MXCW

Command Strings

MXCWAPT command strings for control systems, such as Crestron or Extron.
Version: 8.1 (2021-K)

Table of Contents

MXCW
Command Strings

MXCWAPT Microflex® Complete Wireless Command Strings

The device is connected via Ethernet to a control system, such as AMX, Crestron or Extron.

Connection: Ethernet (TCP/IP; select “Client” in the AMX/Crestron program)
Port: 2202

Conventions

There are 4 types of strings:

GET	Finds the status of a parameter. After the AMX/Crestron sends a GET command, the device responds with a REPORT string
SET	Changes the status of a parameter. After the AMX/Crestron sends a SET command, the device will respond with a REPORT string to indicate the new value of the parameter.
REP	When the device receives a GET or SET command, it will reply with a REPORT command to indicate the status of the parameter. REPORT is also sent by the device when a parameter is changed on the MXCWAPT, through the GUI, or a conference unit.
SAMPLE	Used for metering audio levels.

All messages sent and received are ASCII. Note that the level indicators and gain indicators are also in ASCII.

Most parameters will send a REPORT command when they change. Thus, it is not necessary to constantly query parameters. The device will send a REPORT command when any of these parameters change.

Indexing

Indexing is used to specifically identify upon what the command string is acting.

0	All Channels / All Seat Numbers
1 through 1	Aux Input
1 through 1	Aux Output
1 through 10	Dante Input
1 through 10	Dante Output
1 through 65535	Seat Number
1 through 50	Voting Configuration
1 through 5	Voting Button

Command Strings

MIC_STATUS

Description	Retrieve and control microphone status
Supported Commands	GET, SET, and REP
Indexing	Seat Number **Note:** There are 65535 valid seat numbers, but only some may refer to online registered devices. Indexing seat numbers not referring to online devices may result in no response or in a return value of "UNKNOWN." Using a value of 0 is only applicable to the GET command. The GET command with an index of 0 will GET ALL seat numbers of online registered devices. No response is given for seat numbers not referring to online devices.
Values	OFF ON
Examples	If device is available: < GET 2 MIC_STATUS > < REP 2 MIC_STATUS OFF > < SET 5 MIC_STATUS ON > < REP 5 MIC_STATUS ON > If mic is enabled via another source other than TPCI: < REP 1 MIC_STATUS ON > If device is registered, but unavailable: < GET 2 MIC_STATUS > < REP 2 MIC_STATUS UNKNOWN > < SET 5 MIC_STATUS ON > < REP 5 MIC_STATUS UNKNOWN >

SPEAK_REQUEST

Description	Control speaker requests
Supported Commands	SET
Indexing	Seat Number

	Note: There are 65535 valid seat numbers, but only some may refer to online registered devices. Indexing seat numbers not referring to online devices may result in no response or in a return value of "UNKNOWN." Using a value of 0 is NOT accepted.
Values	TRUE
Examples	If operation mode is automatic and speaker list is NOT full: < SET 2 SPEAK_REQUEST TRUE > < REP 2 SPEAK_LIST_STATUS IN_LIST > If operation mode is manual and request list is NOT full: < SET 2 SPEAK_REQUEST TRUE > < REP 2 REQUEST_LIST_STATUS IN_LIST > If device is registered, but unavailable: < SET 2 SPEAK_REQUEST TRUE > < REP 2 SPEAK_REQUEST UNKNOWN >

SPEAK_RELEASE

Description	Release speakers in list
Supported Commands	SET
Indexing	Seat Number **Note:** There are 65535 valid seat numbers, but only some may refer to online registered devices. Indexing seat numbers not referring to online devices may result in no response or in a return value of "UNKNOWN." Using a value of 0 is NOT accepted.
Values	TRUE
Examples	If in speaker list: < SET 2 SPEAK_RELEASE TRUE > < REP 2 SPEAK_LIST_STATUS NOT_IN_LIST > If in request list: < SET 2 SPEAK_RELEASE TRUE > < REP 2 REQUEST_LIST_STATUS NOT_IN_LIST > If device is registered, but unavailable:

	< SET 2 SPEAK_RELEASE TRUE > < REP 2 SPEAK_RELEASE UNKNOWN > Specified seat number is NOT in speaker or request list (no response given): < SET 2 SPEAK_RELEASE TRUE >

ALL_DELEGATE_MIC_OFF

Description	Turn all delegate microphones off
Supported Commands	SET
Indexing	None
Values	TRUE
Examples	< SET ALL_DELEGATE_MIC_OFF TRUE > < REP 1 MIC_STATUS OFF > < REP 2 MIC_STATUS OFF >

EXCLUSIVE_MUTE

Description	Assign exclusive mute
Supported Commands	SET and REP
Indexing	Seat Number **Note:** There are 65535 valid seat numbers, but only some may refer to online registered devices. Indexing seat numbers not referring to online devices may result in no response or in a return value of "UNKNOWN." Using a value of 0 is NOT accepted. The seat number requesting exclusive mute must correspond to a seat number assigned to a chairman.
Values	OFF ON
Examples	Acquire exclusive mute if the chairman is NOT already in the speaker list: < SET 1 EXCLUSIVE_MUTE ON > < REP 1 EXCLUSIVE_MUTE ON > < REP GLOBAL_MUTE ON > < REP 1 SPEAK_LIST_STATUS IN_LIST >

	Release exclusive mute if no other chairman is holding global mute: < SET 1 EXCLUSIVE_MUTE OFF > < REP 1 EXCLUSIVE_MUTE OFF > < REP GLOBAL_MUTE OFF > < REP 1 SPEAK_LIST_STATUS NOT_IN_LIST > Does NOT acquire exclusive mute (no response given; seat is not a chairman or another chairman already holds exclusive mute): < SET 1 EXCLUSIVE_MUTE ON > If device is registered, but unavailable: < SET 1 EXCLUSIVE_MUTE ON > < REP 1 EXCLUSIVE_MUTE UNKNOWN >

GLOBAL_MUTE

Description	Control global mute
Supported Commands	GET, SET, and REP
Indexing	None **Note:** The REP reflects whether the global mute state has been taken by any controller (including TPCI). There can be more than one controller which simultaneously holds the global mute.
Values	OFF ON
Examples	< GET GLOBAL_MUTE > < REP GLOBAL_MUTE OFF > Acquire global mute: < SET GLOBAL_MUTE ON > < REP GLOBAL_MUTE ON > Release global mute: < SET GLOBAL_MUTE OFF > < REP GLOBAL_MUTE OFF >

REQUEST_LIST_STATUS

Description	Search status of seats in request list
Supported Commands	GET and REP
Indexing	Seat Number

	Note: There are 65535 valid seat numbers, but only some may refer to online registered devices. Indexing seat numbers not referring to online devices may result in no response or in a return value of "UNKNOWN." The GET command with an index of 0 will GET ALL seat numbers of registered devices.
Values	NOT_IN_LIST IN_LIST
Examples	< GET 1 REQUEST_LIST_STATUS > < REP 1 REQUEST_LIST_STATUS NOT_IN_LIST >

SPEAK_LIST_STATUS

Description	Search status of seats in speaker list
Supported Commands	GET and REP
Indexing	Seat Number **Note:** There are 65535 valid seat numbers, but only some may refer to online registered devices. Indexing seat numbers not referring to online devices may result in no response or in a return value of "UNKNOWN." The GET command with an index of 0 will GET ALL seat numbers of registered devices.
Values	NOT_IN_LIST IN_LIST
Examples	< GET 1 SPEAK_LIST_STATUS > < REP 1 SPEAK_LIST_STATUS NOT_IN_LIST >

CLEAR_REQUEST_LIST

Description	Clear seats from request list
Supported Commands	SET
Indexing	None
Values	TRUE

Examples **Note: Text between asterisks, "* *," explains actions taken by the system. Only text within angle brackets, "< >," is displayed.**	< SET CLEAR_REQUEST_LIST TRUE > < REP 1 REQUEST_LIST_STATUS NOT_IN_LIST > < REP 2 REQUEST_LIST_STATUS NOT_IN_LIST > < SET CLEAR_REQUEST_LIST TRUE > *No response is provided when the request list is empty.*

NEXT_MIC_ON

Description	Turn next microphone in request list on
Supported Commands	SET
Indexing	None
Values	TRUE
Examples	< SET NEXT_MIC_ON TRUE > < REP 4 MIC_STATUS ON >

MAX_TOTAL_SPEAKERS

Description	Set maximum number of speakers allowed
Supported Commands	GET, SET, and REP
Indexing	None
Values	Format: Numeric 1 character of fixed output
Examples	< GET MAX_TOTAL_SPEAKERS > < REP MAX_TOTAL_SPEAKERS 2 > < SET MAX_TOTAL_SPEAKERS 2 > < REP MAX_TOTAL_SPEAKERS 2 >

MAX_DELEGATE_SPEAKERS

Description	Set maximum number of delegate speakers
Supported Commands	GET, SET, and REP
Indexing	None

Values	Format: Numeric 1 character of fixed output
Examples	< GET MAX_DELEGATE_SPEAKERS > < REP MAX_DELEGATE_SPEAKERS 3 > < SET MAX_DELEGATE_SPEAKERS 3 > < REP MAX_DELEGATE_SPEAKERS 3 >

MAX_NUM_REQUESTS

Description	Set maximum number of delegates allowed in request list
Supported Commands	GET, SET, and REP
Indexing	None
Values	Format: Numeric 1 character of fixed output
Examples	< GET MAX_NUM_REQUESTS > < REP MAX_NUM_REQUESTS 5 > < SET MAX_NUM_REQUESTS 5 > < REP MAX_NUM_REQUESTS 5 >

OPERATION_MODE

Description	Retrieve and set operation mode
Supported Commands	GET, SET, and REP
Indexing	None
Values	AUTO MANUAL FIFO HANDSFREE
Examples	< GET OPERATION_MODE > < REP OPERATION_MODE MANUAL >

	< SET OPERATION_MODE AUTO > < REP OPERATION_MODE AUTO >

INTERRUPT_MODE

Description	Retrieve and set interruption mode
Supported Commands	GET, SET, and REP
Indexing	None
Values	NOT_ALLOWED HIGHER_PRIORITY EQUAL_AND_HIGHER_PRIORITY
Examples	< GET INTERRUPT_MODE > < REP INTERRUPT_MODE HIGHER_PRIORITY > < SET INTERRUPT_MODE NOT_ALLOWED > < REP INTERRUPT_MODE NOT_ALLOWED >

MIC_PRIORITY

Description	Retrieve and set microphone priority
Supported Commands	GET, SET, and REP
Indexing	Seat Number **Note:** There are 65535 valid seat numbers, but only some may refer to online registered devices. Indexing seat numbers not referring to online devices may result in no response or in a return value of "UNKNOWN." The GET and SET command with an index of 0 will GET and SET ALL seat numbers of registered devices.
Values	Format: Numeric 1 character of fixed output
Examples	< GET 1 MIC_PRIORITY > < REP 1 MIC_PRIORITY 2 > < SET 1 MIC_PRIORITY 3 > < REP 1 MIC_PRIORITY 3 >

LOUDSPEAKER_VOLUME

Description	Set the volume of loudspeakers for conference units
Supported Commands	GET, SET, and REP
Indexing	None
Values	Format: Numeric 3 characters of fixed output Values REP and SET are offset by 30 Actual_Value = SetOrReportedValue - 30 Actual range: -30 to 6 dB in 1 dB steps TPCI range: 000 to 036 in steps of 1
Examples	< GET LOUDSPEAKER_VOLUME > < REP LOUDSPEAKER_VOLUME 004 > < SET LOUDSPEAKER_VOLUME 12 > < REP LOUDSPEAKER_VOLUME 012 >

AUX_INPUT_PAD

Description	Retrieve and set aux input pad
Supported Commands	GET, SET, and REP
Indexing	Aux Input **Note:** Using the GET and SET commands with an index value of 0 or 1 will GET and SET the aux input.
Values	OFF ON
Examples	< GET 1 AUX_INPUT_PAD > < REP 1 AUX_INPUT_PAD OFF > < SET 1 AUX_INPUT_PAD ON > < REP 1 AUX_INPUT_PAD ON >

AUX_INPUT_GAIN

Description	Retrieve and set aux input gain

Supported Commands	GET, SET, and REP
Indexing	Aux Input **Note:** Using the GET and SET commands with an index value of 0 or 1 will GET and SET the aux input.
Values	Format: Numeric 3 characters of fixed output Values REP and SET are offset by 30 Actual_Value = SetOrReportedValue - 30 Actual range: -30 to 10 dB in 1 dB steps TPCI range: 000 to 040 in steps of 1
Examples	< GET 1 AUX_INPUT_GAIN > < REP 1 AUX_INPUT_GAIN 4 > < SET 1 AUX_INPUT_GAIN 12 > < REP 1 AUX_INPUT_GAIN 12 >

AUX_OUTPUT_GAIN

Description	Retrieve and set aux output gain
Supported Commands	GET, SET, and REP
Indexing	Aux Output **Note:** Using the GET and SET commands with an index value of 0 or 1 will GET and SET the aux output.
Values	Format: Numeric 3 characters of fixed output Values REP and SET are offset by 30 Actual_Value = SetOrReportedValue - 30 Actual range: -30 to 0 dB in 1 dB steps TPCI range: 000 to 030 in steps of 1
Examples	< GET 1 AUX_OUTPUT_GAIN > < REP 1 AUX_OUTPUT_GAIN 4 > < SET 1 AUX_OUTPUT_GAIN 12 > < REP 1 AUX_OUTPUT_GAIN 12 >

MIC_GAIN

Description	Control the microphone gain of conference units
Supported Commands	GET, SET, and REP
Indexing	Seat Number **Note:** There are 65535 valid seat numbers, but only some may refer to online registered devices. Indexing seat numbers not referring to online devices may result in no response or in a return value of "UNKNOWN." Using the GET and SET commands with an index value of 0 will GET and SET ALL seat numbers of online registered devices. No response is given for seat numbers not referring to online devices. If AGC is enabled this command will report existing value.
Values	Format: Numeric 3 characters of fixed output Values REP and SET are offset by 30 Actual_Value = SetOrReportedValue - 30 Actual range: -30 to 10 dB in 1 dB steps TPCI range: 000 to 040 in steps of 1
Examples	If device is available: < GET 1 MIC_GAIN > < REP 1 MIC_GAIN 4 > < SET 1 MIC_GAIN 12 > < REP 1 MIC_GAIN 12 > If device is registered, but unavailable: < GET 1 MIC_GAIN > < REP 1 MIC_GAIN UNKNOWN > < SET 1 MIC_GAIN 12 > < REP 1 MIC_GAIN UNKNOWN >

DANTE_INPUT_GAIN

Description	Retrieve and set Dante input gain
Supported Commands	GET, SET, and REP
Indexing	Dante Input

	Note: Using the GET and SET commands with an index value of 0 will GET and SET ALL Dante inputs.
Values	Format: Numeric 3 characters of fixed output Values REP and SET are offset by 30 Actual_Value = SetOrReportedValue - 30 Actual range: -30 to 10 dB in 1 dB steps TPCI range: 000 to 040 in steps of 1
Examples	< GET 1 DANTE_INPUT_GAIN > < REP 1 DANTE_INPUT_GAIN 4 > < SET 1 DANTE_INPUT_GAIN 12 > < REP 1 DANTE_INPUT_GAIN 12 >

DANTE_OUTPUT_GAIN

Description	Retrieve and set Dante output gain
Supported Commands	GET, SET, and REP
Indexing	Dante Output **Note:** Using the GET and SET commands with an index value of 0 will GET and SET ALL Dante outputs.
Values	Format: Numeric 3 characters of fixed output Values REP and SET are offset by 30 Actual_Value = SetOrReportedValue - 30 Actual range: -30 to 0 dB in 1 dB steps TPCI range: 000 to 030 in steps of 1
Examples	< GET 1 DANTE_OUTPUT_GAIN > < REP 1 DANTE_OUTPUT_GAIN 4 > < SET 1 DANTE_OUTPUT_GAIN 12 > < REP 1 DANTE_OUTPUT_GAIN 12 >

AUX_INPUT_AGC

Description	Retrieve and set aux input AGC

Supported Commands	GET, SET, and REP
Indexing	Aux Input **Note:** Using the GET and SET commands with an index value of 0 or 1 will GET and SET the aux input.
Values	OFF ON
Examples	< GET 1 AUX_INPUT_AGC > < REP 1 AUX_INPUT_AGC ON > < SET 1 AUX_INPUT_AGC OFF > < REP 1 AUX_INPUT_AGC OFF >

DANTE_INPUT_AGC

Description	Retrieve and set Dante input AGC
Supported Commands	GET, SET, and REP
Indexing	Dante Input **Note:** Using the GET and SET commands with an index value of 0 will GET and SET ALL Dante inputs.
Values	OFF ON
Examples	< GET 1 DANTE_INPUT_AGC > < REP 1 DANTE_INPUT_AGC ON > < SET 1 DANTE_INPUT_AGC OFF > < REP 1 DANTE_INPUT_AGC OFF >

DANTE_INPUT_MUTE

Description	Retrieve and set Dante input mute
Supported Commands	GET, SET, and REP
Indexing	Dante Input **Note:** Using the GET and SET commands with an index value of 0 will GET and SET ALL Dante inputs.
Values	OFF

	ON
Examples	< GET 1 DANTE_INPUT_MUTE > < REP 1 DANTE_INPUT_MUTE ON > < SET 1 DANTE_INPUT_MUTE OFF > < REP 1 DANTE_INPUT_MUTE OFF >

DANTE_OUTPUT_MUTE

Description	Retrieve and set Dante output mute
Supported Commands	GET, SET, and REP
Indexing	Dante Output **Note:** Using the GET and SET commands with an index value of 0 will GET and SET ALL Dante outputs.
Values	OFF ON
Examples	< GET 1 DANTE_OUTPUT_MUTE > < REP 1 DANTE_OUTPUT_MUTE ON > < SET 1 DANTE_OUTPUT_MUTE OFF > < REP 1 DANTE_OUTPUT_MUTE OFF >

MIC_AGC

Description	Retrieve and set microphone AGC
Supported Commands	GET, SET, and REP
Indexing	Seat Number **Note:** There are 65535 valid seat numbers, but only some may refer to online registered devices. Indexing seat numbers not referring to online devices may result in no response or in a return value of "UNKNOWN." The GET and SET command with an index of 0 will GET and SET ALL seat numbers of online registered devices. No response is given for seat numbers not referring to online devices.
Values	OFF ON

Examples	If device is available: < GET 1 MIC_AGC > < REP 1 MIC_AGC ON > < SET 1 MIC_AGC OFF > < REP 1 MIC_AGC OFF > If device is registered, but unavailable: < GET 1 MIC_AGC > < REP 1 MIC_AGC UKNOWNN > < SET 1 MIC_AGC OFF > < REP 1 MIC_AGC UNKNOWN >

FLASH

Description	Turn on flash to identify a device
Supported Commands	GET, SET, and REP
Indexing	No index flashes APT. Index (Seat Number) flashes microphone **Note:** There are 65535 valid seat numbers, but only some may refer to online registered devices. Indexing seat numbers not referring to online devices may result in no response or in a return value of "UNKNOWN." The GET and SET command with an index of 0 will GET and SET ALL seat numbers of online registered devices. No response is given for seat numbers not referring to online devices.
Values	OFF ON
Examples	Flash APT: < GET FLASH > < REP FLASH ON > < SET FLASH ON > < REP FLASH ON > Flash conference unit if device is available: < GET 1 FLASH > < REP 1 FLASH OFF >

	< SET 1 FLASH ON > < REP 1 FLASH ON > Flash conference unit if device is registered, but unavailable: < GET 1 FLASH > < REP 1 FLASH UNKNOWN > < SET 1 FLASH ON > < REP 1 FLASH UNKNOWN >

ROLE

Description	Retrieve and set role for device
Supported Commands	GET, SET, and REP
Indexing	Seat Number **Note:** There are 65535 valid seat numbers, but only some may refer to online registered devices. Indexing seat numbers not referring to online devices may result in no response or in a return value of "UNKNOWN." The GET and SET command with an index of 0 will GET and SET ALL seat numbers of online registered devices. No response is given for seat numbers not referring to online devices.
Values	DELEGATE CHAIRMAN LISTENER AMBIENT REMOTE_CALLER DUAL_DELEGATE
Examples	1. GET all roles to check availability. If device is available: < GET 1 ROLE > < REP 1 ROLE DELEGATE > < SET 1 ROLE CHAIRMAN > < REP 1 ROLE CHAIRMAN > If device is registered, but unavailable:

	< GET 1 ROLE > < REP 1 ROLE UNKNOWN > < SET 1 ROLE CHAIRMAN > < REP 1 ROLE UNKNOWN > 2. SET seat number to DUAL_DELEGATE. Notice a new device shows online. < SET 1 ROLE DUAL_DELEGATE > < REP 1 ROLE DUAL_DELEGATE > < REP 2 UNIT_AVAILABLE AVAILABLE > 3. Check that the new device is the dual delegate. < GET 0 ROLE > < REP 1 ROLE DUAL_DELEGATE > < REP 2 ROLE DUAL_DELEGATE > 4. Turn the dual delegate unit back to a single delegate unit. < SET 2 ROLE DELEGATE > < REP 2 ROLE DELEGATE > < REP 1 UNIT_AVAILABLE NOT_REGISTERED > 5. Ensure only one delegate exists. < GET 0 ROLE > < REP 2 ROLE DELEGATE > 6. SET the role for all units. < SET 0 ROLE DELEGATE > < REP 2 ROLE DELEGATE > < REP 1 UNIT_AVAILABLE NOT_REGISTERED > < REP 3 ROLE DELEGATE > < REP 4 ROLE DELEGATE > < REP 6 ROLE DELEGATE > < REP 7 ROLE DELEGATE > < REP 8 ROLE DELEGATE >

SEAT_NAME

Description	Retrieve seat name
Supported Commands	GET, SET, and REP
Indexing	Seat Number **Note:** There are 65535 valid seat numbers, but only some may refer to online registered devices. Indexing seat numbers not referring to online devices may result in no response or in a return value of "UNKNOWN."

	The GET and SET command with an index of 0 will GET and SET ALL seat numbers of registered devices.
Values	UTF-8 data is allowed with the exception of characters found in the TPCI protocol: {,},<,> Maximum data size for GET: 512 bytes Maximum data size for SET: 128 bytes **Note:** Padding will occur up to 128 bytes. Padding is only guaranteed to be consistent when dealing with ASCII character sets.
Examples **Note: Text between asterisks, "* *," explains actions taken by the system. Only text within angle brackets, "< >," is displayed.** **Note: 128 bytes total occur in the brackets, "{}." Not all bytes are shown in these examples due to size limitations of the document.**	If NFC card is NOT inserted: < GET 1 SEAT_NAME > < REP 1 SEAT_NAME {JOHN} > < SET 1 SEAT_NAME {DOUG} > < REP 1 SEAT_NAME {DOUG} > If NFC card (Card Name = BILL) is inserted: < GET 1 SEAT_NAME > < REP 1 SEAT_NAME {BILL} > < SET 1 SEAT_NAME {DOUG} > < REP ERR > If NFC card (Card Name = LUKE) is inserted and removed: < GET 1 SEAT_NAME > < REP 1 SEAT_NAME {GARY} > * NFC CARD INSERTED * < REP 1 SEAT_NAME {LUKE} > * NFC CARD REMOVED * < REP 1 SEAT_NAME {GARY} >

RF_POWER

Description	Retrieve and set RF power
Supported Commands	GET, SET, and REP
Indexing	None
Values	OFF LOW MEDIUM

	HIGH MAXIMUM
Examples	< GET RF_POWER > < REP RF_POWER LOW > < SET RF_POWER HIGH > < REP RF_POWER HIGH >

DEVICE_ID

Description	Retrieve and set device ID
Supported Commands	GET, SET, and REP
Indexing	None
Values	Format: Fixed string 31 character for REP 1-31 characters from the set: A-Z,a-z,0-9, and hyphen "-" **Note:** The device ID cannot begin or end with the hyphen "-"
Examples **Note: 31 characters total occur in the brackets, "{}." Not all character spaces are shown in these examples due to size limitations of the document.**	< GET DEVICE_ID > < REP DEVICE_ID {BILL} > < SET DEVICE_ID {DOUG} > < REP DEVICE_ID {DOUG} >

ALL

Description	Retrieve all supported commands
Supported Commands	GET and REP
Indexing	None
Values	None
Examples	< GET ALL > Responds with REP for all supported commands

BATT_CHARGE

Description	View battery charge status
Supported Commands	GET and REP

Indexing	Seat Number **Note:** There are 65535 valid seat numbers, but only some may refer to online registered devices. Indexing seat numbers not referring to online devices may result in no response or in a return value of "UNKNOWN." The GET command with an index of 0 will GET ALL seat numbers of online registered devices. No response is given for seat numbers not referring to online devices.
Values	Format: 3 numbers 000-100: Percent status of charge
Examples	If device is available: < GET 5 BATT_CHARGE > < REP 5 BATT_CHARGE 85 > If device is registered, but unavailable: < GET 3 BATT_CHARGE > < REP 3 BATT_CHARGE UNKNOWN >

BATT_RUN_TIME

Description	Monitor battery life
Supported Commands	GET and REP
Indexing	Seat Number **Note:** There are 65535 valid seat numbers, but only some may refer to online registered devices. Indexing seat numbers not referring to online devices may result in no response or in a return value of "UNKNOWN." The GET command with an index of 0 will GET ALL seat numbers of online registered devices. No response is given for seat numbers not referring to online devices.
Values	Format: 5 numbers 00000-65535: Corresponds to the minutes until the conference unit turns off, according to the current power level
Examples	If device is available: < GET 1 BATT_RUN_TIME > < REP 1 BATT_RUN_TIME 00045 > If device is registered, but unavailable:

	< GET 5 BATT_RUN_TIME > < REP 5 BATT_RUN_TIME UKNOWN

BATT_CYCLE

Description	Monitor individual battery cycles
Supported Commands	GET and REP
Indexing	Seat Number **Note:** There are 65535 valid seat numbers, but only some may refer to online registered devices. Indexing seat numbers not referring to online devices may result in no response or in a return value of "UNKNOWN." The GET command with an index of 0 will GET ALL seat numbers of online registered devices. No response is given for seat numbers not referring to online devices.
Values	Format: 4 numbers 0000-9999
Examples	If device is available: < GET 2 BATT_CYCLE > < REP 2 BATT_CYCLE 0006 > If device is registered, but unavailable: < GET 5 BATT_CYCLE > < REP 5 BATT_CYCLE UNKNOWN >

BATT_HEALTH

Description	Monitor battery health percentages
Supported Commands	GET and REP
Indexing	Seat Number **Note:** There are 65535 valid seat numbers, but only some may refer to online registered devices. Indexing seat numbers not referring to online devices may result in no response or in a return value of "UNKNOWN." The GET command with an index of 0 will GET ALL seat numbers of online registered devices. No response is given for seat numbers not referring to online devices.
Values	Format: 3 numbers

	000-100: Percent status of health 255: Unknown, or not applicable
Examples	If device is available: < GET 4 BATT_HEALTH > < REP 4 BATT_HEALTH 098 > If device is registered, but unavailable: < GET 1 BATT_HEALTH > < REP 1 BATT_HEALTH UNKNOWN >

UNIT_AVAILABLE

Description	Indicates device is available
Supported Commands	GET and REP
Indexing	Seat Number **Note:** There are 65535 valid seat numbers, but only some may refer to online registered devices. Indexing seat numbers not referring to online devices may result in no response or in a return value of "UNKNOWN." The GET command with an index of 0 will GET ALL seat numbers of registered devices.
Values	AVAILABLE OFFLINE NOT_REGISTERED
Examples **Note: Text between asterisks, "* *," explains actions taken by the system. Only text within angle brackets, "< >," is displayed.**	If device is available: < GET 1 UNIT_AVAILABLE > < REP 1 UNIT_AVAILABLE AVAILABLE > If device becomes registered and is available: < REP 2 UNIT_AVAILABLE AVAILABLE > If registered device goes offline: < REP 3 UNIT_AVAILABLE OFFLINE > If device is not registered: < GET 4 UNIT_AVAILABLE > < REP 4 UNIT_AVAILABLE NOT_REGISTERED > If registered device becomes deregistered:

	< GET 5 UNIT_AVAILABLE > < REP 5 UNIT_AVAILABLE AVAILABLE > < REP 5 UNIT_AVAILABLE NOT_REGISTERED > NOT_REGISTERED will not be returned unless command targets an unregistered seat number: < GET 0 UNIT_AVAILABLE > < REP 1 UNIT_AVAILABLE AVAILABLE > < REP 2 UNIT_AVAILABLE AVAILABLE > < REP 3 UNIT_AVAILABLE OFFLINE >

AUDIO_METER_RATE

Description	Set and view audio metering rate	
Supported Commands	GET, SET, and REP	
Indexing	None	
Values	Where **rate** is from 0= off, 100-99999= interval between reported metering samples in milliseconds. Audio sampling messages are sent periodically at the rate specified in the AUDIO_METER_RATE message.	
Examples	Audio metering: < SET AUDIO_METER_RATE 1000 > < REP AUDIO_METER_RATE 1000 > < GET AUDIO_METER_RATE > < REP AUDIO_METER_RATE 1000 > < AUDIO_SAMPLE **auxInPeak auxInRms slot1Peak slot1Rms slot2Peak slot2Rms ... slotXPeak slotXRms** >	*Where **auxInPeak** is the aux input peak audio level.* *Where **auxInRms** is the aux input RMS audio level.* *Where **slot[N]Peak** is the uplink slot [n] peak audio level.* *Where **slot[N]Rms** is the uplink slot [n] RMS audio level.* *Audio RMS Level is a three digit value, which takes on the value -98 dB to 0 dB offset by 98 (i.e. 000-098). actualAudioRMS = aud - 98*

RF_METER_RATE

Description	Set and view RF metering rate	
Supported Commands	GET, SET, and REP	
Indexing	None	
Values	*Where **rate** is from 0= off, 100-99999= interval between reported metering samples in milliseconds.* *RF sampling messages are sent periodically at the rate specified in the RF_METER_RATE message.*	
Examples	RF metering: < SET RF_METER_RATE 1000 > < REP RF_METER_RATE 1000 > < GET RF_METER_RATE > < REP RF_METER_RATE 1000 > < REP **seatNum1** RSSI **value1** > < REP **seatNum2** RSSI **value2** > ... < REP **seatNumX** RSSI **valueX** >	*Where **seatNum[N]** is the seat number being reported on.* *Where **value[N]**is the average estimated receive signal power, in dBm, at the conference unit.*

AUX_INPUT_MUTE

Description	Retrieve and set aux input mute
Supported Commands	GET, SET, and REP
Indexing	Aux Input **Note:** Using the GET and SET commands with an index value of 0 or 1 will GET and SET the aux input.
Values	OFF ON
Examples	< GET 1 AUX_INPUT_MUTE > < REP 1 AUX_INPUT_MUTE ON > < SET AUX_INPUT_MUTE OFF > < REP AUX_INPUT_MUTE OFF >

AUX_OUTPUT_MUTE

Description	Retrieve and set aux output mute

Supported Commands	GET, SET, and REP
Indexing	Aux Output **Note:** Using the GET and SET commands with an index value of 0 or 1 will GET and SET the aux output.
Values	OFF ON
Examples	< GET 1 AUX_OUTPUT_MUTE > < REP 1 AUX_OUTPUT_MUTE ON > < SET 1 AUX_OUTPUT_MUTE OFF > < REP 1 AUX_OUTPUT_MUTE OFF >

MODEL

Description	Retrieve model number
Supported Commands	GET
Indexing	None
Values	Format: Fixed string of 32 characters for REP Characters from the set: A-Z and 0-9
Examples **Note: 32 characters total occur in the brackets, "{}." Not all character spaces are shown in these examples due to size limitations of the document.**	< GET MODEL > < REP MODEL {MXCWAPT} >

START_VOTE

Description	Start a new voting session
Supported Commands	SET
Indexing	None
Values	Format: Numeric 1 to 2 characters of fixed input between 1 and 50 representing the voting configuration

Examples	< SET START_VOTE 3 > < REP VOTING_CONFIGURATION 03 > < REP VOTING_STATE ACTIVE > < REP 1 INTERIM_VOTING_RESULT 0 > < REP 2 INTERIM_VOTING_RESULT 0 > < REP 3 INTERIM_VOTING_RESULT 0 > < REP 1 INTERIM_VOTING_RESULT 1 > < REP 3 INTERIM_VOTING_RESULT 1 > < SET COMPLETE_VOTE TRUE > < REP VOTING_STATE COMPLETE > < REP SHARE_VOTING_RESULTS TRUE > If the voting configuration does not exist: < SET START_VOTE 10 > < REP ERR >

COMPLETE_VOTE

Description	Complete voting session
Supported Commands	SET
Indexing	None
Values	TRUE
Examples	< SET START_VOTE 3 > < REP VOTING_CONFIGURATION 03 > < REP VOTING_STATE ACTIVE > < REP 1 INTERIM_VOTING_RESULT 0 > < REP 2 INTERIM_VOTING_RESULT 0 > < REP 3 INTERIM_VOTING_RESULT 0 > < REP 1 INTERIM_VOTING_RESULT 1 > < REP 3 INTERIM_VOTING_RESULT 1 > < SET COMPLETE_VOTE TRUE > < REP VOTING_STATE COMPLETE > < REP SHARE_VOTING_RESULTS TRUE > If voting state is inactive or complete: < SET COMPLETE_VOTE TRUE > < REP ERR >

PAUSE_VOTE

Description	Pause voting session
Supported Commands	SET

Indexing	None
Values	TRUE
Examples	< SET START_VOTE 5 > < REP VOTING_CONFIGURATION 05 > < REP VOTING_STATE ACTIVE > < REP 1 INTERIM_VOTING_RESULT 0 > < REP 2 INTERIM_VOTING_RESULT 0 > < REP 3 INTERIM_VOTING_RESULT 0 > < REP 4 INTERIM_VOTING_RESULT 0 > < REP 5 INTERIM_VOTING_RESULT 0 > < REP 1 INTERIM_VOTING_RESULT 1 > < REP 4 INTERIM_VOTING RESULT 1 > < SET PAUSE_VOTE TRUE > < REP VOTING_STATE PAUSE > < SET RESUME_VOTE TRUE > < REP VOTING_STATE ACTIVE > < REP 1 INTERIM_VOTING_RESULT 2 > < REP 3 INTERIM_VOTING_RESULT 1 > < SET COMPLETE_VOTE TRUE > < REP VOTING_STATE COMPLETE > < REP SHARE_VOTING_RESULTS TRUE > If voting state is already paused, not active, or complete: < SET PAUSE_VOTE TRUE > < REP ERR >

RESUME_VOTE

Description	Resume a paused voting session
Supported Commands	SET
Indexing	None
Values	TRUE
Examples	< SET START_VOTE 5 > < REP VOTING_CONFIGURATION 05 > < REP VOTING_STATE ACTIVE > < REP 1 INTERIM_VOTING_RESULT 0 > < REP 2 INTERIM_VOTING_RESULT 0 > < REP 3 INTERIM_VOTING_RESULT 0 > < REP 4 INTERIM_VOTING_RESULT 0 > < REP 5 INTERIM_VOTING_RESULT 0 > < REP 1 INTERIM_VOTING_RESULT 1 > < REP 4 INTERIM_VOTING RESULT 1 >

	< SET PAUSE_VOTE TRUE > < REP VOTING_STATE PAUSE > < SET RESUME_VOTE TRUE > < REP VOTING_STATE ACTIVE > < REP 1 INTERIM_VOTING_RESULT 2 > < REP 3 INTERIM_VOTING_RESULT 1 > < SET COMPLETE_VOTE TRUE > < REP VOTING_STATE COMPLETE > < REP SHARE_VOTING_RESULTS TRUE > If voting state is active, inactive, or complete: < SET RESUME_VOTE TRUE > < REP ERR >

CANCEL_VOTE

Description	Cancel voting session
Supported Commands	SET
Indexing	None
Values	TRUE
Examples	< SET START_VOTE 5 > < REP VOTING_CONFIGURATION 05 > < REP VOTING_STATE ACTIVE > < REP 1 INTERIM_VOTING_RESULT 0 > < REP 2 INTERIM_VOTING_RESULT 0 > < REP 3 INTERIM_VOTING_RESULT 0 > < REP 4 INTERIM_VOTING_RESULT 0 > < REP 5 INTERIM_VOTING_RESULT 0 > < REP 1 INTERIM_VOTING_RESULT 1 > < REP 4 INTERIM_VOTING_RESULT 1 > < SET CANCEL_VOTE TRUE > < REP VOTING_STATE INACTIVE > If voting state is inactive or complete: < SET CANCEL_VOTE TRUE > < REP ERR >

VOTING_CONFIGURATION

Description	Retrieve current voting configuration
Supported Commands	GET and REP
Indexing	None

Values	Format: Numeric 2 characters of fixed output between 01 and 50
Examples	< GET VOTING_CONFIGURATION > < REP VOTING_CONFIGURATION 01 >

VOTING_CONFIGURATION_NAME

Description	Retrieve names of voting configurations
Supported Commands	GET and REP
Indexing	Voting Configuration
Values	UTF8 characters are allowed except those used to encapsulate TPCI: {,},<,> e.g. {123456789012345678901234567890 1} Fixed character size: 31
Examples **Note: 31 characters total occur in the brackets, "{}." Not all character spaces are shown in these examples due to size limitations of the document.**	Get all voting configurations if in standalone: < GET 0 VOTING_CONFIGURATION_NAME > < REP 1 VOTING_CONFIGURATION_NAME {2-button voting} > < REP 2 VOTING_CONFIGURATION_NAME {2-button voting secret} > < REP 3 VOTING_CONFIGURATION_NAME {3-button voting} > < REP 4 VOTING_CONFIGURATION_NAME {3-button voting secret} > < REP 5 VOTING_CONFIGURATION_NAME {5-button voting} > < REP 6 VOTING_CONFIGURATION_NAME {5-button voting secret} > Get voting configuration 1 if in standalone: < GET 1 VOTING_CONFIGURATION_NAME > < REP 1 VOTING_CONFIGURATION_NAME {2-button voting} >

VOTING_BUTTON_NAME

Description	Retrieve labels of voting buttons
Supported Commands	GET and REP
Indexing	Voting Button
Values	Voting button name UTF8 characters are allowed except those used to encapsulate TPCI: {,},<,>

	e.g. {12345678901234567890123456789 01} Fixed character size: 31
Examples **Note: 31 characters total occur in the brackets, "{}." Not all character spaces are shown in these examples due to size limitations of the document.**	Get all voting button labels for 2-button voting: < GET 1 0 VOTING_BUTTON_NAME > < REP 1 1 VOTING_BUTTON_NAME {Yes} > < REP 1 2 VOTING_BUTTON_NAME {No} > Get all voting button labels for 3-button voting: < GET 3 0 VOTING_BUTTON_NAME > < REP 3 1 VOTING_BUTTON_NAME {Yes} > < REP 3 2 VOTING_BUTTON_NAME {Abstain} > < REP 3 3 VOTING _BUTTON_NAME {No} > Get all voting button labels for 5-button voting: < GET 5 0 VOTING_BUTTON_NAME > < REP 5 1 VOTING_BUTTON_NAME {++} > < REP 5 2 VOTING_BUTTON_NAME {+} > < REP 5 3 VOTING _BUTTON_NAME {0} > < REP 5 4 VOTING_BUTTON_NAME {-} > < REP 5 5 VOTING_BUTTON_NAME {––} >

VOTING_STATE

Description	Indicates state of the current voting session
Supported Commands	GET and REP
Indexing	None
Values	INACTIVE PAUSE ACTIVE COMPLETE
Examples	Voting session is off: < GET VOTING_STATE > < REP VOTING_STATE INACTIVE > Voting session started and stopped: < SET START_VOTE 3 > < REP VOTING_CONFIGURATION 03 > < REP VOTING_STATE ACTIVE > < REP 1 INTERIM_VOTING_RESULT 0 > < REP 2 INTERIM_VOTING_RESULT 0 >

	< REP 3 INTERIM_VOTING_RESULT 0 > < REP 1 INTERIM_VOTING_RESULT 1 > < REP 3 INTERIM_VOTING_RESULT 1 > < SET COMPLETE_VOTE TRUE > < REP VOTING_STATE COMPLETE > < REP SHARE_VOTING_RESULTS TRUE > Voting session started and cancelled: < SET START_VOTE 5 > < REP VOTING_CONFIGURATION 05 > < REP VOTING_STATE ACTIVE > < REP 1 INTERIM_VOTING_RESULT 0 > < REP 2 INTERIM_VOTING_RESULT 0 > < REP 3 INTERIM_VOTING_RESULT 0 > < REP 4 INTERIM_VOTING_RESULT 0 > < REP 5 INTERIM_VOTING_RESULT 0 > < REP 1 INTERIM_VOTING_RESULT 1 > < REP 4 INTERIM_VOTING_RESULT 1 > < SET CANCEL_VOTE TRUE > < REP VOTING_STATE INACTIVE > Voting session started and paused: < SET START_VOTE 5 > < REP VOTING_CONFIGURATION 05 > < REP VOTING_STATE ACTIVE > < REP 1 INTERIM_VOTING_RESULT 0 > < REP 2 INTERIM_VOTING_RESULT 0 > < REP 3 INTERIM_VOTING_RESULT 0 > < REP 4 INTERIM_VOTING_RESULT 0 > < REP 5 INTERIM_VOTING_RESULT 0 > < REP 1 INTERIM_VOTING_RESULT 1 > < REP 4 INTERIM_VOTING_RESULT 1 > < SET PAUSE_VOTE TRUE > < REP VOTING_STATE PAUSE > < SET RESUME_VOTE PAUSE > < REP VOTING_STATE ACTIVE > < REP 1 INTERIM_VOTING_RESULT 2 > < REP 3 INTERIM_VOTING_RESULT 1 > < SET COMPLETE_VOTE TRUE > < REP VOTING_STATE COMPLETE > < REP SHARE_VOTING_RESULTS TRUE >

INTERIM_VOTING_SELECTION

Description	Shows interim voting selection
Supported Commands	REP
Indexing	Seat Number

	Note: There are 65535 valid seat numbers, but only some may refer to online registered devices. Indexing seat numbers not referring to online devices may result in no response or in a return value of "UNKNOWN."
Values	Voting Button
Examples	Non-secret voting session with seats 10, 11, and 12 casting votes: < SET START_VOTE 3 > < REP VOTING_CONFIGURATION 03 > < REP VOTING_STATE ACTIVE > < REP 1 INTERIM_VOTING_RESULT 0 > < REP 2 INTERIM_VOTING_RESULT 0 > < REP 3 INTERIM_VOTING_RESULT 0 > < REP 10 INTERIM_VOTING_SELECTION 1 > < REP 1 INTERIM_VOTING_RESULT 1 > < REP 11 INTERIM_VOTING_SELECTION 1 > < REP 1 INTERIM_VOTING_RESULT 2 > < REP 12 INTERIM_VOTING_SELECTION 3 > < REP 3 INTERIM_VOTING_RESULT 1 > < SET COMPLETE_VOTE TRUE > < REP VOTING_STATE COMPLETE > < REP SHARE_VOTING_RESULTS TRUE > Secret voting session: < SET START_VOTE 4 > < REP VOTING_CONFIGURATION 04 > < REP VOTING_STATE ACTIVE > < SET COMPLETE_VOTE TRUE > < REP VOTING_STATE COMPLETE >

INTERIM_VOTING_RESULT

Description	Shows voting results as they come in from a non-secret voting session
Supported Commands	REP
Indexing	Voting Button
Values	Interim result = number of results on the specified column
Examples	Non-secret voting session: < SET START_VOTE 3 > < REP VOTING_CONFIGURATION 03 > < REP VOTING_STATE ACTIVE > < REP 1 INTERIM_VOTING_RESULT 0 > < REP 2 INTERIM_VOTING_RESULT 0 > < REP 3 INTERIM_VOTING_RESULT 0 > < REP 1 INTERIM_VOTING_RESULT 1 > < REP 3 INTERIM_VOTING_RESULT 1 > < SET COMPLETE_VOTE TRUE > < REP VOTING_STATE COMPLETE >

	< REP SHARE_VOTING_RESULTS TRUE > Non-secret voting session with seats 10, 11, and 12 casting votes: < SET START_VOTE 3 > < REP VOTING_CONFIGURATION 03 > < REP VOTING_STATE ACTIVE > < REP 1 INTERIM_VOTING_RESULT 0 > < REP 2 INTERIM_VOTING_RESULT 0 > < REP 3 INTERIM_VOTING_RESULT 0 > < REP 10 INTERIM_VOTING_SELECTION 1 > < REP 1 INTERIM_VOTING_RESULT 1 > < REP 11 INTERIM_VOTING_SELECTION 1 > < REP 1 INTERIM_VOTING_RESULT 2 > < REP 12 INTERIM_VOTING_SELECTION 3 > < REP 3 INTERIM_VOTING_RESULT 1 > < SET COMPLETE_VOTE TRUE > < REP VOTING_STATE COMPLETE > < REP SHARE_VOTING_RESULTS TRUE > Secret voting session: < SET START_VOTE 4 > < REP VOTING_CONFIGURATION 04 > < REP VOTING_STATE ACTIVE > < SET COMPLETE_VOTE TRUE > < REP VOTING_STATE COMPLETE >

FINAL_VOTING_SELECTION

Description	Retrieve voting selections
Supported Commands	GET and REP
Indexing	Seat Number **Note:** There are 65535 valid seat numbers, but only some may refer to online registered devices. Indexing seat numbers not referring to online devices may result in no response or in a return value of "UNKNOWN." The GET command with an index of 0 will GET ALL seat numbers of online registered devices. No response is given for seat numbers not referring to online devices.
Values	Voting Button Label
Examples	Get voting selection of all seats after a 3-button voting session (seats 10, 11, and 12 cast votes in the prior session): < GET 0 FINAL_VOTING_SELECTION > < REP 10 FINAL_VOTING_SELECTION 1 > < REP 11 FINAL_VOTING_SELECTION 1 > < REP 12 FINAL_VOTING_SELECTION 3 > Get voting selection of all seats after a secret vote or cancelled voting session:

	< GET 0 FINAL_VOTING_SELECTION > < REP ERR >

FINAL_VOTING_RESULT

Description	Retrieve voting results
Supported Commands	GET and REP
Indexing	Voting Button
Values	Final voting result = number of results on the specified column
Examples	Get last voting sessions result after 3-button voting: < GET 0 FINAL_VOTING_RESULT > < REP 1 FINAL_VOTING_RESULT 5 > < REP 2 FINAL_VOTING_RESULT 1 > < REP 3 FINAL_VOTING_RESULT 2 > Get last voting session result after voting cancelled: < GET 0 FINAL_VOTING_RESULT > < REP ERR >

SHARE_VOTING_RESULTS

Description	Indicates voting results being shared with all delegates. Secret voting sessions have to be specified.
Supported Commands	SET and REP
Indexing	None
Values	TRUE
Examples **Note: Text between asterisks, "* *," explains actions taken by the system. Only text within angle brackets, "< >," is displayed.**	Sharing non-secret voting results: < SET START_VOTE 1 > < REP VOTING_CONFIGURATION 01 > < REP VOTING_STATE ACTIVE > < REP 1 INTERIM_VOTING_RESULT 0 > < REP 2 INTERIM_VOTING_RESULT 0 > < REP 1 INTERIM_VOTING_RESULT 1 > < REP 2 INTERIM_VOTING_RESULT 1 > < SET COMPLETE_VOTE TRUE > < REP VOTING_STATE COMPLETE >

	Non-secret voting results are automatically shared < REP SHARE_VOTING_RESULTS TRUE > Sharing secret voting results: < SET START_VOTE 2 > < REP VOTING_CONFIGURATION 02 > < REP VOTING_STATE ACTIVE > < SET COMPLETE_VOTE TRUE > < REP VOTING_STATE COMPLETE > < SET SHARE_VOTING_RESULTS TRUE > < REP SHARE_VOTING_RESULTS TRUE >

CLOSE_VOTING_RESULTS

Description	Close view of voting results
Supported Commands	SET and REP
Indexing	None
Values	TRUE FALSE
Examples **Note: Text between asterisks, "* *," explains actions taken by the system. Only text within angle brackets, "< >," is displayed.**	Sharing and closing non-secret voting results: < SET START_VOTE 1 > < REP VOTING_CONFIGURATION 01 > < REP VOTING_STATE ACTIVE > < REP 1 INTERIM_VOTING_RESULT 0 > < REP 2 INTERIM_VOTING_RESULT 0 > < REP 1 INTERIM_VOTING_RESULT 1 > < REP 2 INTERIM_VOTING_RESULT 1 > < SET COMPLETE_VOTE TRUE > < REP VOTING_STATE COMPLETE > < REP SHARE_VOTING_RESULTS TRUE > *SET close voting results to TRUE for closing the voting results screen and returning to the home screen* < SET CLOSE_VOTING_RESULTS TRUE > < REP CLOSE_VOTING_RESULTS TRUE > Sharing and closing secret voting results: < SET START_VOTE 2 > < REP VOTING_CONFIGURATION 02 > < REP VOTING_STATE ACTIVE > < SET COMPLETE_VOTE TRUE > < REP VOTING_STATE COMPLETE >

	< SET SHARE_VOTING_RESULTS TRUE > < REP SHARE_VOTING_RESULTS TRUE > *SET close voting results to TRUE for closing the voting results screen and returning to the home screen* < SET CLOSE_VOTING_RESULTS TRUE > < REP CLOSE_VOTING_RESULTS TRUE > < SET CLOSE_VOTING_RESULTS TRUE > *No response is given when the results have already been closed the 1st time on the same voting session.*

AUDIO_INPUT_SPEAKLIST

Description	Show audio input speaklist
Supported Commands	GET, SET and REP
Indexing	None
Values	OFF ON
Examples	If AUDIO_INPUT is enabled via another source other than TPCI: < REP AUDIO_INPUT_SPEAKLIST ON > < GET AUDIO_INPUT_SPEAKLIST > < REP AUDIO_INPUT_SPEAKLIST OFF > < SET AUDIO_INPUT_SPEAKLIST ON> < REP AUDIO_INPUT_SPEAKLIST ON >

WDU_OFF

Description	Turn conference unit off
Supported Commands	SET and REP
Indexing	None
Values	TRUE
Examples	< SET WDU_OFF TRUE > < REP 1 WDU_OFF TRUE > < REP 2 WDU_OFF TRUE >

WDU_LOCK_WELCOME

Description	Set welcome screen lock status
Supported Commands	GET, SET, and REP
Indexing	None
Values	DISABLED (default - doesn't lock welcome screen) ENABLED (locks welcome screen)
Examples	< GET WDU_LOCK_WELCOME > < REP WDU_LOCK_WELCOME ENABLED > < SET WDU_LOCK_WELCOME DISABLED > < REP WDU_LOCK_WELCOME DISABLED >

WELCOME_LOCK_RESET

Description	Reset welcome screen lock to default
Supported Commands	SET and REP
Indexing	None
Values	TRUE
Examples	< SET WELCOME_LOCK_RESET TRUE > < REP WELCOME_LOCK_RESET TRUE >

RETAIN_SEAT_PERSISTENCE

Description	Retain seat information on reboot
Supported Commands	GET, SET, and REP
Indexing	None
Values	DISABLED ENABLED (default)
Examples	If device is available: < GET RETAIN_SEAT_PERSISTENCE > < REP RETAIN_SEAT_PERSISTENCE DISABLED >

	< SET RETAIN_SEAT_PERSISTENCE ENABLED > < REP RETAIN_SEAT_PERSISTENCE ENABLED > If enabled via another source other than TPCI: < REP RETAIN_SEAT_PERSISTENCE ENABLED >