3][, NR1, NR3, NR3][, NR1, NR3, NR3][, ...]

  Output order:
  [Marker frame number (1): 1,
  Frequency (1): Frequency in Hz of the peak value in marker frame 1,
  Level (1): The maximum value in dBm in marker frame 1]
  [, Marker frame number (2): 2,
  Frequency (2): Frequency in Hz of the peak value in marker frame 2,
  Level (2): The maximum value in dBm in marker frame 2]
  [,...]
  [, Marker frame number (10): 10,
  Frequency (10): Frequency in Hz of the peak value in marker frame 10,
  Level (10): The maximum value in dBm in marker frame 10]

  When the zero span is set, the time in unit of "s" is read instead of the frequency in Hz.
- [Example] MulInnMkr$ = Space$(1000)
  Call ibwrt (analyzer%, ":CALC:MARK:MINN2:MAX:LIST?")
  Call ibrd (analyzer%, MulInnMkr$)
- [Relevant commands] :CALCulate<ch>:MARKer<screen>:MINNer<area>:MAXimum:PEAK
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:X:POSition
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:X:WIDTh
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y:UPPer
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y:LOWer
  :UNIT<ch>:POWer<screen>

5.14.33 :CALCulate<ch>:MARKer<screen>:MINNer<area>:X:POSition

- [Command syntax] :CALCulate<ch>:MARKer<screen>:MINNer<area>:X:POSition <real>
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:X:POSition?
- [Function description] Specifies the position of the horizontal axis in the marker frame by the Multi Inner Peak Search function

  This command is used to specify the reference position for marker frame display when marker frame display is on. When the frequency span is set, the specified position is set by using the frequency and is used as the center position in the marker frame.

  When the zero span is set, the specified position is set by using the time and is used as the value at the left end of the frame.
- [Parameter] < real > = Frequency (GHz/MHz/kHz/Hz) or
  < real > = Time (s/ms/μs/ns)
- [Query reply] NR3 (real number value)
  Frequency span in Hz
  Zero span in s
- [Example]

```
'-------------------------- For freq. span mode --------------------------
Call ibwrt (analyzer%, ":CALC:MARK:MINN1:X:POS 1.75GHZ")
Call ibwrt (analyzer%, ":CALC:MARK:MINN1:X:WIDT 200MHZ")

'-------------------------- For zero span mode ---------------------------
Call ibwrt (analyzer%, ":CALC:MARK:MINN1:X:POS 105MS")
Call ibwrt (analyzer%, ":CALC:MARK:MINN1:X:WIDT 150MS")
```

- [Relevant commands] :CALCulate<ch>:MARKer<screen>:MINNer<area>:MAXimum :PEAK
  :CALCulate<ch>:MARKer<screen>:MINNer<area>
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:X:WIDTh
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y:UPPer
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y:LOWer

5.14.34 :CALCulate<ch>:MARKer<screen>:MINNer<area>:X:WIDTh

- [Command syntax] :CALCulate<ch>:MARKer<screen>:MINNer<area>:X:WIDTh < real >
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:X:WIDTh?
- [Function description] Specifies the width of the marker frame on the horizontal axis by the Multi Inner Peak Search function

  This command is used to specify the display width of the marker frame when marker frame display is on. When the frequency span is set, the specified width is set by using the frequency and the center position of the frame is set to the center of the width.

  When the zero span is set, the specified width is set by using the time and is used as the value at the left end of the frame.
- [Parameter] < real > = Frequency (GHz/MHz/kHz/Hz) or
  < real > = Time (s/ms/μs/ns)
- [Query reply] NR3 (real number value)
  Frequency span in Hz
  Zero span in s
- [Example]

```
'-------------------------- For freq. span mode --------------------------
Call ibwrt (analyzer%, ":CALC:MARK:MINN2:X:POS 1.75GHZ")
Call ibwrt (analyzer%, ":CALC:MARK:MINN2:X:WIDT 200MHZ")

'-------------------------- For zero span mode ----------------------------
Call ibwrt (analyzer%, ":CALC:MARK:MINN2:X:POS 105MS")
Call ibwrt (analyzer%, ":CALC:MARK:MINN2:X:WIDT 150MS")
```

- [Relevant commands] :CALCulate<ch>:MARKer<screen>:MINNer<area>:MAXimum :PEAK
  :CALCulate<ch>:MARKer<screen>:MINNer<area>
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:X:POSition
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y:UPPer
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y:LOWer

5.14.35 :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y

- [Command syntax]
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y < bool >
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y?
- [Function description]
  Sets the specified marker frame and the vertical axis search range mode by the Multi Inner Peak Search function

  This command is used to set ON or OFF the vertical axis search range limit.

  When the limit is on, search is done within the specified marker frame.

  When the limit is off, search is done over the entire scale of the vertical axis, regardless of the vertical axis of the marker frame.
- [Parameter]
  < bool > = { OFF | ON }
  OFF: Search is done over the entire scale of the vertical axis
  ON: Search is done within the specified marker frame
- [Query reply]
  { OFF | ON }
- [Example]
  Call ibwrt (analyzer%, ":CALC:MARK:MINN:Y ON")
- [Relevant commands]
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:MAXimum:PEAK
  :CALCulate<ch>:MARKer<screen>:MINNer<area>
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:X:POSition
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:X:WIDTh
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y:UPPer
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y:LOWer

5.14.36 :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y:LOWer

- [Command syntax] :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y:LOWer < real >
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y:LOWer?
- [Function description] Specifies the lower position of the vertical axis of the marker frame by the Multi Inner Peak Search function

  This command is used to specify the lower position of the vertical axis of the marker frame when marker frame display on is specified.
- [Parameter] < real > = level (in dBm)
- [Query reply] NR3 (real number value in dBm)
- [Example] '-------------------------- For freq. span mode ---------------------------
  Call ibwrt (analyzer%, ":CALC:MARK:MINN1:Y:LOW -12DBM")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>:MINNer<area>:MAXimum:PEAK
  :CALCulate<ch>:MARKer<screen>:MINNer<area>
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:X:WIDTh
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:X:POSition
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y:UPPer
  :UNIT<ch>:POWer<screen>

5.14.37 :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y:UPPer

- [Command syntax] :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y:UPPer < real >
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y:UPPer?
- [Function description] Specifies the upper position of the vertical axis of the marker frame by the Multi Inner Peak Search function

  This command is used to specify the upper position of the vertical axis of the marker frame when marker frame display on is specified.
- [Parameter] < real > = level (in dBm)
- [Query reply] NR3 (real number value in dBm)
- [Example] '-------------------------- For freq. span mode --------------------------
  Call ibwrt (analyzer%, ":CALC:MARK:MINN1:Y:UPP -12DBM")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>:MINNer<area>:MAXimum :PEAK
  :CALCulate<ch>:MARKer<screen>:MINNer<area>
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:X:POSition
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:X:WIDTh
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y
  :CALCulate<ch>:MARKer<screen>:MINNer<area>:Y:LOWer
  :UNIT<ch>:POWer<screen>

5.14.38 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:SET:CENTer

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:SET:CENTer
- [Function description] Sets the marker frequency as the center frequency

  This command is used to set the frequency position of the specified marker as the center frequency.

  If a marker number is not specified, the marker that is active when this command is sent is used.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:NUMB2:ACT")
  Call ibwrt (analyzer%, ":CALC:MARK:SET:CENT")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive

5.14.39 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:SET:RLEVel

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:SET:RLEVel
- [Function description] Sets the marker level as the reference level

  This command is used to set the level value of the specified marker as the reference level.

  If a marker number is not specified, the marker that is active when this command is sent is used.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:NUMB3:SET:RLEV")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive

5.14.40 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:SET:CENTer:STEP

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:SET:CENTer :STEP
- [Function description] Sets the marker frequency as the center frequency step size

  This command is used to set the frequency position of the specified marker as the step width for setting a center frequency.

  If a marker number is not specified, the marker that is active when this command is sent is used.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:NUMB2:ACT")

  Call ibwrt (analyzer%, ":CALC:MARK:SET:CENT:STEP")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive

5.14.41 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:SET:MARKer:STEP

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:SET:MARKer :STEP
- [Function description] Sets the marker frequency as the marker frequency step size

  This command is used to set the frequency position or time value of the specified marker as the step size used when the marker is moved.

  If a marker number is not specified, the marker that is active when this command is sent is used.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:NUMB1:SET:MARK:STEP")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive

5.14.42 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MAXimum:SET :CENTer

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MAXimum :SET:CENTer
- [Function description] Sets the marker frequency as the center frequency after peak search is performed

  This command is used to set the marker frequency as the center frequency after peak search is performed by that marker and the marker is moved to the peak position.

  If a marker number is not specified, the marker that is active when this command is sent is used.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:NUMB2:ACT")

  Call ibwrt (analyzer%, ":CALC:MARK:MAX:SET:CENT")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive

5.14.43 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MAXimum:SET :RLEVel

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:MAXimum :SET:RLEVel
- [Function description] Sets the marker level value as the reference level after peak search is performed

  This command is used to set the marker level value as the reference level after peak search is performed by that marker and the marker is moved to the peak position.

  If a marker number is not specified, the marker that is active when this command is sent is used.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:NUMB2:ACT")

  Call ibwrt (analyzer%, ":CALC:MARK:MAX:SET:RLEV")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive

5.14.44 :CALCulate<ch>:DELTamarker<screen>[:NUMBer<mkr>]:SET:CENTer

- [Command syntax] :CALCulate<ch>:DELTamarker<screen>[:NUMBer<mkr>]:SET :CENTer
- [Function description] Sets the delta marker frequency as the center frequency

  This command is used to set the frequency width of the specified delta marker as the center frequency.

  If a marker number is not specified, the marker that is active when this command is sent is used.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:DELT:NUMB3:SET:CENT")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive

5.14.45 :CALCulate<ch>:DELTamarker<screen>[:NUMBer<mkr>]:SET:SPAN

- [Command syntax] :CALCulate<ch>:DELTamarker<screen>[:NUMBer<mkr>]:SET:SPAN
- [Function description] Sets the delta marker frequency as the span frequency

  This command is used to set the frequency width of the specified delta marker as the span frequency. Sets the frequency at the center of the delta marker to the center frequency.

  If a marker number is not specified, the marker that is active when this command is sent is used.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:NUMB2:ACT")
  Call ibwrt (analyzer%, ":CALC:DELT:SET:SPAN")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive

5.14.46 :CALCulate<ch>:DELTamarker<screen>[:NUMBer<mkr>]:SET :CENTer:STEP

- [Command syntax] :CALCulate<ch>:DELTamarker<screen>[:NUMBer<mkr>]:SET :CENTer:STEP
- [Function description] Sets the delta marker frequency as the center frequency step size

  This command is used to set the frequency width of the specified delta marker as the step size of the center frequency.

  If a marker number is not specified, the marker that is active when this command is sent is used.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:DELT:NUMB3:SET:CENT:STEP")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive

5.14.47 :CALCulate<ch>:DELTamarker<screen>[:NUMBer<mkr>]:SET :MARKer:STEP

- [Command syntax] :CALCulate<ch>:DELTamarker<screen>[:NUMBer<mkr>]:SET :MARKer:STEP
- [Function description] Sets the delta marker frequency as the marker step size

  This command is used to set the frequency width or time width of the specified delta marker as the marker step size.

  If a marker number is not specified, the marker that is active when this command is sent is used.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:NUMB2:ACT")
  Call ibwrt (analyzer%, ":CALC:DELT:SET:MARK:STEP")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>] :ACTive

5.14.48 :CALCulate<ch>:DELTamarker<screen>[:STATe]

- [Command syntax] :CALCulate<ch>:DELTamarker<screen>[:STATe] < bool >
  :CALCulate<ch>:DELTamarker<screen>[:STATe]?
- [Function description] Sets ON or OFF the display of the delta marker

  This command is used to set ON or OFF the display of the delta marker of the specified channel on the specified screen.

  When delta marker on is specified, the positions of levels, frequency or time are fixed and the differences of those of the marker that was active when the delta marker is displayed are determined.
- [Parameter] < bool > = { OFF | ON }
  OFF: Sets the display of the delta marker off
  ON: Sets the display of the delta marker on
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:NUMB2:ACT")
  Call ibwrt (analyzer%, ":CALC:DELT:STAT ON")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive

5.14.49 :CALCulate<ch>:DELTamarker<screen>:FIXed[:STATe]

- [Command syntax] :CALCulate<ch>:DELTamarker<screen>:FIXed[:STATe] < bool >
  :CALCulate<ch>:DELTamarker<screen>:FIXed[:STATe]?
- [Function description] Sets ON or OFF the display of the fixed delta marker

  This command is used to set ON or OFF the display of the fixed delta marker of the specified channel on the specified screen.

  When fixed delta marker on is specified, differences of levels, frequency or time from those of the marker that was active when the delta marker is displayed are determined.

  A fixed delta marker differs from an ordinary delta marker in that the fixed delta marker can determine the frequency positions or level values to be used as reference values even when they are out of the screen, while an ordinary marker can determine difference values only when they are within the screen.
- [Parameter] < bool > = { OFF | ON }
  OFF: Sets the display of the fixed delta marker off
  ON: Sets the display of the fixed delta marker on
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":CALC:DELT:FIX:STAT ON")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:ACTive

5.14.50 :CALCulate<ch>:DELTamarker<screen>:FIXed:MAXimum[:PEAK]

- [Command syntax] :CALCulate<ch>:DELTamarker<screen>:FIXed:MAXimum[:PEAK]
- [Function description] Searches the peak and sets a fixed marker

  This command is use to search the peak and place a fixed marker on the peak point.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:DELT:FIX:MAX:PEAK")
- [Relevant commands] :CALCulate<ch>:DELTamarker<screen>[:STATe]

5.14.51 :CALCulate<ch>:DELTamarker<screen>:INVerse[:STATe]

- [Command syntax] :CALCulate<ch>:DELTamarker<screen>:INVerse[:STATe] < bool >
  :CALCulate<ch>:DELTamarker<screen>:INVerse[:STATe]?
- [Function description] Sets ON or OFF the display of the (1/delta) marker

  This command is used to set ON or OFF the display of the (1/delta) marker.

  The (1/delta) marker converts the frequency indicated by the currently positioned delta marker into time data for display in the case of frequency span. The (1/delta) marker converts the time indicated by the currently positioned delta marker into frequency data for display in the case of zero span.
- [Parameter] < bool > = { OFF | ON }
  OFF: Sets the display of the (1/delta) marker off
  ON: Sets the display of the (1/delta) marker on
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":CALC:DELT:INV:STAT ON")
- [Relevant commands] :CALCulate<ch>:DELTamarker<screen>[:STATe]

5.14.52 :CALCulate<ch>:DELTamarker<screen>:X?

- [Command syntax] :CALCulate<ch>:DELTamarker<screen>:X?
- [Function description] Reads the frequency value of the delta marker

  This command is used to read the absolute frequency value of the delta marker.
- [Parameter] None
- [Query reply] NR3 (real number value: frequency or time)
  Frequency span in Hz
  Zero span in s
- [Example] DeltMkrFreq$ = Space$(20)
  Call ibwrt (analyzer%, ":CALC:DELT:X?")
  Call ibrd (analyzer%, DeltMkrFreq$)
- [Relevant commands] :CALCulate<ch>:DELTamarker<screen>:Y?

5.14.53 :CALCulate<ch>:DELTamarker<screen>:Y?

- [Command syntax] :CALCulate<ch>:DELTamarker<screen>:Y?
- [Function description] Reads the level value of the delta marker

  This command is used to read the absolute level value of the delta marker.
- [Parameter] None
- [Query reply] NR3 (real number value of the delta marker level in dBm)
- [Example] DeltMkLvl$ = Space$(20)
  Call ibwrt (analyzer%, ":CALC:DELT:Y?")
  Call ibrd (analyzer%, DeltMkrLvl$)
- [Relevant commands] :CALCulate<ch>:DELTamarker<screen>:X?
  :UNIT<ch>:POWer<screen>

5.14.54 :CALCulate<ch>:MARKer<screen>:ROBJect

- [Command syntax] :CALCulate<ch>:MARKer<screen>:ROBJect < type >
- [Function description] Specifies a reference for the relative value display of the marker

  This command is used to specify an object that indicates a reference level when a relative marker read value is displayed.

  The following seven options are available: The reference level position is automatically the delta marker position when delta marker functions are used.
  - Delta marker
  - Anchor
  - Limit line 1 or 2
  - Display Line
  - Reference Line
  - Trace numbers 1 to 4
  - Delta marker in the screen, which is not the marker, in the 2-screen display

  An execution error occurs if the specified object is not displayed.
- [Parameter] < type > = { DELTamarker | ANCHor | LIM1 | LIM2 | DLINe | RLINe | TRA1 | TRA2 | TRA3 | TRA4 | OSCReen | NREFerence }

  DELTmarker: Delta marker reference

  ANCHor: Anchor reference

  LIM1 - 2: Limit line 1 or 2 reference

  DLINe: Display line reference

RLINe: Reference line reference
TRA1 - 4: Trace 1, 2, 3, or 4 reference
OSCReen: Delta marker reference in the screen, which is not the marker, in the 2-screen display
NREFerence: Reference is cancelled and relative value display is cleared.

- [Query reply] { DELT | ANCH | LIM1 | LIM2 | DLIN | RLIN | TRA1 | TRA2 | TRA3 | TRA4 | OSCR | NREF }
- [Example] Call ibwrt (analyzer%, ":CALC:RLIN:STAT ON")
  Call ibwrt (analyzer%, ":CALC:MARK:ROBJ RLIN")
- [Relevant commands] :CALCulate<ch>:DELTamarker<screen>[:STATe]
  :CALCulate<ch>:CURSor<screen>:ANCHor
  :CALCulate<ch>:LIMit<screen>:UPPer:STATe
  :CALCulate<ch>:LIMit<screen>:LOWer:STATe
  :CALCulate<ch>:DLINe<screen>:STATe
  :CALCulate<ch>:RLINe<screen>:STATe

5.14.55 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FCOunt[:STATe]

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FCOunt[:STATe]< bool >
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FCOunt[:STATe]?
- [Function description] Sets ON or OFF the frequency counter function

  This command is used to set ON or OFF the frequency counter function at the position of the marker with the specified number.

  If the marker is not displayed when this command sent the frequency counter on setting, the marker is displayed before the frequency counter function is set on.

  When a marker number is omitted, the marker last specified for active is used.
- [Parameter] < bool > = { OFF | ON }
  OFF: Sets the frequency counter function off
  ON: Sets the frequency counter function on
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:NUMB3:FCO ON")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FCOunt:FREQuency?

5.14.56 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FCOunt:FREQuency?

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FCOunt :FREQuency?
- [Function description] Reads the results of measurement by the frequency counter function

  This command is used to read counting results by the frequency counter function.

  The resolution of the results is 0.01 Hz.

  Counting is performed at the end of sweeping and the result is displayed.

  This means if counting values are read before sweeping, the previous counting result will be returned.

  An query error occurs if the counter function is not on for the marker of the specified number. An indefinite string <NO DATA> will be returned instead of a counting value in that case.
- [Parameter] None
- [Query reply] NR3 (real number value of counting result in Hz)
- [Example] FreqCOUNT$ = Space$(100)

  Call ibwrt (analyzer%, ":CALC:MARK:NUMB3:FCO ON")

  Call ibwrt (analyzer%, ":INIT:TS")

  Call ibwrt (analyzer%, ":CALC:MARK:NUMB3:FCO:FREQ?")

  Call ibrd(analyzer%, FreqCOUNT$)

  OutputMsgs "FREQ COUNTER VALUE = "&FreqCOUNT$
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>] :FCOunt[:STATe]

5.14.57 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:STRack[:STATe]

- [Command syntax] CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:STRack[:STATe] < bool >
  CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:STRack[:STATe]?
- [Function description] Sets ON or OFF the Signal Tracking function

  This command is used to set ON or OFF the Signal Tracking function by the marker.

  The signal tracking function tracks a signal that moves within a screen and sets it to the center frequency position.
- [Parameter] < bool > = { OFF | ON }
  OFF: Sets the Signal Tracking function off
  ON: Sets the Signal Tracking function on
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:NUMB1:STR ON")
- [Relevant commands]

5.14.58 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion:XDBDown

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion:XDBDown
- [Function description] Executes the X dB Down function

  This command is used to execute the X dB Down function.

  The X dB Down function finds a point, at which the difference between the level of the specified marker and the level at the point first exceeds the X dB Down value in the direction of lower frequencies or in the minus direction of time, on the waveform.

  The X dB Down function also finds a point, at which the difference between the level of the specified marker and the level at the point first exceeds the X dB Down value in the direction of higher frequencies or in the plus direction of time, on the waveform.

  The marker display mode is determined by the :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown:MODE command.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:NUMB1:FUNC:XDBD")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown:LEVel
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion:XDBDown:LEFT]

:CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown:RIGHt

:CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown:MODE

:CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown :CONTinuous[:STATe]

:CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown :RMARker[:STATe]

5.14.59 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown:LEFT

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown:LEFT
- [Function description] Executes the X dB Down Left function

  This command is used to execute the X dB Down Left function.

  The X dB Down Left function finds a waveform that dropped first by the difference of the X dB Down value or more in the direction of lower frequencies or in the minus direction of time and moves the marker to that position.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:FUNC:XDBD:LEFT")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown:LEVel

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown:RIGHt

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown:MODE

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown :CONTinuous[:STATe]

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown :RMARker[:STATe]

5.14.60 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion:XDBDown :RIGHt

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown:RIGHt
- [Function description] Executes the X dB Down Right function

  This command is used to execute the X dB Down Right function.

  The X dB Down Right function finds a waveform that dropped first by the difference of the X dB Down value or more in the direction of higher frequencies or in the plus direction of time and moves the marker to that position.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:FUNC:XDBD:RIGH")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown:LEVel

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown:LEFT

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown:MODE

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown :CONTinuous[:STATe]

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown :RMARker[:STATe]

5.14.61 :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown:LEVel

- [Command syntax] :CALCulate<ch>:MARKer<screen>:FUNCtion :XDBDown:LEVel < real >

  :CALCulate<ch>:MARKer<screen>:FUNCtion :XDBDown:LEVel?
- [Function description] Sets the dB Down value used in the X dB Down function.

  This command is used to set the dB Down value for judgment in the execution of the X dB Down or X dB Down Left/Right function.
- [Parameter] <real> = dB Down value (dB)
- [Query reply] NR3 (real number value of relative level in dB)
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:FUNC:XDBD:LEV 10DB")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown:LEFT

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown:RIGHt

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown :CONTinuous[:STATe]

5.14.62 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion:XDBDown :PEAK

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown:PEAK
- [Function description] Executes the X dB Down function after peak search

  This command is used to perform peak search before execution of the X dB Down function and execute that function with the searched peak point used as the reference position.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:FUNC:XDBD:PEAK")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown:LEFT

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown:RIGHt

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown:LEVel

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown:MODE

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown :CONTinuous[:STATe]

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown :RMARker[:STATe]

5.14.63 :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown:MODE

- [Command syntax] :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown:MODE <type>

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown:MODE?
- [Function description] Selects a display mode after execution of the X dB Down function

  This command is used to select a display mode in the marker display area after execution of the X dB Down function. The following three options are available:
  - Display of relative values
  - Display of absolute values of the left marker
  - Display of absolute values of the right marker
- [Parameter] < type > = { RELative | ABSLeft | ABSRight }

  RELative: display of relative values

  ABSLeft: display of absolute values of the left marker

  ABSRight: display of absolute values of the right marker
- [Query reply] { REL | ABSL | ABSR }
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:FUNC:XDBD:MODE ABSL")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown:LEFT

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown:PEAK

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown:RIGHt

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown :CONTinuous[:STATe]

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown :RMARker[:STATe]

5.14.64 :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown :CONTinuous[:STATe]

- [Command syntax] :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBdown :CONTinuous[:STATe] < bool >

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBdown :CONTinuous[:STATe]?
- [Function description] Sets ON or OFF the continuous X dB Down function

  The X dB Down function is executed ordinarily only when a command is sent, but there is a mode in which this function is executed automatically after each sweep.

  This mode is called continuous X dB Down function. This command is used to set ON or OFF this function.
- [Parameter] < bool > = { OFF | ON }

  OFF: Sets the continuous X dB Down function off.

  ON: Sets the continuous X dB Down function on.
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:FUNC:XDBD:CONT ON")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown:LEFT

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown:PEAK

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown:RIGHt

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown:MODE

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown :RMARker[:STATe]

5.14.65 :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown :RMARker[:STATe]

- [Command syntax]

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown :RMARker[:STATe] < bool >

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown :RMARker[:STATe]?

- [Function description]

  Sets ON or OFF the display of the reference marker when the X dB Down function is executed

  This command is used to specify whether or not the marker is displayed at the level reference position when the X dB Down function is executed.

  When ON is specified, a special marker is placed, in addition to the marker on the X dB Down position, on the reference signal position used when marker positions are searched for. This special marker is called a reference marker.

- [Parameter]

  < bool > = { OFF | ON }

  OFF: Sets the display of the reference marker off

  ON: Sets the display of the reference marker on

- [Query reply]

  { OFF | ON }

- [Example]

  Call ibwrt (analyzer%, ":CALC:MARK:FUNC:XDBD:RMAR :STAT ON")

- [Relevant commands]

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown:LEFT

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown:PEAK

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :XDBDown:RIGHt

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown:MODE

  :CALCulate<ch>:MARKer<screen>:FUNCtion:XDBDown :CONTinuous[:STATe]

5.14.66 :CALCulate<ch>:MARKer<screen>:FUNCtion:NOISe:BWIDth

- [Command syntax] :CALCulate<ch>:MARKer<screen>:FUNCtion:NOISe :BWIDth < real >
  :CALCulate<ch>:MARKer<screen>:FUNCtion:NOISe:BWIDth?
- [Function description] Sets a bandwidth for noise measurement in Noise/Hz measurement

  This command is used to specify the bandwidth for noise measurement made in Noise/Hz measurement (noise measurement).
- [Parameter] < real > = bandwidth for noise measurement (MHz/kHz/Hz)
- [Query reply] NR3 (real number value in Hz)
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:FUNC:NOIS:BWID 100HZ")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :NOISe:STATe
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :NOISe:MODE

5.14.67 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion:NOISe :STATe

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :NOISe:STATe < bool >
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :NOISe:STATe?
- [Function description] Sets ON or OFF the Noise/Hz function

  This command is used to set ON or OFF the Noise/Hz measurement at the active marker position. When the Noise/Hz measurement is set on, an average detector is selected internally and automatically.

  If the marker is not displayed when this command is sent, the marker is set on before measurement is conducted.
- [Parameter] < bool > = { OFF | ON }
  OFF: Sets the Noise/Hz function off
  ON: Sets the Noise/Hz function on
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:FUNC:NOIS:STAT ON")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>:FUNCtion:NOISe:BWIDth
  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :NOISe:MODE

5.14.68 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion:NOISe :MODE

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :NOISe:MODE < type >

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :NOISe:MODE?
- [Function description] Selects an operation mode for Noise/Hz function

  This command is used to select an operation mode for Noise/Hz measurement before it is started. The following three options are available:
  - dBm/Hz operation
  - dBμV/√Hz operation
  - dBc/Hz operation

  If the marker is not displayed when this command is sent, the marker is set on before measurement is conducted.
- [Parameter] < type> = { DBM | DBUV | DBC }

  DBM: Selects the dBm/Hz operation

  DBUV: Selects the dBμV/√Hz operation

  DBC: Selects the dBc/Hz operation
- [Query reply] { DBM | DBUV | DBC }
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:FUNC:NOIS:MODE DBM")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>:FUNCtion:NOISe:BWIDth

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :NOISe:STATe

5.14.69 :CALCulate<ch>:MARKer<screen>:FUNCtion:NOISe?

- [Command syntax] :CALCulate<ch>:MARKer<screen>:FUNCtion:NOISe?
- [Function description] Reads the results of measurement by the Noise/Hz function

  This command is used to read the measurement results of Noise/Hz measurement (noise power measurement).

  The unit of the measurement results differs with the operation mode specified by the command :CALCulate<ch>:MARKer<screen> [:NUMBer<mkr>]:FUNCtion:NOISe:MODE.
- [Parameter] None
- [Query reply] NR3 (real number value in dBm/Hz, dBμV/√Hz or dBc/Hz)
- [Example] Mkr$ = Space$(100)

  Call ibwrt (analyzer%, ":CALC:MARK:FUNC:NOIS?")

  Call ibrd(analyzer%, Mkr$)
- [Relevant commands] :CALCulate<ch>:MARKer<screen>:FUNCtion:NOISe:BWIDth

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :NOISe:STATe

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :NOISe:MODE

5.14.70 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion:AM?

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion:AM?
- [Function description] Reads the results of measurement by %AM measurement function

  This command is used to read the results of %AM measurement. The value to be read is an AM modulation factor in percentage, determined from the level difference between the delta marker and the ordinary marker.
- [Parameter] None
- [Query reply] None
- [Example] AM_Result$ = Space$(100)

  Call ibrd(analyzer%, InstID$)

  Call ibwrt (analyzer%, ":CALC:MARK:FUNC:AM:STAT ON")

  Call ibwrt (analyzer%, ":INIT:TS")

  Call ibwrt (analyzer%, ":CALC:MARK:FUNC:AM?")

  Call ibrd (analyzer%, AM_Result$)
- [Relevant commands] CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :AM:STATe

  :CALCulate<ch>:MARKer<screen>:MAXimum:DELTa

5.14.71 :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion:AM:STATe

- [Command syntax] :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :AM:STATe < bool >

  :CALCulate<ch>:MARKer<screen>[:NUMBer<mkr>]:FUNCtion :AM:STATe?
- [Function description] Sets ON or OFF %AM measurement

  This command is used to set ON or OFF %AM measurement.

  When this command is executed, the delta marker is placed on the maximum peak point of a waveform and the normal marker is placed on the next peak point for display of the AM modulation factor in percentage determined from the level difference. A peak point judgment parameter that is an appropriate setting of Peak Delta Y is required because a search will be used.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:MARK:FUNC:AM:STAT ON")
- [Relevant commands] :CALCulate<ch>:MARKer<screen>:MAXimum:DELTa

5.14.72 :CALCulate<ch>:DLINe<screen>

- [Command syntax] :CALCulate<ch>:DLINe<screen> < real >

  :CALCulate<ch>:DLINe<screen>?
- [Function description] Sets the display position of the display line

  This command is used to set a display position of the display line (level).
- [Parameter] < real > = display level of the display line (in dBm)
- [Query reply] NR3 (real number value in dBm)
- [Example] Call ibwrt (analyzer%, ":CALC:DLIN -25DBM")
- [Relevant commands] :CALCulate<ch>:DLINe<screen>:STATe

  :UNIT<ch>:POWer<screen>

5.14.73 :CALCulate<ch>:DLINe<screen>:STATe

- [Command syntax] :CALCulate<ch>:DLINe<screen>:STATe < bool >
  :CALCulate<ch>:DLINe<screen>:STATe?
- [Function description] Sets ON or OFF the display of the display line

  This command is used to set ON or OFF the display of the display line.
- [Parameter] < bool > = { OFF | ON }
  OFF: Sets the display of the display line off
  ON: Sets the display of the display line on
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":CALC:DLIN:STAT ON")
- [Relevant commands] :CALCulate<ch>:DLINe<screen>

5.14.74 :CALCulate<ch>:RLINe<screen>

- [Command syntax] :CALCulate<ch>:RLINe<screen> < real >
  :CALCulate<ch>:RLINe<screen>?
- [Function description] Sets the display position of the reference line

  This command is used to set the display position (level) of the reference line.
- [Parameter] < real > = display level of the reference line (in dBm)
- [Query reply] NR3 (real number value in dBm)
- [Example] Call ibwrt (analyzer%, ":CALC:RLIN -25DBM")
- [Relevant commands] :CALCulate<ch>:RLINe<screen>:STATe
  :UNIT<ch>:POWer<screen>

5.14.75 :CALCulate<ch>:RLINe<screen>:STATe

- [Command syntax] :CALCulate<ch>:RLINe<screen>:STATe < bool >
  :CALCulate<ch>:RLINe<screen>:STATe?
- [Function description] Sets ON or OFF the display of the reference line

  This command is used to set ON or OFF the display of the reference line.
- [Parameter] < bool > = { OFF | ON }
  OFF: Sets the display of the reference line off
  ON: Sets the display of the reference line on
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":CALC:RLIN:STAT ON")
- [Relevant commands] :CALCulate<ch>:RLINe<screen>

5.14.76 :CALCulate<ch>:LIMit<screen>:AUTO

- [Command syntax] :CALCulate<ch>:LIMit<screen>:AUTO
- [Function description] Automatically adjusts the level position of the limit line

  This command is used to automatically adjust the level position of the limit line.

  This command redraws the limit line using a peak point of the waveform and switches the reference of Y data to User Define automatically.

  This command is effective only when relative mode is set by the command :CALCulate<ch>:LIMit<screen>:CONTrol:Y:MODE. Traces are searched in the search for a peak point as the reference in the ascending order of the trace number if the traces are displayed.

  For example, trace No. 1 is searched for if it is displayed, and trace No. 2 is searched for if trace No. 1 is blank and trace No. 2 is in the Write state.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:LIM:AUTO")
- [Relevant commands] :CALCulate<ch>:LIMit<screen>:CONTrol:X:DOMain
  :CALCulate<ch>:LIMit<screen>:CONTrol:Y:MODE
  :CALCulate<ch>:LIMit<screen>:CONTrol:Y:OFFSet

5.14.77 :CALCulate<ch>:LIMit<screen>:CONTrol:X:DOMain

- [Command syntax] :CALCulate<ch>:LIMit<screen>:CONTrol:X:DOMain < type >
  :CALCulate<ch>:LIMit<screen>:CONTrol:X:DOMain?
- [Function description] Selects the domain of the limit line to be used

  This instrument manages limit line data, dividing it into two domains, or two systems for the frequency domain and two systems for the time domain. Therefore, a domain must be specified before editing of the limit line data is begun. This command is used to specify which domain to use. Since the other sorts of limit line control data listed below are also divided into the frequency domain and the time domain, this command must be sent before these sorts of data are handled.

  X Data Mode
  X Data Reference
  X Data User Define
  X Data Offset
  Y Data Mode
  Y Data Reference
  Y Data User Define
  Y Data Offset
  Limit Line Copy Table 1 → 2
  Limit Line Copy Table 2 → 1
- [Parameter] < type > = { FREQuency | TIME }
  FREQuency: Specify the frequency domain to use thereafter
  TIME: Specify the time domain to use thereafter
- [Query reply] { FREQ | TIME }
- [Example]

```
'----------- Set object domain to frequency -----------------------------
Call ibwrt (analyzer%, ":CALC:LIM:CONT:X:DOM FREQ")
'----------- Set each limit line data ---------------------------------------
Call ibwrt (analyzer%, ":CALC:LIM:LOW:DATA -100MHZ,-45DB")
Call ibwrt (analyzer%, ":CALC:LIM:LOW:DATA -50MHZ,-45DB")
Call ibwrt (analyzer%, ":CALC:LIM:LOW:DATA -10MHZ,-0DB")
Call ibwrt (analyzer%, ":CALC:LIM:LOW:DATA 10MHZ,45DB")
```

5.14.77 :CALCulate<ch>:LIMit<screen>:CONTrol:X:DOMain

- [Relevant commands]
  :CALCulate<ch>:LIMit<screen>:CONTrol:X:MODE
  :CALCulate<ch>:LIMit<screen>:CONTrol:X:REFerence
  :CALCulate<ch>:LIMit<screen>:CONTrol:X:USER
  :CALCulate<ch>:LIMit<screen>:CONTrol:X:OFFSet
  :CALCulate<ch>:LIMit<screen>:CONTrol:Y:MODE
  :CALCulate<ch>:LIMit<screen>:CONTrol:Y:REFerence
  :CALCulate<ch>:LIMit<screen>:CONTrol:Y:USER
  :CALCulate<ch>:LIMit<screen>:CONTrol:Y:OFFSet
  :CALCulate<ch>:LIMit<screen>:LOWer:COPY
  :CALCulate<ch>:LIMit<screen>:LOWer:DATA
  :CALCulate<ch>:LIMit<screen>:LOWer:DELete
  :CALCulate<ch>:LIMit<screen>:LOWer:STATe
  :CALCulate<ch>:LIMit<screen>:UPPer:COPY
  :CALCulate<ch>:LIMit<screen>:UPPer:DATA
  :CALCulate<ch>:LIMit<screen>:UPPer:DELete
  :CALCulate<ch>:LIMit<screen>:UPPer:STATe

5.14.78 :CALCulate<ch>:LIMit<screen>:CONTrol:X:MODE

- [Command syntax] :CALCulate<ch>:LIMit<screen>:CONTrol:X:MODE < type >
  :CALCulate<ch>:LIMit<screen>:CONTrol:X:MODE?
- [Function description] Selects horizontal axis data attributes of the limit line to be used

  This command is used to select horizontal data axis attributes of the limit line to be used in absolute or relative value. If the absolute value is selected, all limit line data is handled as absolute frequencies or absolute time.

  If the relative value is selected, the reference position selected by the command :CALCulate<ch>:LIMit<screen>:CONTrol:X:REFerence is handled as the center.
- [Parameter] < type > = { ABSolute | RELative }

  ABSolute: Handles the input data as absolute value (frequency or time) data.

  RELative: Handles the input data as relative value (frequency or time) data.
- [Query reply] { ABS | REL }
- [Example] Call ibwrt (analyzer%, ":CALC:LIM:CONT:X:MODE REL")
- [Relevant commands] :CALCulate<ch>:LIMit<screen>:LOWer:DATA
  :CALCulate<ch>:LIMit<screen>:UPPer:DATA
  :CALCulate<ch>:LIMit<screen>:CONTrol:X:REFerence

5.14.79 :CALCulate<ch>:LIMit<screen>:CONTrol:X:REFerence

- [Command syntax] :CALCulate<ch>:LIMit<screen>:CONTrol:X:REFerence < type >
  :CALCulate<ch>:LIMit<screen>:CONTrol:X:REFerence?
- [Function description] Specifies the reference position when the horizontal axis relative value attributes of the limit line to be used are selected

  This command is used to select the reference position when the attribute of the horizontal axis limit line data to be used is set to the relative value mode. The reference position can be set at the center of the screen, the left part of the screen, or anywhere specified by the values defined by the user.

  When user-defined values are used, the values specified are set for the reference position by the
  command:CALCulate<ch>:LIMit<screen>:CONTrol:X:USER
- [Parameter] < type > = { CENTer | LEFT | USER }
  CENTer: Sets the center of the screen as the reference position
  LEFT: Sets the leftmost part of the screen as the reference position
  USER: Sets the user-specified position as the reference position
- [Query reply] { CENT | LEFT | USER }
- [Example]
  ```
  '--------------- Select the reference type to USER defined mode -------
  Call ibwrt (analyzer%, ":CALC:LIM:CONT:X:REF USER)
  '--------------- Set the reference position to 850MHz -----------------
  Call ibwrt (analyzer%, ":CALC:LIM:CONT:X:USER 850MHZ")
  ```
- [Relevant commands] :CALCulate<ch>:LIMit<screen>:LOWer:DATA
  :CALCulate<ch>:LIMit<screen>:UPPer:DATA
  :CALCulate<ch>:LIMit<screen>:CONTrol:X:MODE
  :CALCulate<ch>:LIMit<screen>:CONTrol:X:USER

5.14.80 :CALCulate<ch>:LIMit<screen>:CONTrol:X:USER

- [Command syntax] :CALCulate<ch>:LIMit<screen>:CONTrol:X:USER < real >
  :CALCulate<ch>:LIMit<screen>:CONTrol:X:USER?
- [Function description] Sets the user-defined reference position when the horizontal axis relative value attributes of the limit line to be used are selected

  This command is used to set the reference position when the attribute of the horizontal axis limit line data to be used is set to the relative value mode and the reference position is set to the user define mode. The reference position is specified by the frequency or time, depending on the span state.
- [Parameter] < real > = user-defined reference position
  Frequency span: frequency (GHz/MHz/kHz/Hz)
  Zero span: time (s/ms/μs/ns)
- [Query reply] NR (real number value)
  Frequency span in Hz
  Zero span in s
- [Example] Call ibwrt (analyzer%, ":CALC:LIM:CONT:X:USER 850MHZ")
- [Relevant commands] :CALCulate<ch>:LIMit<screen>:LOWer:DATA
  :CALCulate<ch>:LIMit<screen>:UPPer:DATA
  :CALCulate<ch>:LIMit<screen>:CONTrol:X:MODE
  :CALCulate<ch>:LIMit<screen>:CONTrol:X:REFerence

5.14.81 :CALCulate<ch>:LIMit<screen>:CONTrol:X:OFFSet

- [Command syntax] :CALCulate<ch>:LIMit<screen>:CONTrol:X:OFFSet < real >
  :CALCulate<ch>:LIMit<screen>:CONTrol:X:OFFSet?
- [Function description] Sets the offset value when the horizontal axis relative value attributes of the limit line to be used are selected

  This command is used to set the offset value from the reference position when the attribute of the horizontal axis limit line data to be used is set to the relative value mode. The offset value is specified by the frequency or time, depending on the span state.
- [Parameter] < real > = offset from the user-defined reference position
  Frequency span: frequency (GHz/MHz/kHz/Hz)
  Zero span: time (s/ms/μs/ns)
- [Query reply] NR3 (real number value)
  Frequency span in Hz
  Zero span in s
- [Example] Call ibwrt (analyzer%, ":CALC:LIM:CONT:X:USER 850MHZ")
  Call ibwrt (analyzer%, ":CALC:LIM:CONT:X:OFFS 100MHZ")
- [Relevant commands] :CALCulate<ch>:LIMit<screen>:LOWer:DATA
  :CALCulate<ch>:LIMit<screen>:UPPer:DATA
  :CALCulate<ch>:LIMit<screen>:CONTrol:X:MODE
  :CALCulate<ch>:LIMit<screen>:CONTrol:X:REFerence

5.14.82 :CALCulate<ch>:LIMit<screen>:CONTrol:Y:MODE

- [Command syntax] :CALCulate<ch>:LIMit<screen>:CONTrol:Y:MODE < type >
  :CALCulate<ch>:LIMit<screen>:CONTrol:Y:MODE?
- [Function description] Selects vertical axis data attributes of the limit line to be used

  This command is used to select vertical axis data attributes of the limit line to be used in absolute or relative value. If the absolute value is selected, all limit line data is handled as absolute frequencies or absolute time. If the relative value is selected, the reference position selected by the command :CALCulate<ch>:LIMit<screen>:CONTrol:Y :REFerence is handled as the center.
- [Parameter] < type > = { ABSolute | RELative }
  ABSolute: Handles the input data as absolute value data
  RELative: Handles the input data as relative value data
- [Query reply] { ABS | REL }
- [Example] Call ibwrt (analyzer%, ":CALC:LIM:CONT:Y:MODE REL")
- [Relevant commands] :CALCulate<ch>:LIMit<screen>:LOWer:DATA
  :CALCulate<ch>:LIMit<screen>:UPPer:DATA
  :CALCulate<ch>:LIMit<screen>:CONTrol:Y:REFerence

5.14.83 :CALCulate<ch>:LIMit<screen>:CONTrol:Y:REFerence

- [Command syntax] :CALCulate<ch>:LIMit<screen>:CONTrol:Y:REFerence < type >
  :CALCulate<ch>:LIMit<screen>:CONTrol:Y:REFerence?
- [Function description] Specifies the reference position when the vertical axis relative value attributes of the limit line to be used are selected

  This command is used to select the reference position when the relative value mode is set for the vertical axis data attributes of the limit line to be used. The reference position can be set at the top of the screen, the bottom of the screen, or anywhere specified by the values defined by the user. When user-defined values are used, the values specified are set for the reference position by the :CALCulate<ch>:LIMit<screen>:CONTrol:Y:USER command.
- [Parameter] < type > = { TOP | BOTTom | USER }
  TOP: Sets the top edge of the screen as the reference position
  BOTTom:Sets the bottom edge of the screen as the reference position
  USER: Sets the user-specified position as the reference position
- [Query reply] { TOP | BOTT | USER }
- [Example]
  ```
  '-------------- Select the reference type to USER defined mode -------
  Call ibwrt (analyzer%, ":CALC:LIM:CONT:Y:REF USER")
  '-------------- Set the reference position to -35dBm ----------------
  Call ibwrt (analyzer%, ":CALC:LIM:CONT:Y:USER -35DB")
  ```
- [Relevant commands] :CALCulate<ch>:LIMit<screen>:LOWer:DATA
  :CALCulate<ch>:LIMit<screen>:UPPer:DATA
  :CALCulate<ch>:LIMit<screen>:CONTrol:Y:MODE
  :CALCulate<ch>:LIMit<screen>:CONTrol:Y:USER

5.14.84 :CALCulate<ch>:LIMit<screen>:CONTrol:Y:USER

- [Command syntax] :CALCulate<ch>:LIMit<screen>:CONTrol:Y:USER < real >
  :CALCulate<ch>:LIMit<screen>:CONTrol:Y:USER?
- [Function description] Sets the user-defined reference position when the vertical axis relative value attributes of the limit line to be used are selected

  This command is used to set the reference position when the attribute of the vertical axis limit line data to be used is set to the relative value mode and the reference position is set to the user define mode.
- [Parameter] < real > = user-defined reference position (in dBm)
- [Query reply] NR3 (real number value in dBm)
- [Example] Call ibwrt (analyzer%, ":CALC:LIM:CONT:Y:USER -25DBM")
- [Relevant commands] :CALCulate<ch>:LIMit<screen>:LOWer:DATA
  :CALCulate<ch>:LIMit<screen>:UPPer:DATA
  :CALCulate<ch>:LIMit<screen>:CONTrol:Y:MODE
  :CALCulate<ch>:LIMit<screen>:CONTrol:Y:REFerence
  :UNIT<ch>:POWer<screen>

5.14.85 :CALCulate<ch>:LIMit<screen>:CONTrol:Y:OFFSet

- [Command syntax] :CALCulate<ch>:LIMit<screen>:CONTrol:Y:OFFSet < real >
  :CALCulate<ch>:LIMit<screen>:CONTrol:Y:OFFSet?
- [Function description] Sets the offset value when the vertical axis relative value attributes of the limit line to be used are selected.

  This command is used to set the offset value when the relative value mode is set for the attribute of the vertical axis limit line data to be used.
- [Parameter] < real > = offset level from the user-defined reference position (in dB)
- [Query reply] NR3 (real number value in dB)
- [Example] Call ibwrt (analyzer%, ":CALC:LIM:CONT:Y:USER -35DB")
  Call ibwrt (analyzer%, ":CALC:LIM:CONT:Y:OFFS 3DB")
- [Relevant commands] :CALCulate<ch>:LIMit<screen>:LOWer:DATA
  :CALCulate<ch>:LIMit<screen>:UPPer:DATA
  :CALCulate<ch>:LIMit<screen>:CONTrol:Y:MODE
  :CALCulate<ch>:LIMit<screen>:CONTrol:Y:REFerence

5.14.86 :CALCulate<ch>:LIMit<screen>:FAIL?

- [Command syntax] :CALCulate<ch>:LIMit<screen>:FAIL?
- [Function description] Reads the Pass/Fail judgment by the limit line

  You can compare the waveform and limit line to see whether the waveform is above or below the limit line when you use the Pass/Fail judgment function with the limit line displayed. A PASS judgment is made if the relationship between the waveform and limit line satisfies the predetermined relationship over all points. A FAIL judgment is made if any point is off the predetermined relationship. This command is used to read the judgment result.

  Judgment is made after each sweep.
- [Parameter] None
- [Query reply] { PASS | FAIL }
- [Example]

  ```
  Judge$ = Space$(30)
  Call ibwrt (analyzer%, ":CALC:LIM:LOW:PASS ABOVE")
  Call ibwrt (analyzer%, ":CALC:LIM:UPP:PASS BEL")
  Call ibwrt (analyzer%, ":CALC:LIM:FAIL?")
  Call ibrd (analyzer%, Judge$)
  ```
- [Relevant commands] :CALCulate<ch>:LIMit<screen>:LOWer:DATA

  :CALCulate<ch>:LIMit<screen>:UPPer:DATA

  :CALCulate<ch>:LIMit<screen>:CONTrol:Y:MODE

  :CALCulate<ch>:LIMit<screen>:CONTrol:Y:REFerence

  :CALCulate<ch>:LIMit<screen>:STATe

  :CALCulate<ch>:LIMit<screen>:LOWer:PASS

  :CALCulate<ch>:LIMit<screen>:UPPer:PASS

5.14.87 :CALCulate<ch>:LIMit<screen>:UPPer:COPY :CALCulate<ch>:LIMit<screen>:LOWer:COPY

- [Command syntax] :CALCulate<ch>:LIMit<screen>:UPPer:COPY
  :CALCulate<ch>:LIMit<screen>:LOWer:COPY
- [Function description] Copies the data for limit line 1 to limit line 2
  Copies the data for limit line 2 to limit line 1

  The command :CALCulate<ch>:LIMit<screen>:UPPer:COPY is used to copy the data of Limit Line Table 1 to Limit Line Table 2.

  The command :CALCulate<ch>:LIMit<screen>:LOWer:COPY is used to copy the data of Limit Line Table 2 to Limit Line Table 1.

  Before sending this command, use the :CALCulate<ch>:LIMit<screen>:CONTrol:X:DOMain command to determine whether the table to be copied is for the frequency domain or the time domain.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:LIM:CONT:X:DOM FREQ")
  Call ibwrt (analyzer%, ":CALC:LIM:LOW:COPY")
- [Relevant commands] :CALCulate<ch>:LIMit<screen>:CONTrol:X:DOMain
  :CALCulate<ch>:LIMit<screen>:UPPer:DATA
  :CALCulate<ch>:LIMit<screen>:LOWer:DATA

5.14.88 :CALCulate<ch>:LIMit<screen>:UPPer:DATA :CALCulate<ch>:LIMit<screen>:LOWer:DATA

- [Command syntax] :CALCulate<ch>:LIMit<screen>:UPPer:DATA < real,real >
  :CALCulate<ch>:LIMit<screen>:LOWer:DATA < real,real >
- [Function description] Inputs data to limit line 1
  Inputs data to limit line 2

  This command is used to input data to Limit Line Table 1 or 2.

  Up to 50 items of data can be input to the Limit Line Table. If the target table contains data before this command is sent, the new data is added after the existing data in the table.Therefore, if you want to input data from the beginning, use the :CALCulate<ch>:LIMit<screen>:UPPer:DELete command or :CALCulate<ch>:LIMit<screen>:LOWer:DELete command to delete all data from the target table before sending this command. Note that before sending this command, use the :CALCulate<ch>:LIMit<screen>:CONTrol:X:DOMain command to determine whether the table to which the data is to be input is for the frequency domain or the time domain.
- [Parameter] < real, real > =
  Frequency domain: Frequency (GHz/MHz/kHz/Hz), level (dBm or dB)
  Time domain: Time (s/ms/μs/ns), level (dBm or dB)
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:LIM:CONT:X:DOM FREQ")
  Call ibwrt (analyzer%, ":CALC:LIM:LOW:DATA -20MHZ, -45DB")
  Call ibwrt (analyzer%, ":CALC:LIM:LOW:DATA -10MHZ, -45DB")
  Call ibwrt (analyzer%, ":CALC:LIM:LOW:DATA -10MHZ, 0DB")
  Call ibwrt (analyzer%, ":CALC:LIM:LOW:DATA 0MHZ, 0DB")
  Call ibwrt (analyzer%, ":CALC:LIM:LOW:DATA 10MHZ, 0DB")
  Call ibwrt (analyzer%, ":CALC:LIM:LOW:DATA 10MHZ, -45DB")
  Call ibwrt (analyzer%, ":CALC:LIM:LOW:DATA 20MHZ, -45DB")
- [Relevant commands] :CALCulate<ch>:LIMit<screen>:CONTrol:X:DOMain
  :CALCulate<ch>:LIMit<screen>:UPPer:DATA
  :CALCulate<ch>:LIMit<screen>:UPPer:COPY
  :CALCulate<ch>:LIMit<screen>:LOWer:DATA
  :CALCulate<ch>:LIMit<screen>:LOWer:COPY
  :UNIT<ch>:POWer<screen>

5.14.89 :CALCulate<ch>:LIMit<screen>:UPPer:DELete :CALCulate<ch>:LIMit<screen>:LOWer:DELete

- [Command syntax] :CALCulate<ch>:LIMit<screen>:UPPer:DELete
  :CALCulate<ch>:LIMit<screen>:LOWer:DELete
- [Function description] Deletes the data from Limit Line Table 1
  Deletes the data from Limit Line Table 2

  This command is used to delete all data input to Limit Line Table 1 or 2
  Before sending this command, use the command :CALCulate<ch>:LIMit<screen>:CONTrol:X:DOMain to determine whether the table to be deleted is for the frequency domain or the time domain.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":CALC:LIM:CONT:X:DOM FREQ")
  Call ibwrt (analyzer%, ":CALC:LIM:LOW:DEL")
- [Relevant commands] :CALCulate<ch>:LIMit<screen>:CONTrol:X:DOMain
  :CALCulate<ch>:LIMit<screen>:UPPer:DATA
  :CALCulate<ch>:LIMit<screen>:LOWer:DATA

5.14.90 :CALCulate<ch>:LIMit<screen>:UPPer:PASS :CALCulate<ch>:LIMit<screen>:LOWer:PASS

- [Command syntax] :CALCulate<ch>:LIMit<screen>:UPPer:PASS < type>
  :CALCulate<ch>:LIMit<screen>:UPPer:PASS?
  :CALCulate<ch>:LIMit<screen>:LOWer:PASS < type>
  :CALCulate<ch>:LIMit<screen>:LOWer:PASS?
- [Function description] Sets the judgment condition for Pass/Fail judgment by Limit Line Table 1 or 2

  This command is used to set the judgment condition for Pass/Fail judgment using Limit Line Table 1 or 2. The judgment condition is expressed by the positional relationship between the waveform and the limit line. ABOVE designates that the waveform is above the limit line with all points. BELOW designates that the waveform is below the limit line with all points. A PASS judgment is made when the waveform and limit line are on the same level.
- [Parameter] < type > = { ABOVe | BELow }
  ABOVe: A PASS judgment is made when the waveform is positioned above the limit line
  BELow: A FAIL judgment is made when the waveform is positioned below the limit line
- [Query reply] { ABOV | BEL }
- [Example] Call ibwrt (analyzer%, ":CALC:LIM:CONT:X:DOM FREQ")
  Call ibwrt (analyzer%, ":CALC:LIM:LOW:PASS ABOV")
- [Relevant commands] :CALCulate<ch>:LIMit<screen>:CONTrol:X:DOMain
  :CALCulate<ch>:LIMit<screen>:UPPer:DATA
  :CALCulate<ch>:LIMit<screen>:LOWer:DATA

5.14.91 :CALCulate<ch>:LIMit<screen>:UPPer:STATe :CALCulate<ch>:LIMit<screen>:LOWer:STATe

- [Command syntax]
  :CALCulate<ch>:LIMit<screen>:UPPer:STATe < bool >
  :CALCulate<ch>:LIMit<screen>:UPPer:STATe?
  :CALCulate<ch>:LIMit<screen>:LOWer:STATe < bool >
  :CALCulate<ch>:LIMit<screen>:LOWer:STATe?
- [Function description]
  Sets ON or OFF the display of limit line 1 or 2

  This command is used to set ON or OFF the display of limit line 1 or 2. When Pass/Fail judgment is set on by the command :CALCulate<ch>:LIMit<screen>:STATe with the display of limit line 1 or 2 set on, a waveform is compared with the limit line each time a sweep is performed.
- [Parameter]
  < bool > = { OFF | ON }
  OFF: Does not display limit line 1 or 2
  ON: Displays limit line 1 or 2
- [Query reply]
  { OFF | ON }
- [Example]

```
'------------- Define limit line #2 data ----------------------------------------
Call ibwrt (analyzer%, ":CALC:LIM:CONT:X:DOM FREQ")
Call ibwrt (analyzer%, ":CALC:LIM:LOW:DATA -10MHZ, -45DB")
Call ibwrt (analyzer%, ":CALC:LIM:LOW:DATA -10MHZ,  0DB")
Call ibwrt (analyzer%, ":CALC:LIM:LOW:DATA  0MHZ,  0DB")
Call ibwrt (analyzer%, ":CALC:LIM:LOW:DATA 10MHZ,  0DB")
Call ibwrt (analyzer%, ":CALC:LIM:LOW:DATA 10MHZ, -45DB")
'------------- Set limit line #2 to ON ------------------------------------------
Call ibwrt (analyzer%, ":CALC:LIM:LOW:STAT ON")
```

- [Relevant commands]
  :CALCulate<ch>:LIMit<screen>:CONTrol:X:DOMain
  :CALCulate<ch>:LIMit<screen>:UPPer:DATA
  :CALCulate<ch>:LIMit<screen>:LOWer:DATA

5.14.92 :CALCulate<ch>:LIMit<screen>:STATe

- [Command syntax] :CALCulate<ch>:LIMit<screen>:STATe < bool >
  :CALCulate<ch>:LIMit<screen>:STATe?
- [Function description] Sets ON or OFF the Pass/Fail judgment function that uses limit line 1 or 2

  This command is used to set ON or OFF the Pass/Fail judgment function that uses limit line 1 or 2. When limit line 1 or 2 is displayed, the waveform and limit line are compared for their positional relationship to perform the Pass/Fail judgment according to the set judgment condition.
- [Parameter] < bool > = { OFF | ON }
  OFF: Does not perform the Pass/Fail judgment that uses limit line 1 or 2
  ON: Performs the Pass/Fail judgment that uses limit line 1 or 2
- [Query reply] { OFF | ON }
- [Example]
  '--------- Set the judgement condition ---------------------
  Call ibwrt (analyzer%, ":CALC:LIM:UPP:PASS BEL")
  Call ibwrt (analyzer%, ":CALC:LIM:UPP:STAT ON")
  '---------- Start Pass/Fail judgement ----------------------
  Call ibwrt (analyzer%, ":CALC:LIM:STAT ON")
- [Relevant commands] :CALCulate<ch>:LIMit<screen>:UPPer:PASS
  :CALCulate<ch>:LIMit<screen>:LOWer:PASS

5.14.93 :CALCulate<ch>:WINDow<screen>:POSition

- [Command syntax] :CALCulate<ch>:WINDow<screen>:POSition < real >
  :CALCulate<ch>:WINDow<screen>:POSition?
- [Function description] Sets the Measuring Window position

  This command is used to set the display position of the Measuring Window by the frequency when frequency span is used or by the time when zero span is used. As the display position, specify the center of the Window for frequency or the left start position of the Window for time.
- [Parameter] < real > = Window display position
  When frequency span is used: frequency (GHz/MHz/kHz/Hz)
  When zero span is used: time (s/ms/μs/ns)
- [Query reply] NR3 (Real value)
  When frequency span is used: frequency (in Hz)
  When zero span is used: time (in s)
- [Example] '---------- Set window's position and width ----------------------
  Call ibwrt (analyzer%, ":CALC:WIND:POS 2.122GHZ")
  Call ibwrt (analyzer%, ":CALC:WIND:WIDT 3.84MHZ")
- [Relevant commands] :CALCulate<ch>:WINDow<screen>:STATe
  :CALCulate<ch>:WINDow<screen>:WIDTh

5.14.94 :CALCulate<ch>:WINDow<screen>:WIDTh

- [Command syntax] :CALCulate<ch>:WINDow<screen>:WIDTh < real >
  :CALCulate<ch>:WINDow<screen>:WIDTh?
- [Function description] Sets the width of the Measuring Window

  This command is used to set the display width of the Measuring Window by the frequency when frequency span is used or by the time when zero span is used. When frequency span is used, a Window of half the specified width left and right from the center of the Window is displayed.When zero span is used, a Window of the specified width from the leftmost position is displayed.
- [Parameter] < real > = Window width
  When frequency span is used: frequency (GHz/MHz/kHz/Hz)
  When zero span is used: time (s/ms/µs/ns)
- [Query reply] NR3 (Real value)
  When frequency span is used: frequency (in Hz)
  When zero span is used: time (in s)
- [Example]
  '--------- Set window's position and width ---------------------
  Call ibwrt (analyzer%, ":CALC:WIND:POS 2.122GHZ")
  Call ibwrt (analyzer%, ":CALC:WIND:WIDT 3.84MHZ")
- [Relevant commands] :CALCulate<ch>:WINDow<screen>:STATe
  :CALCulate<ch>:WINDow<screen>:POSition

5.14.95 :CALCulate<ch>:WINDow<screen>:STATe

- [Command syntax] :CALCulate<ch>:WINDow<screen>:STATe < bool>
  :CALCulate<ch>:WINDow<screen>:STATe?
- [Function description] Sets ON or OFF the display of the Measuring Window

  This command is used to set ON or OFF the display of the Measuring Window.
- [Parameter] < bool > = { OFF | ON }
  OFF: Sets the display of the Measuring Window off
  ON: Sets the display of the Measuring Window on
- [Query reply] { OFF | ON }
- [Example]
  '---------- Set window's position and width ---------------------
  Call ibwrt (analyzer%, ":CALC:WIND:POS 2.122GHZ")
  Call ibwrt (analyzer%, ":CALC:WIND:WIDT 3.84MHZ")
  '---------- Measurement Window ON ----------------------------
  Call ibwrt (analyzer%, ":CALC:WIND:STAT ON")
- [Relevant commands] :CALCulate<ch>:WINDow<screen>:POSition
  :CALCulate<ch>:WINDow<screen>:WIDTh

5.14.96 :CALCulate<ch>:CURSor<screen>:ANCHor

- [Command syntax] :CALCulate<ch>:CURSor<screen>:ANCHor < bool >
  :CALCulate<ch>:CURSor<screen>:ANCHor?
- [Function description] Sets ON or OFF the anchor function of the X and Y cursors

  This command is used to set an anchor at the intersection of the X and Y cursors when the cursors are displayed. This anchor function allows you to compare the relative values of the anchor position and waveform.
- [Parameter] < bool > = { OFF | ON }
  OFF: Sets the anchor function of the X and Y cursors off
  ON: Sets the intersection of the X and Y cursors as the anchor and turns on the anchor function
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":CALC:CURS:ANCH ON")
- [Relevant commands] :CALCulate<ch>:CURSor<screen>:X
  :CALCulate<ch>:CURSor<screen>:Y
  :CALCulate<ch>:CURSor<screen>:STATe

5.14.97 :CALCulate<ch>:CURSor<screen>:X

- [Command syntax] :CALCulate<ch>:CURSor<screen>:X < real >
  :CALCulate<ch>:CURSor<screen>:X?
- [Function description] Sets the display position of the X cursor when the X and Y cursors are displayed

  This command is used to set the display position of the X cursor by the frequency when frequency span is used or by the time when zero span is used, when the X and Y cursors are displayed.
- [Parameter] < real > = display position of the X cursor
  When frequency span is used: frequency (GHz/MHz/kHz/Hz)
  When zero span is used: time (s/ms/μs/ns)
- [Query reply] NR3 (Real value)
  When frequency span is used: frequency (in Hz)
  When zero span is used: time (in s)
- [Example] Call ibwrt (analyzer%, ":CALC:CURS:X 2GHZ")
- [Relevant commands] :CALCulate<ch>:CURSor<screen>:ANCHor
  :CALCulate<ch>:CURSor<screen>:STATe
  :CALCulate<ch>:CURSor<screen>:Y

5.14.98 :CALCulate<ch>:CURSor<screen>:Y

- [Command syntax] :CALCulate<ch>:CURSor<screen>:Y < real >
  :CALCulate<ch>:CURSor<screen>:Y?
- [Function description] Sets the display position of the Y cursor when the X and Y cursors are displayed

  This command is used to set the display position of the Y cursor by the level value when the X and Y cursors are displayed.
- [Parameter] < real > = display position of the Y cursor (in dBm)
- [Query reply] NR3 (real number value in dBm)
- [Example] Call ibwrt (analyzer%, ":CALC:CURS:Y -35DB")
- [Relevant commands] :CALCulate<ch>:CURSor<screen>:ANCHor
  :CALCulate<ch>:CURSor<screen>:STATe
  :CALCulate<ch>:CURSor<screen>:X
  :UNIT<ch>:POWer<screen>

5.14.99 :CALCulate<ch>:CURSor<screen>:STATe

- [Command syntax] :CALCulate<ch>:CURSor<screen>:STATe < bool >
  :CALCulate<ch>:CURSor<screen>:STATe?
- [Function description] Sets ON or OFF the display of the X and Y cursors

  This command is used to set ON or OFF the display of the X and Y cursors.
- [Parameter] < bool > = { OFF | ON }
  OFF: Sets the display of the X and Y cursors off
  ON: Sets the display of the X and Y cursors on
- [Query reply] { OFF | ON }
- [Example] Call ibwrt (analyzer%, ":CALC:CURS:STAT ON")
- [Relevant commands] :CALCulate<ch>:CURSor<screen>:ANCHor
  :CALCulate<ch>:CURSor<screen>:X
  :CALCulate<ch>:CURSor<screen>:Y

5.15 Unit Command

This section describes the Unit subsystem.

In the Unit subsystem, the command used to set the level unit is defined.

Command	Function	Reference Page
:UNIT<ch> :POWer<screen>	Sets the level unit system	5-304

5.15.1 :UNIT<ch>:POWer<screen>

- [Command syntax] :UNIT<ch>:POWer<screen> < type >
  :UNIT<ch>:POWer<screen>?
- [Function description] Sets the level unit system
- [Parameter] < type > = { DBM | DBMV | DBUV | DBUE | DBPW | VOLT | WATT }
  DBM: Unit dBm
  DBMV: Unit dBmV
  DBUV: Unit dBμV
  DBUE: Unit dBμVemf
  DBPW: Unit dBpW
  VOLT: Unit Volt
  WATT: Unit Watt
- [Query reply] { DBM | DBMV | DBUV | DBUE | DBPW | VOLT | WATT }
- [Example] Call ibwrt (analyzer%, ":UNIT:POW WATT")
- [Relevant commands] :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:RLEVel
  :DISPlay<ch>[:WINDow<screen>]:TRACe:Y[:SCALe]:RLEVel:OFFSet

5.16 System Commands

This section describes the System subsystem.

In the System subsystem, the commands used to initialize and switch measurement systems of this instrument are defined.

Command	Function	Reference Page
:SYSTem		
:PRESet	Initialization of the current measurement systems	5-305
:ALL	Initialization of all measurement systems	5-306
:SELect	Selection of a measurement system	5-306
:MODulation	Selection of the modulation analysis system	5-307
:STANdard	Selects the Standard mode	5-308
:ERRor?	Last error inquiry	5-309
:ALL?	Error log contents inquiry	5-309

5.16.1 :SYSTem:PRESet

- [Command syntax] :SYSTem:PRESet
- [Function description] Initialization of the current measurement systems

  Initializes the parameters corresponding to the measurement systems (Spectrum Analyzer, Modulation Analyzer, etc.) that can be supported by this instrument.

  Initializes the system parameters of the measurement system being used when this command is issued.

  As a result of the initialization, the measurement statuses and setting parameters are set to the factory defaults. When ALL is sent as the parameter of this command, the measurement statuses and setting parameters of all measurement systems are simultaneously initialized and this instrument is set to the spectrum analyzer measurement system status.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":SYST:PRES")
- [Relevant commands] :SYSTem:PRESet:ALL

5.16.2 :SYSTem:PRESet:ALL

- [Command syntax] :SYSTem:PRESet:ALL
- [Function description] Initialization of all measurement systems

  Initializes all parameters corresponding to the measurement systems (Spectrum Analyzer, Modulation Analyzer, etc.) that can be supported by this instrument.

  As a result of the initialization, the measurement statuses and setting parameters are set to the factory defaults.

  Executing this command sets this instrument to the spectrum analyzer measurement system status.
- [Parameter] None
- [Query reply] None
- [Example] Call ibwrt (analyzer%, ":SYST:PRES:ALL")
- [Relevant commands] *RST

5.16.3 :SYSTem:SELect

- [Command syntax] :SYSTem:SELect < type >

  :SYSTem:SELect?
- [Function description] Selection of a measurement system

  Switches the measurement systems (Spectrum Analyzer, Modulation Analyzer, etc.) that can be supported by this instrument. The measurement systems that can be switched differ depending on the options supported by this instrument.
- [Parameter] < type > = { SANalyzer | MANalyzer | .... }

  SANalyzer: Spectrum analyzer measurement system

  MANalyzer: Modulation analyzer measurement system (Parameters are added depending on the options settings of this instrument.)
- [Query reply] { SAN | MAN | .... }
- [Example] '------ Change to Spectrum Analyzer mode --------

  Call ibwrt (analyzer%, ":SYST:SEL SAN")
- [Relevant commands] :SYTem:SELect:MODulation

5.16.4 :SYSTem:SELect:MODulation

- [Command syntax] :SYSTem:SELect:MODulation < type >
  :SYSTem:SELect:MODulation?
- [Function description] Selection of the modulation analysis system

  Switches the modulation analysis measurement systems (OFDM Analyzer, etc.) that can be supported by this instrument in the modulation analyzer measurement system status. The measurement systems that can be switched differ depending on the options supported by this instrument.

  To switch directly to the desired modulation analysis measurement system using this command, the measurement system status of this instrument must be set to the modulation analyzer measurement system status in advance.
- [Parameter] < type > = { OFDM }

  OFDM: OFDM modulation analysis measurement system (Parameters are added depending on the options settings of this instrument.)
- [Query reply] { OFDM }
- [Example] '------ Change to Modulation Analyzer mode --------
  Call ibwrt (analyzer%, ":SYST:SEL MAN")
  Call ibwrt (analyzer%, ":SYST:SEL:MOD OFDM")
- [Relevant commands] :SYTem:SELect

5.16.5 :SYSTem:SELect:STANdard

- [Command syntax] :SYSTem:SELect:STANdard < type1 >, < type2 >

  :SYSTem:SELect:STANdard?

- [Function description] Selects the Standard mode.

  Selects the Standard mode concerning the channel setting and the default value of the measurement function in the spectrum analyzer measurement system.

  Can be selected only when the Reference information exists.

- [Parameter] < type1 > = { OFF | 3GPP | cdma2000 | ... }

  OFF: The Standard mode is set to OFF. <type2> is omitted.

  3GPP etc. :

  Standard name.
  Sets the same string which is set in [Type] in the dialog box displayed by [Special]→[STD].

  < type2 > = { 3GPP_UL | ... }

  3GPP_UL etc. :

  Sets the same string which is set in [Meas. Mode] in the dialog box displayed by [Special]→[STD].
  (The parameter is included by adding the Standard information file.)

- [Query reply] The Standard mode is set to OFF. { OFF }

  The Standard mode is set to ON. { < type1 > , < type2 > }

- [Example]

```
'------ Change STD mode to OFF--------
Call ibwrt (analyzer%, ":SYST:SEL:STAN ""OFF""")

'------ Change STD mode to 3GPP Up Link--------
Call ibwrt (analyzer%, ":SYST:SEL:STAN ""3GPP"", ""3GPP_UL""")
```

- [Relevant commands]

5.16.6 :SYSTem:ERRor?

- [Command syntax] :SYSTem:ERRor?
- [Function description] Last error inquiry

  Among various errors that occurred while controlling this instrument, the error code number of the last error that occurred and the corresponding error message string are returned.
- [Parameter] None
- [Query reply] NR1 , < str >

  Error number, error message string

  For more information on the error numbers and contents, refer to Section 8.8 Error Message List of the R3681 SERIES USER'S GUIDE.
- [Example]

```
ErrMess$ = Space(60)
Call ibwrt (analyer%, ":SYST:ERR?")
Call ibrd (analyzer%, ErrMess$)
OutputMsgs " Last Error : "&ErrMess$
```

- [Relevant commands] :SYSTem:ERRor:ALL?

5.16.7 :SYSTem:ERRor:ALL?

- [Command syntax] :SYSTem:ERRor:ALL?
- [Function description] Error log contents inquiry

  Up to 10 most recent errors encountered while controlling this instrument are saved in the instrument. This command is used to output all the saved error numbers and messages.

  When receiving data from this command, sufficient available buffer space for saving 10 error numbers and messages is necessary.
- [Parameter] None
- [Query reply] NR1 , < str >

  Error number, error message string

  For more information on the error numbers and contents, refer to Section 8.8 Error Message List of the R3681 SERIES USER'S GUIDE.
- [Example]

```
ErrMess$ = Space(6200)
Call ibwrt (analyer%, ":SYST:ERR:ALL?")
Call ibrd (analyzer%, ErrMess$)
OutputMsgs " Last Error : "&ErrMess$
```

- [Relevant commands] :SYSTem:ERRor?

5.17 Diagnostic Commands

This section describes the Diagnostic subsystem.

In the Diagnostic subsystem, the commands used to read the result of the self-diagnostic function to be executed when the power of this instrument is turned on are defined.

Command	Function	Reference Page
:DIAGnostic		
:PON?	Reading of the result of the Power on DIAG	5-310
:SELFtest?	Execution of the Self-Test and reading of the result	5-311

5.17.1 :DIAGnostic:PON?

- [Command syntax] :DIAGnostic:PON?
- [Function description] Reading of the result of the Power on DIAG

  Reads the PASS/FAIL result of the Power on DIAG to be executed when the power of this instrument is turned on.
- [Parameter] None
- [Query reply] { PASS | FAIL }

  PASS: The results of all items of the Power on DIAG are PASS

  FAIL: The result of one or more items of the Power on DIAG are FAIL
- [Example]

```
Call ibwrt (analyzer%, ":DIAG:PON?")
Call ibrd(analyzer%, res$)
OutputMsgs "Power on Diag:" & res$
```

- [Relevant commands] :DIAGnostic:SELFtest?

5.17.2 :DIAGnostic:SELFtest?

- [Command syntax] :DIAGnostic:SELFtest?
- [Function description] Execution of the Self-Test and reading of the result

  Executes the post-diagnosis function of this instrument. The PASS/FAIL result can be read after the execution.
- [Parameter] None
- [Query reply] { PASS | FAIL }

  PASS: The results of all items of the Self-Test are PASS

  FAIL: The result of one or more items of the Self-Test are FAIL
- [Example] Call ibwrt (analyzer%, ":DIAG:SELF?")

  Call ibrd(analyzer%, res$)

  OutputMsgs "Selftest:" & res$
- [Relevant commands] :DIAGnostic:PON?

5.18 Status Commands

This section describes the Status subsystem.

In the Status subsystem, the commands used to set the data to the registers supported by this instrument (listed below) and read the data on the registers are defined.

- Standard operation status register
- Questionable status register
- Measuring status register

MEMO: For more information on the registers, refer to Chapter 2 "2.6 Status Bytes."

Command	Function	Reference Page
:STATus		
:OPERation		
:ENABle	Setting the standard operation status enable register to Enable	5-313
:EVENt	Reading of the standard operation status event register	5-313
:MEASure		
:ENABle	Setting the measuring status enable register to Enable	5-315
:EVENt	Reading of the measuring status event register	5-315
:QUEStionable		
:ENABle	Setting the questionable status enable register to Enable	5-314
:EVENt	Reading of the questionable status event register	5-314

5.18.1 :STATus:OPERation:ENABle

- [Command syntax] :STATus:OPERation:ENABle < int >
  :STATus:OPERation:ENABle?
- [Function description] Setting the standard operation status enable register to Enable

  Sets Enable conditions for the standard operation status enable register.
- [Parameter] < int > = 0 - 65535
  The set value is the logical sum of the positions of bits that you want to set to Enable in a decimal number.
  For example, when you want to set the stop bits for calibration and averaging to Enable, set 257 (CALIbrating bit 1 + AVERaging bit 256).
- [Query reply] NR1 (Integer without unit)
- [Example]
```
'---- Set to enable, CALIbrating bit and AVERaging bit ----
Call ibwrt (analyzer%, ":STAT:OPER:ENAB 257")
```
- [Relevant commands] :STATus:OPERation:EVENt?

5.18.2 :STATus:OPERation:EVENt?

- [Command syntax] :STATus:OPERation:EVENt?
- [Function description] Reading of the standard operation status event register

  Reads the current status of the standard operation status event register.
- [Parameter] None
- [Query reply] NR1 (Integer: The setting status of bits in the register)
- [Example]
```
StdOPR$ = Space$(128)
Call ibwrt(analyzer%, "STAT:OPER:EVEN?")        ' Read ST OP Reg.
Call ibrd(analyzer%, StdOPR$)
If (Val(StdOPR$) AND 1) > 0 Then OutputMsgs "Calibration bit"
If (Val(StdOPR$) And 4) > 0 Then OutputMsgs "Ranging bit"
If (Val(StdOPR$) And 8) > 0 Then OutputMsgs "Sweep bit"
If (Val(StdOPR$) And 16) > 0 Then OutputMsgs "Measurement bit"
```
- [Relevant commands] :STATus:OPERation:ENABle

5.18.3 :STATus:QUEStionable:ENABle

- [Command syntax] :STATus:QUEStionable:ENABle < int >
  :STATus:QUEStionable:ENABle?
- [Function description] Setting the questionable status enable register to Enable

  Sets Enable conditions for the questionable status enable register.
- [Parameter] < int > = 0 - 65535
  The set value is the logical sum of the positions of bits that you want to set to Enable in a decimal number.
- [Query reply] NR1 (Integer without unit)
- [Example] '---- Set to enable, Measurement Uncal bit ----
  Call ibwrt (analyzer%, ":STAT:QUES:ENAB 512")
- [Relevant commands] :STATus:QUEStionable:EVENt?

5.18.4 :STATus:QUEStionable:EVENt?

- [Command syntax] :STATus:QUEStionable:EVENt?
- [Function description] Reading of the questionable status event register

  Reads the current status of the questionable status event register.
- [Parameter] None
- [Query reply] NR1 (Integer: The setting status of bits in the register)
- [Example]
  ```
  QuesREG$ = Space$(128)
  Call ibwrt(analyzer%, "STAT:QUES:EVEN?")        ' Read Reg.
  Call ibrd(analyzer%, QuesREG$)
  If (Val(QuesREG$) And 512) > 0 Then OutputMsgs "Meas. Uncal bit"
  ```
- [Relevant commands] :STATus:QUEStionable:ENABle

5.18.5 :STATus:OPERation:MEASure:ENABle

- [Command syntax] :STATus:OPERation:MEASure:ENABle < int >
  :STATus:OPERation:MEASure:ENABle?
- [Function description] Setting the measuring status enable register to Enable

  Sets Enable conditions for the measuring status enable register.
- [Parameter] < int > = 0 - 65535
  The set value is the logical sum of the positions of bits that you want to set to Enable in a decimal number.
- [Query reply] NR1 (Integer without unit)
- [Example] '---- Set to enable, Measuring end bit for SA ----
  Call ibwrt (analyzer%, ":STAT:OPER:MEAS:ENAB 1")
- [Relevant commands] :STATus:OPERation:MEASure:EVENt?

5.18.6 :STATus:OPERation:MEASure:EVENt?

- [Command syntax] :STATus:OPERation:MEASure:EVENt?
- [Function description] Reading of the measuring status event register

  Reads the current status of the measuring status event register.
- [Parameter] None
- [Query reply] NR1 (Integer: The setting status of bits in the register)
- [Example] MeasREG$ = Space$(20)
  Call ibwrt(analyzer%, "STAT:OPER:MEAS:EVEN?")' Read Reg.
  Call ibrd(analyzer%, MeasREG$)
  If (Val(MeasREG$) And 1) > 0 Then OutputMsgs "SA Meas. bit"
- [Relevant commands] :STATus:QUEStionable:MEASure:ENABle

5.19 Hard Copy Commands

This section describes the Hard Copy subsystem.

The Hard Copy subsystem defines the commands for outputting screen data of this instrument as hard copy.

Command	Function	Reference Page
:HCOPy		
[:IMMediate]	Outputting a copy to a file or a printer	5-316
:DESTination	Specifying the output destination	5-317
:MMEMory		
:FILE		
:NUMBer	Specifying the output file number	5-318
:TYPE	Specifying the output file type	5-319

5.19.1 :HCOPy[:IMMediate]

- [Command syntax] :HCOPy[:IMMediate]
- [Function description] Outputs a copy to a file or a printer

  This command outputs the hard copy data on the current screen to a file or the printer port set as the standard output destination. Before using this command, the output destination (file or printer) must be specified.

  When output to a file is specified, the four-digit number attached to the file name and the file format must be specified.

  When output to a printer is specified, it is necessary to connect this instrument to the printer and set the printer driver before using this command.
- [Parameter] None
- [Query reply] None
- [Example]

```
'---------- Set output device, file name and file type ----------------
'------------ Output device is file              ----------------
'------------ Output file name is ADV0001        ----------------
'------------ Set file type to PNG               ----------------
Call ibwrt (analyzer%, ":HCOP:DEST MMEM")
Call ibwrt (analyzer%, ":HCOP:MMEM:FILE:NUMB 0001")
Call ibwrt (analyzer%, ":HCOP:MMEM:FILE:TYPE PNG")
Call ibwrt (analyzer%, ":HCOP:IMM")             ' Execute output
```

- [Relevant commands] :HCOPy:DESTination
  :HCOPy:MMEMory:FILE:NUMBer
  :HCOPy:MMEMory:FILE:TYPE

5.19.2 :HCOPy:DESTination

- [Command syntax] :HCOPy:DESTination < type >
  :HCOPy:DESTination?
- [Function description] Specifies the output destination

  This command specifies the output destination of the hard copy data on the current screen.
  The following three output destinations can be specified.

  - A file on the hard disk in this instrument
  - A file on the external disk
  - The printer that is the standard output port

  If a file on the hard disk is specified as the output destination, the hard copy data on the current screen is output in the format that was selected from the bitmap format (BMP format) or the portable network graphics format (PNG format) in advance. The directory on the hard disk of the output destination is "D:\ADVANTEST\R3681\MyData." For the file name, a file name with a four-digit file number specified by the :HCOPy:MMEMory:FILE:NUMBer command is used. This file number is not changed unless it is specified by the :HCOPy:MMEMory:FILE:NUMBer command again. Unless the file number is changed before output, data is overwritten on the same file. If 0001 is specified as the file number, for example, the output file name is ADV0001.

  If a file on the external disk is specified as the output destination, the directory, in which the output destination file is located, is "F:\IMAGE."

  If a printer is specified, data is output to the printer connected to the PRINTER port on the rear side of this instrument. Before outputting data to the printer, it is necessary to install the driver suited for the printer in this instrument with the special installer for the printer in question by using the "Printers Settig..." function defined in the Config menu in the menu bar.
- [Parameter] < type > = { MMEMory | FMEMory | PRINt }

  MMEMory: Data is output to hard disk D in this instrument in file format

  FMEMory: Data is output to external disk F in file format

  PRINt: Data is output to the printer connected to the PRINTER output port on the rear side of this instrument
- [Query reply] { MMEM | FMEM | PRIN }

- [Example] Call ibwrt (analyzer%, ":HCOP:DEST PRIN") ' Set destination to printer
- [Relevant commands] :HCOPy:[:IMMediate]
  :HCOPy:MMEMory:FILE:NUMBer
  :HCOPy:MMEMory:FILE:TYPE

5.19.3 :HCOPy:MMEMory:FILE:NUMBer

- [Command syntax] :HCOPy:MMEMory:FILE:NUMBer < int >
  :HCOPy:MMEMory:FILE:NUMBer?
- [Function description] Specifies the file number

  This command specifies the file number used as the output file name when a file is specified as the hard copy data output destination on the screen.

  The four-digit number specified in this command is used as a part of the file name. By specifying the sequential number when outputting a file, files can be accumulated without overwriting. If a file with the same file name exists, data is overwritten unconditionally.

  The relation between the file number and the file name is shown below.

  When 1234 is specified as the file number, the file name is ADV1234.
- [Parameter] < int > = File number
  An integer of up to four digits (0 - 9999)
- [Query reply] NR1 (Integer: No unit)
- [Example] Call ibwrt (analyzer%, ":HCOP:DEST MMEM")
  Call ibwrt (analyzer%, ":HCOP:MMEM:FILE:NUMB 1234")
- [Relevant commands] :HCOPy:[:IMMediate]
  :HCOPy:DESTination
  :HCOPy:MMEMory:FILE:TYPE

5.19.4 :HCOPy:MMEMory:FILE:TYPE < type >

- [Command syntax] :HCOPy:MMEMory:FILE:TYPE < type >
  :HCOPy:MMEMory:FILE:TYPE?
- [Function description] Specifies the output file type

  This command specifies the output file type when a file is selected as the output destination of the hard copy data on the screen.
  The following two file types can be specified.

  - Bit map file (BMP format)
  - Portable network graphics file (PNG format)

  The output file extension differs depending on the file format selected.
  For the BMP format, the extension is BMP. For the PNG format, the extension is PNG.
- [Parameter] < type > = { BITMap | PNGraphic }
  BITMap: BMP format
  PNGraphic: PNG format
- [Query reply] { BITM | PNG }
- [Example] Call ibwrt (analyzer%, ":HCOP:DEST MMEM")
  Call ibwrt (analyzer%, ":HCOP:MMEM:FILE:NUMB 1234")
  Call ibwrt (analyzer%, ":HCOP:MMEM:FILE:TYPE BITM")
- [Relevant commands] :HCOPy:[:IMMediate]
  :HCOPy:DESTination
  :HCOPy:MMEMory:FILE:NUMBer

5.20 Cross Reference Index of Commands and Function Buttons

This section shows a cross reference table that maps function buttons and associated SCPI commands.

5.20.1 {FREQ} Button

Command	Function	Reference Page
[:SENSe<ch>]		
:FREQuency<screen>		
:CENTer	Sets the center frequency	5-29
:STEP	Sets the center frequency setting resolution in Up/Down operation	5-32
:AUTO	Sets the center frequency setting resolution mode in Up/ Down operation	5-33
:STARt	Sets the start frequency	5-29
:STOP	Sets the stop frequency	5-30
:SPAN	Sets the span frequency	5-30
:FULL	Sets the span to the maximum span	5-31
:PREVious	Returns the span to the setting before span change	5-31
:ZERO	Sets the span to zero span	5-32
:OFFSet	Sets the offset frequency against the center frequency	5-33
:STATe	Sets the offset frequency state against the center frequency	5-34
:CHANnel		
:NUMBer	Sets the channel number	5-35
[:SENSe<ch>]		
:PRESelector<screen>	Manually adjusts the pre-selector filter	5-49
:AUTO	Executes automatic adjustment of the pre-selector filter	5-48

5.20.2 {LEVEL} Button

Command	Function	Reference Page
:DISPlay<ch>		
[:WINDow<screen>]		
:TRACe		
:Y		
[:SCALe]		
:RLEVel	Sets reference level	5-199
:OFFSet	Sets Offset value to the reference level value	5-200
:STATe	Sets the offset value to the reference level ON/OFF	5-201
:PDIVision	Sets 1 division value at log scaling	5-202
:SPACing	Sets vertical scale type	5-202
:UNIT<ch>		
:POWer<screen>	Sets the level unit system	5-304
:INPut<ch>		
:ATTenuation<screen>	Sets the input RF attenuator	5-16
:AUTO	Selects an input RF attenuator setting mode	5-17
:MINimum	Sets the minimum set value of the input RF attenuator	5-17
:STATe	Sets the minimum setting function mode for the input RF attenuator	5-18
:GAIN<screen>		
:STATe	Sets the input gain amp function mode	5-19
[:SENSe<ch>]		
:CORRection		
:CSET		
:STATe	Switches the RF input level correction function ON or OFF	5-60
:DATA	Enters the RF input level correction data	5-60
:DELete	Deletes all the RF input level correction data	5-61

5.20.3 {BW} Button

Command	Function	Reference Page
[:SENSe<ch>]		
:BANDwidth<screen>		
[:RESolution]	Sets the resolution bandwidth (RBW)	5-35
:AUTO	Selects the resolution bandwidth (RBW) setting mode	5-36
:RATio	Sets the setting ratio between the span frequency and the resolution bandwidth (RBW)	5-40
:STATe	Sets the setting ratio mode between the span frequency and the resolution bandwidth (RBW)	5-41
:VIDeo	Selects the video bandwidth (VBW) setting mode	5-36
:AUTO	Sets the video bandwidth (VBW)	5-37
:RATio	Sets the setting ratio between the resolution bandwidth (RBW) and the video bandwidth (VBW)	5-38
:STATe	Sets the setting ratio mode between the resolution bandwidth (RBW) and the video bandwidth (VBW)	5-39
:PLL	Selects the loop filter width in the PLL circuit	5-42
:COUPle<screen>		
:ALL		
:AUTO	Changes the coupling set items to the automatic setting mode	5-43
:ADC<screen>		
:DITHer	Sets the ADC Dither function	5-43

5.20.4 {SWP} Button

Command	Function	Reference Page
[:SENSe<ch>]		
:SWEep<screen>		
:TIME	Sets the sweep time	5-49
:AUTO	Selects the sweep time setting mode	5-50
:TRIGger<ch>		
[:SEQuence<screen>]		
:SOURce	Sets the trigger	5-180
:SLOPe	Sets the trigger polarity of each trigger source	5-181
:LEVel		
:VIDeo	Sets the trigger level for Video trigger	5-181
:EXTernal	Sets the trigger level when EXT2 (EXT2 Input Connector) is triggered	5-182
:IF	Sets the trigger level for IF trigger	5-182
:DELay	Sets a trigger delay value	5-183
[:SENSe<ch>]		
:SWEep		
:GATE	Sets the gated sweep to ON or OFF	5-51
:DELay	Sets the gate signal position	5-51
:WIDTh	Sets the gate signal width	5-52
:AUTO	Switches the gate signal mode	5-52
:SOURce	Sets the gated sweep trigger	5-53
:SLOPe	Sets the trigger polarity of each trigger source	5-53
:LEVel		
:EXTernal	Sets the trigger level when using an EXT2 (external input terminal 2) trigger	5-54
:IF	Sets the trigger level for IF trigger	5-54

5.20.5 {TRACE} Button

Command	Function	Reference Page
:DISPlay<ch>		
[:WINDow:<screen>]		
:TRACe		
[:NUMBer{1|2|3|4}]		
:ACTive	Selects the active trace	5-190
:MODE	Sets the display mode of the specified trace	5-192
:NCORrection		
:STATe	Sets the trace normalize function ON/OFF	5-193
: STORe	Stores the reference waveform data to be used for the trace normalize function	5-194
[:NUMBer{1|2}]		
:STORe	Stores the waveform data of trace 1 or 2	5-195
:AANalog		
:STATe	Sets the quasi analog display mode ON/OFF	5-195
[:SENSe<ch>]		
:DETector<screen>		
:TRACe		
[:NUMBer<trace>]		
:FUNCtion	Selects the trace detector	5-44
:AUTO	Selects the trace detector determination mode	5-45
:AVERage<screen>		
:TYPE	Selects the average detection mode of the average detector	5-46
:AUTO	Selects the mode to be used when selecting the average detection mode of the average detector	5-47
:AANalog		
:SAMPle		
:COUNt	Sets the quasi analog function and sampling count	5-61

{TRACE} Button (Cont'd)

Command	Function	Reference Page
:CALCulate<ch>		
:CURSor<screen>		
:ANCHor	Sets ON or OFF the anchor function of the X and Y cursors	5-301
:X	Sets the display position of the X cursor when X and Y cursors are displayed	5-302
:Y	Sets the display position of the Y cursor when X and Y cursors are displayed	5-302
:STATe	Sets ON or OFF the display of the X and Y cursors	5-303
[:SENSe<ch>]		
:SWEep<screen>		
:COUNt	Sets the sweep averaging count and MAX/MIN HOLD count	5-61

5.20.6 {MKR}/{MKR→} Button

Command	Function	Reference Page
:CALCulate<ch>		
:MARKer<screen>		
[:NUMBer<mkr>]		
:ACTive	Specifies an operation target marker (active marker) among the multimarkers	5-225
:FUNCtion		
[:STATe]	Sets ON or OFF marker functions	5-226
:NUMBer<mkr>		
[:STATe]	Sets ON or OFF the specified multimarker	5-226
[:NUMBer<mkr>]		
:X	Specifies a frequency position and a time position of the specified multimarker	5-227
:Y?	Reads the level value of the specified multimarker	5-229
:MAXimum		
[:PEAK]	Searches the maximum peak point using the specified multimarker	5-230
:NEXT	Searches the next peak using the specified multimarker	5-231
:LEFT	Searches the next peak in the left direction using the specified multimarker	5-232
:RIGHt	Searches the next peak in the right direction using the specified multimarker	5-234
:MINimum		
[:PEAK]	Searches the minimum peak using the specified multimarker	5-236
:NEXT	Searches the next minimum peak using the specified multimarker	5-237
:TRACe	Moves the specified marker to the specified trace	5-238
:RESet	Sets all the markers except marker No. 1 off	5-239
:LIST		
[:STATe]	Displays the marker list of the markers displayed	5-239
:STEP	Sets a marker step size	5-242
:AUTO	Sets a marker step size mode	5-242
:CALCulate<ch>		
:MARKer<screen>		
[:NUMBer<mkr>]		
:STRack		
[:STATe]	Sets ON or OFF the Signal Tracking function	5-241

{MKR}/{MKR→} Button (Cont'd)

Command	Function	Reference Page
:CALCulate<ch> :MARKer<screen> [:NUMBer<mkr>] :SET :CENTer :STEP	Sets the marker frequency as the center frequency step size	5-259
:MARKer :STEP	Sets the marker frequency as the marker frequency step size	5-260
:MAXimum :SET :CENTer	Sets the marker frequency as the center frequency after peak search is performed	5-260
:RLEVel	Sets the marker level value as the reference level after peak search is performed	5-261
:ROBJect	Specifies a reference for the relative value display of the marker	5-266
:CALCulate<ch> :DELTamarker<screen> [:NUMBer<mkr>] :SET :CENTer	Sets the delta marker frequency as the center frequency	5-261
:STEP	Sets the delta marker frequency as the center frequency step size	5-262
:SPAN	Sets the delta marker frequency as the span frequency	5-262
:MARKer :STEP	Sets the delta marker frequency as the marker step size	5-263
[:STATe]	Sets ON or OFF the display of the delta marker	5-263
:FIXed [:STATe]	Sets ON or OFF the display of the fixed delta marker	5-264
:MAXimum [:PEAK]	Searches the peak and sets a fixed marker	5-264
:INVerse [:STATe]	Sets ON or OFF the display of the (1/delta) marker	5-265
:CALCulate<ch> :MARKer<screen> [:NUMBer<mkr>] :SET :CENTer	Sets the marker frequency as the center frequency	5-258
:RLEVel	Sets the marker level value as the reference level	5-259

5.20.7 {SEARCH} Button

Command	Function	Reference Page
:CALCulate<ch>		
:MARKer<screen>		
:MAXimum		
:LIST	Searches peak points and displays a marker list	5-240
:CONTinuous	Sets ON or OFF the continuous peak point search mode	5-240
:DELTa	Specifies a deviation for peak point judgment at the time of peak point search	5-241
:SEARch		
:X		
:MODE	Sets a peak search range specification mode on the horizontal axis	5-243
:POSition	Specifies the reference position of the peak search range on the horizontal axis	5-244
:WIDTh	Specifies a search width from the reference position of the peak search range on the horizontal axis	5-245
:COUPling	Sets a move mode of the peak search range on the horizontal axis	5-246
:Y		
:MODE	Sets a peak search range specification mode on the vertical axis	5-247
:DLINe	Specifies the peak search range with Display Line used as the reference	5-248
:LUPPer	Specifies the peak search range with Limit Line1 used as the reference	5-249
:LLOWer	Specifies the peak search target range with Limit Line2 used as the reference	5-250
:MINNer<area>	Sets ON or OFF the display of the marker frame for the Multi Inner Peak Search function	5-251
:X		
:POSition	Specifies a position on the horizontal axis in the marker frame by the Multi Inner Peak Search function	5-254
:WIDTh	Specifies the width of the marker frame on the horizontal axis by the Multi Inner Peak Search function	5-255
:Y	Sets the specified marker frame and the vertical axis search range mode of the Multi Inner Peak Search function	5-256
:LOWer	Specifies the lower position of the vertical axis of the marker frame by the Multi Inner Peak Search function	5-257
:UPPer	Specifies the upper position of the vertical axis of the marker frame by the Multi Inner Peak Search function	5-258

5.20.8 {MEAS} Button

Command	Function	Reference Page
:CALCulate<ch>		
:MARKer<screen>		
[:NUMBer<mkr>]		
:FCOunt		
[:STATe]	Sets ON or OFF the frequency counter function	5-267
:FREQuency?	Reads the results of measurement by the frequency counter function	5-268
[:SENSe]		
:FCOunt<screen>		
:AVERage		
:COUNt	Sets the averaging count in count operation in the frequency counter function	5-121
[:STATe]	Sets the averaging function in count operation to ON or OFF in the frequency counter function	5-122
:CALCulate<ch>		
:MARKer<screen>		
[:NUMBer<mkr>]		
:FUNCtion		
:NOISe?	Reads the results of measurement by the Noise/Hz function	5-279
:NOISe		
:BWIDth	Sets a bandwidth for noise measurement in Noise/Hz function	5-277
:STATe	Sets ON or OFF the Noise/Hz function	5-277
:MODE	Selects an operation mode for Noise/Hz function	5-278
:CALCulate<ch>		
:MARKer<screen>		
[:NUMBer<mkr>]		
:FUNCtion		
:XDBDown	Executes the X dB Down function	5-269
:LEFT	Executes the X dB Down LEFT function	5-270
:RIGHt	Executes the X dB Down RIGHT function	5-271
:LEVel	Sets the Down width of the X dB Down function	5-272
:PEAK	Executes the X dB Down function after peak search	5-273
:MODE	Selects a display mode after execution of the X dB Down function	5-274
:CONTinuous		
[:STATe]	Sets ON or OFF the continuous X dB Down function	5-275
:RMARker		
[:STATe]	Sets ON or OFF the display of the reference marker when the X dB Down function is executed	5-276

{MEAS} Button (Cont'd)

<table>
<tr><th>Command</th><th>Function</th><th>Reference Page</th></tr>
<tr><td>:CALCulate<ch>
:MARKer<screen>
[:NUMBer<mkr>]
:FUNCtion
:AM?</td><td>Reads the results of %AM measurement</td><td>5-279</td></tr>
<tr><td>:STATe</td><td>Sets ON or OFF %AM measurement</td><td>5-280</td></tr>
<tr><td>[:SENSe<ch>]
:HARMonics
:FFRequency</td><td>Sets the reference signal frequency in harmonic measurement</td><td>5-110</td></tr>
<tr><td>:STATe</td><td>Sets the reference signal frequency mode in harmonic measurement</td><td>5-110</td></tr>
<tr><td>:NUMBer</td><td>Sets the harmonics order to be used in harmonic measurement</td><td>5-111</td></tr>
<tr><td>:CONFigure<ch>
:HARMonics</td><td>Sets the harmonic measurement mode</td><td>5-136</td></tr>
<tr><td>:NORMal</td><td>Terminates each measurement mode</td><td>5-138</td></tr>
<tr><td>:MEASure<ch>
:HARMonics?
:FUNDamental?</td><td>Harmonic measurement execution and the fundamental wave measurement results reading</td><td>5-167</td></tr>
<tr><td>:NUMBer{2|3 4|5 6|7|8|9|10}?</td><td>Harmonic measurement execution and all measurement results reading</td><td>5-166</td></tr>
<tr><td>[:SENSe<ch>]
:IM
:LIM
[:STATe]</td><td>Sets ON or OFF the Pass/Fail judgment for distorted signals in IM measurement</td><td>5-107</td></tr>
<tr><td>:ORDer</td><td>Sets the maximum order of the signal to be measured in IM measurement</td><td>5-106</td></tr>
<tr><td>:THReshold{3|5|7|9}</td><td>Sets the Pass/Fail judgment value for distorted signal in IM measurement</td><td>5-107</td></tr>
<tr><td>:DATA
:MODE</td><td>Specifies the measurement parameter setting mode for IM measurement</td><td>5-108</td></tr>
<tr><td>:SAVE</td><td>Stores the measurement parameters for IM measurement</td><td>5-109</td></tr>
<tr><td>:CONFigure<ch>
:IM</td><td>Sets the IM measurement mode</td><td>5-136</td></tr>
<tr><td>:NORMal</td><td>Terminates each measurement mode</td><td>5-138</td></tr>
</table>

{MEAS} Button (Cont'd)

Command	Function	Reference Page
:MEASure<ch>		
:IM		
[:NUMBer{1|3|5|7|9}]?	Executes IM measurement and reads the results	5-159
:REFerence?	Executes IM measurement and reads the reference frequency data	5-161
:DELTa?	Executes IM measurement and reads the frequency difference between two signals	5-161
:UPPer		
[:NUMBer{1|3|5|7|9}]?	Executes IM measurement and reads the results of specified-order modulation distortion	5-164
:LOWer		
[:NUMBer{1|3|5|7|9}]?	Executes IM measurement and reads the results of specified-order modulation distortion	5-165
:IP3?	IM measurement execution and 3rd order intercept point value reading	5-162
:IPOint		
[:NUMBer{3|5|7|9}]?	IM measurement execution and intercept point value reading	5-163

5.20.9 {POWER} Button

Command	Function	Reference Page
[:SENSe<ch>]		
:CPOWer<screen>		
:AVERage		
:COUNt	Sets the averaging count in channel power measurement	5-62
[:STATe]	Sets the averaging operation mode to ON or OFF in channel power measurement	5-63
:MODE	Specifies the operation type in averaging operation mode in channel power measurement	5-64
:WINDow	Sets the measurement window display to ON or OFF in channel power measurement	5-65
:POSition	Specifies the measurement window display position in channel power measurement	5-66
:WIDTh	Specifies the measurement window display width in channel power measurement	5-67
:DATA		
:MODE	Specifies the measurement parameter setting mode in channel power measurement	5-68
:SAVE	Saves the measurement parameters in channel power measurement	5-69
:CONFigure<ch>		
:CPOWer<screen>	Sets the channel power measurement mode	5-132
:NORMal	Terminates each measurement mode	5-138
:MEASure<ch>		
:CPOWer<screen>?	Channel Power measurement execution and measurement result (Trace) reading	5-143
:PDENsity?	Channel Power measurement execution and average power density (Trace) reading	5-143
:RMS?	Channel Power measurement execution and measurement result (RMS) reading	5-144
:PDENsity?	Channel Power measurement execution and average power density (RMS) reading	5-144

{POWER} Button (Cont'd)

Command	Function	Reference Page
[:SENSe<ch>]		
:APOWer<screen>		
:AVERage		
:COUNt	Sets the averaging count in average power measurement	5-70
[:STATe]	Sets the averaging operation mode to ON or OFF in average power measurement	5-70
:MODE	Specifies the operation type in averaging operation mode in average power measurement	5-71
:WINDow	Sets the measurement window display to ON or OFF in average power measurement	5-72
:POSition	Specifies the measurement window display position in average power measurement	5-73
:WIDTh	Specifies the measurement window display width in average power measurement	5-74
:DATA		
:MODE	Specifies the measurement parameter setting mode in average power measurement	5-75
:SAVE	Saves the measurement parameters in average power measurement	5-76
:CONFigure<ch>		
:APOWer<screen>	Sets the average power measurement mode	5-133
:NORMal	Terminates each measurement mode	5-138
:MEASure<ch>		
:APOWer<screen>?	Average Power measurement execution and measurement result (Trace) reading	5-145
:PDENsity?	Average Power measurement execution and average power density (Trace) reading	5-145
:RMS?	Average Power measurement execution and measurement result (RMS) reading	5-146
:PDENsity?	Average Power measurement execution and average power density (RMS) reading	5-146

{POWER} Button (Cont'd)

Command	Function	Reference Page
[:SENSe<ch>]		
:OBW<screen>		
:AVERage		
:COUNt	Sets the averaging count for OBW measurement	5-77
[:STATe]	Sets the averaging operation mode to ON or OFF in OBW measurement	5-77
:MODE	Specifies the operation type in averaging operation mode in OBW measurement	5-78
:PERCent	Specifies the OBW% value in OBW measurement	5-78
:DATA		
:MODE	Specifies the measurement parameter setting mode in OBW measurement	5-79
:SAVE	Saves the measurement parameters in OBW measurement	5-80
:CONFigure<ch>		
:OBW<screen>	Sets the OBW measurement mode	5-133
:NORMal	Terminates each measurement mode	5-138
:MEASure<ch>		
:OBW<screen>?	Executes OBW measurement and reads all the measurement results	5-147
:OBW?	Executes OBW measurement and reads the measurement result(only the OBW value)	5-147
:FCENter?	Executes OBW measurement and reads the measurement results(only the OBW center frequency value)	5-148

{POWER} Button (Cont'd)

Command	Function	Reference Page
[:SENSe<ch>]		
:ACP		
:AVERage		
:COUNt	Sets the averaging count in ACP measurement	5-77
[:STATe]	Sets the averaging operation mode to ON or OFF in ACP measurement	5-77
:MODE	Specifies the operation type in averaging operation mode in ACP measurement	5-78
:DATA		
:MODE	Specifies the measurement parameter setting mode in ACP measurement	5-79
:SAVE	Saves the measurement parameters in ACP measurement	5-80
:CBWidth	Sets the carrier bandwidth that becomes the target of the reference power operation in ACP measurement	5-85
:CSBW		
:DATA	Sets the adjacent channel position and adjacent channel band in ACP measurement	5-85
:DELete	Initializes the adjacent channel position and adjacent channel band data in ACP measurement	5-86
:RNYQuist	Sets the Root Nyquist filter operation mode to ON or OFF in ACP measurement	5-87
:SRATe	Sets the Symbol Rate for Root Nyquist filter operation in ACP measurement	5-88
:RFACtor	Sets the Rolloff Factor for Root Nyquist filter operation in ACP measurement	5-88
:NCORrection		
[:STATe]	Sets the noise correction function to ON or OFF in ACP measurement	5-89
:POWer		
:LEVel		
:AUTO	Executes the Auto Level Set function in ACP measurement	5-89
:CONFigure<ch>		
:ACP	Sets the ACP measurement mode	5-134
:NORMal	Terminates each measurement mode	5-138

{POWER} Button (Cont'd)

Command	Function	Reference Page
:MEASure<ch>		
:ACP		
[:NUMBer{1|2|3|4|5}]?	Executes ACP measurement and reads all the measurement results	5-149
:RPOWer?	Executes ACP measurement and reads the results of reference power measurement	5-150
:UPPer		
[:NUMBer{1|2|3|4|5}]?	Executes ACP measurement and reads all the measurement results of the specified channels on the upper side	5-151
:LOWer		
[:NUMBer{1|2|3|4|5}]?	Executes ACP measurement and reads all the measurement results of the specified channels on the lower side	5-152
[:SENSe<ch>]		
:MCACp		
:RNYQuist	Sets Root Nyquist filter operation mode to ON or OFF in Multi Carrier ACP measurement	5-90
:SRATe	Sets the Symbol Rate for Root Nyquist filter operation in Multi Carrier ACP measurement	5-91
:RFACtor	Sets the Rolloff Factor for Root Nyquist filter operation in Multi Carrier ACP measurement	5-91
:AVERage		
:COUNt	Sets the averaging count in Multi Carrier ACP measurement	5-92
[:STATe]	Sets the averaging operation mode to ON or OFF in Multi Carrier ACP measurement	5-92
:MODE	Specifies the operation type in averaging operation mode in Multi Carrier ACP measurement	5-93
:DATA		
:MODE	Specifies the measurement parameter setting mode in Multi Carrier ACP measurement	5-94
:SAVE	Saves the measurement parameters in Multi Carrier ACP measurement	5-95

{POWER} Button (Cont'd)

Command	Function	Reference Page
[:SENSe<ch>] :MCACp :PARameter{1|2|...|16} :STATe	Sets the measurement carrier and adjacent channel to ON or OFF in Multi Carrier ACP measurement	5-96
:FREQuency	Sets the offset frequency of the measurement carrier and adjacent channel in Multi Carrier ACP measurement	5-97
:BWIDth	Sets the channel bandwidth of the measurement carrier and adjacent channel in Multi Carrier ACP measurement	5-98
:REFerence	Sets the reference power area of the measurement carrier and adjacent channel in Multi Carrier ACP measurement	5-99
:LIMit	Sets the limit value for checking pass/fail of the measurement carrier and adjacent channel measurement result in Multi Carrier ACP measurement	5-100
:NCORrection [:STATe]	Sets the noise correction function to ON or OFF in Multi Carrier ACP measurement	5-101
:POWer :LEVel :AUTO	Executes the Auto Level Set function in Multi Carrier ACP measurement	5-101
:CARRier :ADJust	Sets the Carrier Freq Adjustment value in Multi Carrier ACP measurement	5-102
:STATe	Sets the Carrier Freq Adjustment function to ON or OFF in Multi Carrier ACP measurement	5-102
:CONFigure<ch> :MCACp	Sets the Multi Carrier ACP measurement mode	5-134
:NORMal	Terminates each measurement mode	5-138
:MEASure<ch> :MCACp [:NUMBer{1|2|3|4|5|6}]?	Executes Multi Carrier ACP measurement and reads all the measurement results	5-153
:CPOWer :[NUMBer{1|2|...|10}]?	Executes Multi Carrier ACP measurement and reads the specified carrier power value	5-154

{POWER} Button (Cont'd)

Command	Function	Reference Page
[:SENSe<ch>]		
:SPURious		
:DATA		
[:NUMBer{1|2|3}]	Registers the sweep parameters to be used to the Spurious table in spurious measurement	5-103
:DELete	Clears all data registered in the Spurious table used in spurious measurement	5-104
:ACTive	Selects a table for spurious measurement	5-104
:DATA		
:MODE	Selects the mode in which the Spurious table is used by the setting parameters for spurious measurement	5-105
:SAVE	Stores the setting parameters to be used for spurious measurement	5-106
:CONFigure<ch>		
:SPURious	Sets the spurious measurement mode	5-135
:NORMal	Terminates each measurement mode	5-138
:MEASure<ch>		
:SPURious?		
[:NUMBer{1|2|3|...|15}]	Executes spurious measurement and reads the results of measurement in the specified measurement area	5-155
[:SENSe<ch>]		
:SEMask		
:CBWidth	Sets the reference power operation width in Spectrum Emission Mask measurement	5-112
:RNYQuist	Sets the Root Nyquist filter operation mode to ON or OFF in Spectrum Emission Mask measurement	5-113
:SRATe	Sets the Symbol Rate for Root Nyquist filter operation in Spectrum Emission Mask measurement	5-114
:RFACtor	Sets the Rolloff Factor for Root Nyquist filter operation in Spectrum Emission Mask measurement	5-114
:DATA	Sets the parameter table for measurement in Spectrum Emission Mask measurement	5-115
:DELete	Clears all data in the parameter table for measurement in Spectrum Emission Mask measurement	5-116
:RPOWer		
:MODE	Sets the reference power calculation mode in Spectrum Emission Mask measurement	5-116

{POWER} Button (Cont'd)

Command	Function	Reference Page
[:SENSe<ch>]		
:SEMask		
:AVERage		
:COUNt	Sets the averaging count during averaging measurement in Spectrum Emission Mask measurement	5-117
[:STATe]	Sets the averaging measurement function to ON or OFF in Spectrum Emission Mask measurement	5-117
:MODE	Specifies the type of operation to be used in the averaging operation mode in Spectrum Emission Mask measurement	5-118
:DATA		
:MODE	Selects the setting parameters used for Spectrum Emission Mask measurement	5-119
:SAVE	Saves the setting parameters used in Spectrum Emission Mask measurement	5-120
:POWer		
:LEVel		
:AUTO	Executes the Auto Level Set function in Spectrum Emission Mask measurement	5-121
:CONFigure<ch>		
:SEMask	Sets the Spectrum Emission Mask measurement mode	5-135
:NORMal	Terminates each measurement mode	5-138
:MEASure<ch>		
:SEMask		
[:NUMBer{1|2|3|4|5}]?	Executes Spectrum Emission Mask measurement and reads the results	5-157
:RPOWer?	Spectrum Emission Mask measurement execution and measurement result reading	5-158
:FAIL?	Spectrum Emission Mask measurement execution and overall Pass/Fail result reading	5-159
[:SENSe<ch>]		
:CCDF		
:BANDwidth		
[:RESolution]	Sets the resolution bandwidth (RBW) in the CCDF measurement	5-122
:POINt	Sets the number of measurement samples in the CCDF measurement	5-123
:GATE	Sets the gate function in the CCDF measurement to ON or OFF	5-123
:THReshold	Sets the threshold level of the gate function in the CCDF measurement	5-124

{POWER} Button (Cont'd)

Command	Function	Reference Page
:DISPlay<ch>		
[:WINDow]		
:TRACe		
:CCDF		
:STATe	Sets the reference waveform display in the CCDF measurement to ON or OFF	5-203
:GAUSsian		
:STATe	Sets the ideal gaussian noise in the CCDF measurement to ON or OFF	5-203
:X		
[:SCALe]		
:CCDF	Sets the maximum horizontal axis value for the waveform display in the CCDF measurement	5-204
:CONFigure<ch>		
:CCDF	Sets the CCDF measurement mode	5-137
:NORMal	Terminates each measurement mode	5-138
:MEASure		
:CCDF		
[:NUMBer{1|2|3|4|5|6}]?	CCDF measurement execution and the measurement result reading	5-168
:PFACtor?	CCDF measurement execution and Peak Factor reading	5-169
:APOWer?	CCDF measurement execution and Average Power reading	5-169
:PRATio		
[:NUMBer{1|2|3|4|5|6}]?	CCDF measurement execution and the power ratio reading	5-170

{POWER} Button (Cont'd)

Command	Function	Reference Page
[:SENSe<ch>]		
:MAPower		
:WINDow		
[:NUMBer<win>]	Sets the window display to ON or OFF in the Multi-Average Power measurement	5-124
:POSition	Specifies the window display position in the Multi-Average Power measurement	5-125
:WIDTh	Specifies the window display width in the Multi-Average Power measurement	5-125
:ACTive	Specifies the active window in the Multi-Average Power measurement	5-126
:RESet	Sets all windows excluding window No.1 to OFF	5-126
:COUPling	Sets the window display, which is coupled with the Average Power in the Multi-Average Power measurement, to ON or OFF	5-127
:PRATio	Sets the Power Ratio measurement to ON or OFF in the Multi-Average Power measurement	5-127
:AVERage		
:COUNt	Sets the number of times averaging is performed in the Multi-Average Power measurement	5-128
[:STATe]	Sets the averaging calculation mode to ON or OFF in the Multi-Average Power measurement	5-128
:MODE	Specifies the calculation type of the averaging calculation mode in the Multi-Average Power measurement	5-129
:DATA		
:MODE	Specifies the measurement parameter setting mode in the Multi-Average Power measurement	5-130
:SAVE	Saves the measurement parameters in the Multi-Average Power measurement	5-131

{POWER} Button (Cont'd)

Command	Function	Reference Page
:CONFigure<ch>		
:MAPower	Sets the Multi-Average Power measurement mode	5-137
:NORMal	Terminates each measurement mode	5-138
:MEASure<ch>		
:MAPower		
[:NUMBer{1|2...|9|10}]?	Multi-Average Power measurement execution and the measurement result (Trace) reading	5-171
:PDENsity		
[:NUMBer{1|2...|9|10}]?	Multi-Average Power measurement execution and the average power density (Trace) reading	5-172
:PRATio		
[:NUMBer{1|2...|9|10}]?	Multi-Average Power measurement execution and Power Ratio (Trace) reading	5-173
:RMS		
[:NUMBer{1|2...|9|10}]?	Multi-Average Power measurement execution and the measurement result (RMS) reading	5-174
:PDENsity		
[:NUMBer{1|2...|9|10}]?	Multi Average Power measurement execution and the average power density (RMS) reading	5-175
:PRATio		
[:NUMBer{1|2...|9|10}]?	Multi-Average Power measurement execution and Power Ratio (RMS) reading	5-176

5.20.10 {PASS FAIL} Button

Command	Function	Reference Page
:CALCulate<ch>		
:LIMit<screen>		
:AUTO	Automatically adjusts the level position of the limit line	5-282
:CONTrol		
:X		
:DOMain	Selects the domain of the limit line to be used	5-283
:MODE	Selects horizontal axis data attributes of the limit line to be used	5-285
:REFerence	Specifies the reference position when the horizontal axis relative value attributes of the limit line to be used are selected	5-286
:USER	Specifies the user-defined reference position when the horizontal axis relative value attributes of the limit line to be used are selected	5-287
:OFFSet	Specifies the offset value when the horizontal axis relative value attributes of the limit line to be used are selected	5-288
:Y		
:MODE	Selects vertical axis data attributes of the limit line to be used	5-289
:REFerence	Specifies the reference position when the vertical axis relative value attributes of the limit line to be used are selected	5-290
:USER	Specifies the user-defined reference position when the vertical axis relative value attributes of the limit line to be used are selected	5-291
:OFFSet	Specifies the offset value when the vertical axis relative value attributes of the limit line to be used are selected	5-291
:CALCulate<ch>		
:LIMit<screen>		
:FAIL?	Reads the Pass/Fail judgment by the limit line	5-292
{:UPPer | :LOWer}		
:COPY	Copies the data for limit line 1 to limit line 2 or Copies the data for limit line 2 to limit line 1	5-293
:DATA	Inputs the data to limit line 1 or 2	5-294
:DELete	Deletes the data from Limit Line Table 1 or 2	5-295
:PASS	Sets the judgment condition for Pass/Fail judgment by Limit Line Table 1 or 2	5-296
:STATe	Sets ON or OFF the display of limit line 1 or 2	5-297

5.20.11 {DISPLAY} Button

Command	Function	Reference Page
:CALCulate<ch>		
:DLINe<screen>	Sets the display position of the display line	5-280
:STATe	Sets ON or OFF the display of the display line	5-281
:RLINe<screen>	Sets the display position of the reference line	5-281
:STATe	Sets ON or OFF the display of the reference line	5-282
:CALCulate<ch>		
:WINDow<screen>		
:POSition	Sets the position of the Measuring Window	5-299
:WIDTh	Sets the width of the Measuring Window	5-300
:STATe	Sets ON or OFF the display of the Measuring Window	5-301
[:SENSe<ch>]		
:SWEep<screen>		
:WINDow	Sets the window sweep to ON or OFF	5-50
:CALCulate<ch>		
:CURSor<screen>		
:ANCHor	Sets ON or OFF the anchor function of the X and Y cursors	5-301
:X	Sets the display position of the X cursor when X and Y cursors are displayed	5-302
:Y	Sets the display position of the Y cursor when X and Y cursors are displayed	5-302
:STATe	Sets ON or OFF the display of the X and Y cursors	5-303

{DISPLAY} Button (Cont'd)

Command	Function	Reference Page
:DISPlay<ch> [:WINDow<screen>] :ACTive	Specifies active screen when waveform zoom function is on	5-191
:DISPlay<ch> [:WINDow] :TRACe :SPLit	Sets two screen display mode ON/OFF	5-194
:X [:SCALe] :ZOOM :MODE	Selects waveform zoom function, and release of zoom function	5-196
:FREQuency :CENTer	Specifies zoom frequency when the waveform zoom function is on	5-197
:SPAN	Specifies zoom width when the waveform zoom function is on	5-198
:TIME :DELay	Specifies zoom time position when the waveform zoom function is on	5-198
:WIDTh	Specifies zoom time width when the waveform zoom function is on	5-199
:DISPlay :ANNotation :CURSor	Sets the display of the XY cursor information data to ON or OFF	5-186
:DLINe	Sets ON/OFF of Display Line data display	5-187
:RLINe	Sets ON/OFF of Reference Line data display	5-188
:WINDow	Sets ON/OFF of Measuring Window data display	5-189

5.20.12 Menu Bar Related Functions

Command	Function	Reference Page
[:SENSe]		
:ROSCillator		
:SOURce		
:FREQuency	Sets the frequency of the external frequency reference	5-55
:AUTO	Switches the frequency reference standard (internal/external)	5-56
:ADJust		
:COARse	Coarse adjustment of the correction value for adjusting the internal 10 MHz frequency reference	5-57
:FINE	Fine adjustment of the correction value for adjusting the internal 10 MHz frequency reference	5-58
:SAVE	Saves the correction value for adjusting the internal 10 MHz frequency reference	5-59
:DEFault	Clears the correction value for adjusting the internal 10 MHz frequency reference	5-59

Command	Function	Reference Page
:INITiate<screen>		
:CONTinuous	Sets continuous sweep mode ON/OFF	5-177
[:IMMediate]	Starts a sweep or measurement	5-178
:RESTart	Resets and restarts a sweep	5-178
:ABORt	Aborts a sweep	5-178

Menu Bar Related Functions (Cont'd)

Command	Function	Reference Page
:DISPlay		
:ANNotation		
:DATE	Sets the date display ON/OFF	5-186
:FORMat	Sets the date display type	5-187
:LOGO	Sets ON/OFF of ADVANTEST logo	5-188
:TITLe	Sets screen title	5-189
:CALibration		
:SANalyzer	Executes calibration using the external CAL signal (including RF ATT)	5-209
:ATTenuation		
:NONE	Executes calibration using the internal CAL signal (excluding RF ATT)	5-210
:MMEMory		
:STORe		
:STATe	Executes the Save function for every setting state of this instrument	5-212
:STANdard		
:STATe	Executes the SAVE function for the Standard information	5-212
:LOAD		
:STATe	Executes the LOAD function for every setting state of this instrument	5-213
:STANdard		
:STATe	Executes the LOAD function for the Standard information	5-213
:SELect		
:ITEM		
:SETup	Sets the Save conditions for the setting parameters	5-214
:TRACe	Sets the Save conditions for the trace data	5-214
:NCORrection	Sets the Save conditions for the Normalize function correction data	5-215
:LIMit	Sets the Save conditions for the limit line data for the limit line function	5-215
:CORRection	Sets the Save condition for the level correction data for the input level correction function	5-216
:SPURious	Sets the Save condition for the setting parameter for the Spurious measurement function	5-216
:SEMask	Sets the Save condition for the setting parameter for the Spectrum Emission Mask Measurement function	5-217

Menu Bar Related Functions (Cont'd)

Command	Function	Reference Page
:SYSTem		
:PRESet	Initializes the parameters of each measurement system	5-305
:ALL	Initializes all measurement systems	5-306
:SELect	Selects a measurement system	5-306
:MODulation	Selects a modulation analysis system	5-307
:STANdard	Selects the Standard mode	5-308
:ERRor?	Inquiry about the error that occurred last	5-309
:ALL?	Inquiry about the details of the error log	5-309
:DIAGnostic		
:PON?	Reads the Power on DIAG result	5-310
:SELFtest?	Executes a self-test and reading of the result	5-311
:HCOPy		
[:IMMediate]	Outputs a copy to a file or a printer	5-316
:DESTination	Specifies the output destination	5-317
:MMEMory		
:FILE		
:NUMBer	Specifies an output file number	5-318
:TYPE	Specifies the output file type	5-319

5.20.13 Functions Dedicated to Remote Control

Command	Function	Reference Page
:TRACe<ch>		
[:DATA<screen>]?	Outputs the trace data	5-205

Command	Function	Reference Page
:FORMat		
:BORDer	Sets the order of trace data output byte	5-206
:TRACe		
[:DATA]	Sets the output format of the trace data	5-207

Command	Function	Reference Page
:STATus		
:OPERation		
:ENABle	Enables the standard operation status enable register	5-313
:EVENt?	Reads the standard operation status event register	5-313
:MEASure		
:ENABle	Enables the measuring status enable register	5-315
:EVENt	Reads the measuring status event register	5-315
:QUEStionable		
:ENABle	Enables the questionable status enable register	5-314
:EVENt	Reads the questionable status event register	5-314

ALPHABETICAL INDEX

[Symbol]

[A]

[B]

[C]

[E]

[F]

[G]

[I]

[L]

[M]

[O]

[P]

[R]

[S]

[T]

IMPORTANT INFORMATION FOR ADVANTEST SOFTWARE

PLEASE READ CAREFULLY: This is an important notice for the software defined herein. Computer programs including any additions, modifications and updates thereof, operation manuals, and related materials provided by Advantest (hereafter referred to as "SOFTWARE"), included in or used with hardware produced by Advantest (hereafter referred to as "PRODUCTS").

SOFTWARE License

All rights in and to the SOFTWARE (including, but not limited to, copyright) shall be and remain vested in Advantest. Advantest hereby grants you a license to use the SOFTWARE only on or with Advantest PRODUCTS.

Restrictions

(1) You may not use the SOFTWARE for any purpose other than for the use of the PRODUCTS.
(2) You may not copy, modify, or change, all or any part of, the SOFTWARE without permission from Advantest.
(3) You may not reverse engineer, de-compile, or disassemble, all or any part of, the SOFTWARE.

Liability

Advantest shall have no liability (1) for any PRODUCT failures, which may arise out of any misuse (misuse is deemed to be use of the SOFTWARE for purposes other than it's intended use) of the SOFTWARE. (2) For any dispute between you and any third party for any reason whatsoever including, but not limited to, infringement of intellectual property rights.

LIMITED WARRANTY

1. Unless otherwise specifically agreed by Seller and Purchaser in writing, Advantest will warrant to the Purchaser that during the Warranty Period this Product (other than consumables included in the Product) will be free from defects in material and workmanship and shall conform to the specifications set forth in this Operation Manual.

2. The warranty period for the Product (the "Warranty Period") will be a period of one year commencing on the delivery date of the Product.

3. If the Product is found to be defective during the Warranty Period, Advantest will, at its option and in its sole and absolute discretion, either (a) repair the defective Product or part or component thereof or (b) replace the defective Product or part or component thereof, in either case at Advantest's sole cost and expense.

4. This limited warranty will not apply to defects or damage to the Product or any part or component thereof resulting from any of the following:

   (a) any modifications, maintenance or repairs other than modifications, maintenance or repairs (i) performed by Advantest or (ii) specifically recommended or authorized by Advantest and performed in accordance with Advantest 's instructions;

   (b) any improper or inadequate handling, carriage or storage of the Product by the Purchaser or any third party (other than Advantest or its agents);

   (c) use of the Product under operating conditions or environments different than those specified in the Operation Manual or recommended by Advantest, including, without limitation, (i) instances where the Product has been subjected to physical stress or electrical voltage exceeding the permissible range and (ii) instances where the corrosion of electrical circuits or other deterioration was accelerated by exposure to corrosive gases or dusty environments;

   (d) use of the Product in connection with software, interfaces, products or parts other than software, interfaces, products or parts supplied or recommended by Advantest;

   (e) incorporation in the Product of any parts or components (i) provided by Purchaser or (ii) provided by a third party at the request or direction of Purchaser or due to specifications or designs supplied by Purchaser (including, without limitation, any degradation in performance of such parts or components);

   (f) Advantest's incorporation or use of any specifications or designs supplied by Purchaser;

   (g) the occurrence of an event of force majeure, including, without limitation, fire, explosion, geological change, storm, flood, earthquake, tidal wave, lightning or act of war; or

   (h) any negligent act or omission of the Purchaser or any third party other than Advantest.

5. **EXCEPT TO THE EXTENT EXPRESSLY PROVIDED HEREIN, ADVANTEST HEREBY EXPRESSLY DISCLAIMS, AND THE PURCHASER HEREBY WAIVES, ALL WARRANTIES, WHETHER EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, (A) ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE AND (B) ANY WARRANTY OR REPRESENTATION AS TO THE VALIDITY, SCOPE, EFFECTIVENESS OR USEFULNESS OF ANY TECHNOLOGY OR ANY INVENTION.**

6. **THE REMEDY SET FORTH HEREIN SHALL BE THE SOLE AND EXCLUSIVE REMEDY OF THE PURCHASER FOR BREACH OF WARRANTY WITH RESPECT TO THE PRODUCT.**

7. **ADVANTEST WILL NOT HAVE ANY LIABILITY TO THE PURCHASER FOR ANY INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR PUNITIVE DAMAGES, INCLUDING, WITHOUT LIMITATION, LOSS OF ANTICIPATED PROFITS OR REVENUES, IN ANY AND ALL CIRCUMSTANCES, EVEN IF ADVANTEST HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES AND WHETHER ARISING OUT OF BREACH OF CONTRACT, WARRANTY, TORT (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE), STRICT LIABILITY, INDEMNITY, CONTRIBUTION OR OTHERWISE. TORT (INCLUDING, WITHOUT LIMITATION, NEGLIGENCE), STRICT LIABILITY, INDEMNITY, CONTRIBUTION OR OTHERWISE.**

8. **OTHER THAN THE REMEDY FOR THE BREACH OF WARRANTY SET FORTH HEREIN, ADVANTEST SHALL NOT BE LIABLE FOR, AND HEREBY DISCLAIMS TO THE FULLEST EXTENT PERMITTED BY LAW ANY LIABILITY FOR, DAMAGES FOR PRODUCT FAILURE OR DEFECT, WHETHER ARISING OUT OF BREACH OF CONTRACT, TORT (INCLUDING, WITHOUT LIMITATION, NEGLEGENCE), STRICT LIABILITY, INDEMNITY, CONTRIBUTION OR OTHERWISE.**

CUSTOMER SERVICE DESCRIPTION

In order to maintain safe and trouble-free operation of the Product and to prevent the incurrence of unnecessary costs and expenses, Advantest recommends a regular preventive maintenance program under its maintenance agreement.

Advantest's maintenance agreement provides the Purchaser on-site and off-site maintenance, parts, maintenance machinery, regular inspections, and telephone support and will last a maximum of ten years from the date the delivery of the Product. For specific details of the services provided under the maintenance agreement, please contact the nearest Advantest office listed at the end of this Operation Manual or Advantest 's sales representatives.

Some of the components and parts of this Product have a limited operating life (such as, electrical and mechanical parts, fan motors, unit power supply, etc.). Accordingly, these components and parts will have to be replaced on a periodic basis. If the operating life of a component or part has expired and such component or part has not been replaced, there is a possibility that the Product will not perform properly. Additionally, if the operating life of a component or part has expired and continued use of such component or part damages the Product, the Product may not be repairable. Please contact the nearest Advantest office listed at the end of this Operation Manual or Advantest's sales representatives to determine the operating life of a specific component or part, as the operating life may vary depending on various factors such as operating condition and usage environment.

SALES & SUPPORT OFFICES

Advantest Korea Co., Ltd.
22BF, Kyobo KangNam Tower,
1303-22, Seocho-Dong, Seocho-Ku, Seoul #137-070, Korea
Phone: +82-2-532-7071
Fax: +82-2-532-7132

Advantest (Suzhou) Co., Ltd.
Shanghai Branch Office:
Bldg. 6D, NO.1188 Gumei Road, Shanghai, China 201102 P.R.C.
Phone: +86-21-6485-2725
Fax: +86-21-6485-2726

Shanghai Branch Office:
406/F, Ying Building, Quantum Plaza, No. 23 Zhi Chun Road,
Hai Dian District, Beijing,
China 100083
Phone: +86-10-8235-3377
Fax: +86-10-8235-6717

Advantest (Singapore) Pte. Ltd.
438A Alexandra Road, #08-03/06
Alexandra Technopark Singapore 119967
Phone: +65-6274-3100
Fax: +65-6274-4055

Advantest America, Inc.
3201 Scott Boulevard, Suite, Santa Clara, CA 95054, U.S.A
Phone: +1-408-988-7700
Fax: +1-408-987-0691

ROHDE & SCHWARZ Europe GmbH
Mühldorfstraße 15 D-81671 München, Germany
(P.O.B. 80 14 60 D-81614 München, Germany)
Phone: +49-89-4129-13711
Fax: +49-89-4129-13723

ADVANTEST®

http://www.advantest.co.jp

ADVANTEST CORPORATION
Shin-Marunouchi Center Building, 1-6-2 Marunouchi, Chiyoda-ku, Tokyo 100-0005, Japan
Phone: +81-3-3214-7500