



MXCW

Command Strings

MXCWAPT command strings for control systems, such as Crestron or Extron.
Version: 6 (2019-E)

Table of Contents

		Conventions	3
MXCWCommand Strings	3	Indexing	3
MXCWAPT Microflex® Complete Wireless Command Strings	3	Command Strings	4

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 4000	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	<p>Seat Number</p> <p>Note:</p> <p>There are 4000 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."</p> <p>Using a value of 0 is only applicable to the GET command.</p> <p>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.</p>
Values	<p>OFF</p> <p>ON</p>
Examples	<p>If device is available:</p> <pre>< GET 2 MIC_STATUS > < REP 2 MIC_STATUS OFF ></pre> <p>< SET 5 MIC_STATUS ON > < REP 5 MIC_STATUS ON ></p> <p>If mic is enabled via another source other than TPCI:</p> <pre>< REP 1 MIC_STATUS ON ></pre> <p>If device is registered, but unavailable:</p> <pre>< GET 2 MIC_STATUS > < REP 2 MIC_STATUS UNKNOWN ></pre> <pre>< SET 5 MIC_STATUS ON > < REP 5 MIC_STATUS UNKNOWN ></pre>

SPEAK_REQUEST

Description	Control speaker requests
--------------------	--------------------------

Supported Commands	SET
Indexing	<p>Seat Number</p> <p><i>Note:</i></p> <p>There are 4000 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."</p> <p>Using a value of 0 is NOT accepted.</p>
Values	TRUE
Examples	<p>If operation mode is automatic and speaker list is NOT full:</p> <pre>< SET 2 SPEAK_REQUEST TRUE > < REP 2 SPEAK_LIST_STATUS IN_LIST ></pre> <p>If operation mode is manual and request list is NOT full:</p> <pre>< SET 2 SPEAK_REQUEST TRUE > < REP 2 REQUEST_LIST_STATUS IN_LIST ></pre> <p>If device is registered, but unavailable:</p> <pre>< SET 2 SPEAK_REQUEST TRUE > < REP 2 SPEAK_REQUEST UNKNOWN ></pre>

SPEAK_RELEASE

Description	Release speakers in list
Supported Commands	SET
Indexing	<p>Seat Number</p> <p><i>Note:</i></p> <p>There are 4000 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."</p> <p>Using a value of 0 is NOT accepted.</p>
Values	TRUE
Examples	<p>If in speaker list:</p> <pre>< SET 2 SPEAK_RELEASE TRUE > < REP 2 SPEAK_LIST_STATUS NOT_IN_LIST ></pre> <p>If in request list:</p>

	<pre>< SET 2 SPEAK_RELEASE TRUE > < REP 2 REQUEST_LIST_STATUS NOT_IN_LIST ></pre> <p>If device is registered, but unavailable:</p> <pre>< SET 2 SPEAK_RELEASE TRUE > < REP 2 SPEAK_RELEASE UNKNOWN ></pre> <p>Specified seat number is NOT in speaker or request list (no response given):</p> <pre>< SET 2 SPEAK_RELEASE TRUE ></pre>
--	--

ALL_DELEGATE_MIC_OFF

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

EXCLUSIVE_MUTE

Description	Assign exclusive mute
Supported Commands	SET and REP
Indexing	<p>Seat Number</p> <p>Note:</p> <p>There are 4000 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."</p> <p>Using a value of 0 is NOT accepted.</p> <p>The seat number requesting exclusive mute must correspond to a seat number assigned to a chairman.</p>
Values	<p>OFF</p> <p>ON</p>
Examples	Acquire exclusive mute if the chairman is NOT already in the speaker list:

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

GLOBAL_MUTE

Description	Control global mute
Supported Commands	GET, SET, and REP
Indexing	<p>None</p> <p><i>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.</i></p>
Values	<p>OFF</p> <p>ON</p>
Examples	<pre>< GET GLOBAL_MUTE > < REP GLOBAL_MUTE OFF ></pre> <p>Acquire global mute:</p> <pre>< SET GLOBAL_MUTE ON > < REP GLOBAL_MUTE ON ></pre> <p>Release global mute:</p> <pre>< SET GLOBAL_MUTE OFF > < REP GLOBAL_MUTE OFF ></pre>

REQUEST_LIST_STATUS

Description	Search status of seats in request list
--------------------	--

Supported Commands	GET and REP
Indexing	<p>Seat Number</p> <p><i>Note:</i></p> <p>There are 4000 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."</p> <p>The GET command with an index of 0 will GET ALL seat numbers of registered devices.</p>
Values	<p>NOT_IN_LIST</p> <p>IN_LIST</p>
Examples	<p>< GET 1 REQUEST_LIST_STATUS ></p> <p>< REP 1 REQUEST_LIST_STATUS NOT_IN_LIST ></p>

SPEAK_LIST_STATUS

Description	Search status of seats in speaker list
Supported Commands	GET and REP
Indexing	<p>Seat Number</p> <p><i>Note:</i></p> <p>There are 4000 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."</p> <p>The GET command with an index of 0 will GET ALL seat numbers of registered devices.</p>
Values	<p>NOT_IN_LIST</p> <p>IN_LIST</p>
Examples	<p>< GET 1 SPEAK_LIST_STATUS ></p> <p>< REP 1 SPEAK_LIST_STATUS NOT_IN_LIST ></p>

CLEAR_REQUEST_LIST

Description	Clear seats from request list
Supported Commands	SET
Indexing	None

Values	TRUE
Examples	<pre>< SET CLEAR_REQUEST_LIST TRUE > < REP 1 REQUEST_LIST_STATUS NOT_IN_LIST > < REP 2 REQUEST_LIST_STATUS NOT_IN_LIST ></pre>

NEXT_MIC_ON

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

MAX_TOTAL_SPEAKERS

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

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
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 <i>Note:</i> There are 4000 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	<p>Format: Numeric</p> <p>3 characters of fixed output</p> <p>Values REP and SET are offset by 30</p> <p>Actual_Value = SetOrReportedValue - 30</p> <p>Actual range: -30 to 6 dB in 1 dB steps</p> <p>TPCI range: 000 to 036 in steps of 1</p>
Examples	<pre>< GET LOUDSPEAKER_VOLUME > < REP LOUDSPEAKER_VOLUME 004 > < SET LOUDSPEAKER_VOLUME 12 > < REP LOUDSPEAKER_VOLUME 012 ></pre>

AUX_INPUT_PAD

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

AUX_INPUT_GAIN

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

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

AUX_OUTPUT_GAIN

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

MIC_GAIN

Description	Control the microphone gain of conference units
--------------------	---

Supported Commands	GET, SET, and REP
Indexing	<p>Seat Number</p> <p>Note:</p> <p>There are 4000 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."</p> <p>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.</p> <p>If AGC is enabled this command will report existing value.</p>
Values	<p>Format: Numeric</p> <p>3 characters of fixed output</p> <p>Values REP and SET are offset by 30</p> <p>Actual_Value = SetOrReportedValue - 30</p> <p>Actual range: -30 to 10 dB in 1 dB steps</p> <p>TPCI range: 000 to 040 in steps of 1</p>
Examples	<p>If device is available:</p> <p>< GET 1 MIC_GAIN > < REP 1 MIC_GAIN 4 ></p> <p>< SET 1 MIC_GAIN 12 > < REP 1 MIC_GAIN 12 ></p> <p>If device is registered, but unavailable:</p> <p>< GET 1 MIC_GAIN > < REP 1 MIC_GAIN UNKNOWN ></p> <p>< SET 1 MIC_GAIN 12 > < REP 1 MIC_GAIN UNKNOWN ></p>

DANTE_INPUT_GAIN

Description	Retrieve and set Dante input gain
Supported Commands	GET, SET, and REP
Indexing	<p>Dante Input</p> <p>Note: Using the GET and SET commands with an index value of 0 will GET and SET ALL Dante inputs.</p>

Values	<p>Format: Numeric</p> <p>3 characters of fixed output</p> <p>Values REP and SET are offset by 30</p> <p>Actual_Value = SetOrReportedValue - 30</p> <p>Actual range: -30 to 10 dB in 1 dB steps</p> <p>TPCI range: 000 to 040 in steps of 1</p>
Examples	<pre>< GET 1 DANTE_INPUT_GAIN > < REP 1 DANTE_INPUT_GAIN 4 > < SET 1 DANTE_INPUT_GAIN 12 > < REP 1 DANTE_INPUT_GAIN 12 ></pre>

DANTE_OUTPUT_GAIN

Description	Retrieve and set Dante output gain
Supported Commands	GET, SET, and REP
Indexing	<p>Dante Output</p> <p><i>Note: Using the GET and SET commands with an index value of 0 will GET and SET ALL Dante outputs.</i></p>
Values	<p>Format: Numeric</p> <p>3 characters of fixed output</p> <p>Values REP and SET are offset by 30</p> <p>Actual_Value = SetOrReportedValue - 30</p> <p>Actual range: -30 to 10 dB in 1 dB steps</p> <p>TPCI range: 000 to 040 in steps of 1</p>
Examples	<pre>< GET 1 DANTE_OUTPUT_GAIN > < REP 1 DANTE_OUTPUT_GAIN 4 > < SET 1 DANTE_OUTPUT_GAIN 12 > < REP 1 DANTE_OUTPUT_GAIN 12 ></pre>

AUX_INPUT_AGC

Description	Retrieve and set aux input AGC
Supported Commands	GET, SET, and REP

Indexing	Aux Input <i>Note: Using the GET and SET commands with an index value of 0 or 1 will GET and SET the aux input.</i>
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 <i>Note: Using the GET and SET commands with an index value of 0 will GET and SET ALL Dante inputs.</i>
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 <i>Note: Using the GET and SET commands with an index value of 0 will GET and SET ALL Dante inputs.</i>
Values	OFF ON

Examples	<pre>< GET 1 DANTE_INPUT_MUTE > < REP 1 DANTE_INPUT_MUTE ON > < SET 1 DANTE_INPUT_MUTE OFF > < REP 1 DANTE_INPUT_MUTE OFF ></pre>
-----------------	--

DANTE_OUTPUT_MUTE

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

MIC_AGC

Description	Retrieve and set microphone AGC
Supported Commands	GET, SET, and REP
Indexing	Seat Number <i>Note:</i> There are 4000 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 UNKNOWNN >

< 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	<p>No index flashes APT. Index (Seat Number) flashes microphone</p> <p><i>Note:</i></p> <p>There are 4000 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."</p> <p>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.</p>
Values	<p>OFF</p> <p>ON</p>
Examples	<p>Flash APT:</p> <pre> < GET FLASH > < REP FLASH ON > < SET FLASH ON > < REP FLASH ON > </pre> <p>Flash conference unit if device is available:</p> <pre> < GET 1 FLASH > < REP 1 FLASH OFF > < SET 1 FLASH ON > < REP 1 FLASH ON > </pre> <p>Flash conference unit if device is registered, but unavailable:</p>

```

< 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	<p>Seat Number</p> <p>Note:</p> <p>There are 4000 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."</p> <p>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.</p>
Values	<p>DELEGATE</p> <p>CHAIRMAN</p> <p>LISTENER</p> <p>AMBIENT</p>
Examples	<p>If device is available:</p> <pre> < GET 1 ROLE > < REP 1 ROLE DELEGATE > < SET 1 ROLE CHAIRMAN > < REP 1 ROLE CHAIRMAN > </pre> <p>If device is registered, but unavailable:</p> <pre> < GET 1 ROLE > < REP 1 ROLE UNKNOWN > < SET 1 ROLE CHAIRMAN > < REP 1 ROLE UNKNOWN > </pre>

SEAT_NAME

Description	Retrieve seat name
Supported Commands	GET, SET, and REP

Indexing	<p>Seat Number</p> <p>Note:</p> <p>There are 4000 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."</p> <p>The GET and SET command with an index of 0 will GET and SET ALL seat numbers of registered devices.</p>
Values	<p>UTF-8 data is allowed with the exception of characters found in the TPCI protocol: {,},<,></p> <p>Maximum data size for GET: 512 bytes</p> <p>Maximum data size for SET: 128 bytes</p> <p>Note: Padding will occur up to 128 bytes. Padding is only guaranteed to be consistent when dealing with ASCII character sets.</p>
Examples	<p>If NFC card is NOT inserted:</p> <pre>< GET 1 SEAT_NAME > < REP 1 SEAT_NAME {JOHN} > < SET 1 SEAT_NAME {DOUG} > < REP 1 SEAT_NAME {DOUG} ></pre> <p>If NFC card (Card Name = BILL) is inserted:</p> <pre>< GET 1 SEAT_NAME > < REP 1 SEAT_NAME {BILL} > < SET 1 SEAT_NAME {DOUG} > < REP ERR ></pre> <p>If NFC card (Card Name = LUKE) is inserted and removed:</p> <pre>< 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} ></pre> <p><i>Note: Text between asterisks, "*"*, " explains actions taken by the system. Only text within angle brackets, "<>," is displayed.</i></p> <p><i>Note: 128 bytes total occur in the brackets, "{}." Not all bytes are shown in these examples due to size limitations of the document.</i></p>

RF_POWER

Description	Retrieve and set RF power
Supported Commands	GET, SET, and REP
Indexing	None

Values	<p>OFF</p> <p>LOW</p> <p>MEDIUM</p> <p>HIGH</p> <p>MAXIMUM</p>
Examples	<pre>< GET RF_POWER > < REP RF_POWER LOW > < SET RF_POWER HIGH > < REP RF_POWER HIGH ></pre>

DEVICE_ID

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

ALL

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

BATT_CHARGE

Description	View battery charge status
Supported Commands	GET and REP
Indexing	<p>Seat Number</p> <p><i>Note:</i></p> <p>There are 4000 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."</p> <p>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.</p>
Values	<p>Format: 3 numbers</p> <p>000-100: Percent status of charge</p>
Examples	<p>If device is available:</p> <p>< GET 5 BATT_CHARGE > < REP 5 BATT_CHARGE 85 ></p> <p>If device is registered, but unavailable:</p> <p>< GET 3 BATT_CHARGE > < REP 3 BATT_CHARGE UNKNOWN ></p>

BATT_RUN_TIME

Description	Monitor battery life
Supported Commands	GET and REP
Indexing	<p>Seat Number</p> <p><i>Note:</i></p> <p>There are 4000 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."</p> <p>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.</p>
Values	<p>Format: 5 numbers</p> <p>00000-65535: Corresponds to the minutes until the conference unit turns off, according to the current power level</p>

Examples	<p>If device is available:</p> <pre>< GET 1 BATT_RUN_TIME > < REP 1 BATT_RUN_TIME 00045 ></pre> <p>If device is registered, but unavailable:</p> <pre>< GET 5 BATT_RUN_TIME > < REP 5 BATT_RUN_TIME UNKNOWN</pre>
-----------------	--

BATT_CYCLE

Description	Monitor individual battery cycles
Supported Commands	GET and REP
Indexing	<p>Seat Number</p> <p><i>Note:</i></p> <p>There are 4000 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."</p> <p>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.</p>
Values	<p>Format: 4 numbers</p> <p>0000-9999</p>
Examples	<p>If device is available:</p> <pre>< GET 2 BATT_CYCLE > < REP 2 BATT_CYCLE 0006 ></pre> <p>If device is registered, but unavailable:</p> <pre>< GET 5 BATT_CYCLE > < REP 5 BATT_CYCLE UNKNOWN ></pre>

BATT_HEALTH

Description	Monitor battery health percentages
Supported Commands	GET and REP
Indexing	<p>Seat Number</p> <p><i>Note:</i></p>

	<p>There are 4000 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."</p> <p>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.</p>
Values	<p>Format: 3 numbers</p> <p>000-100: Percent status of health</p> <p>255: Unknown, or not applicable</p>
Examples	<p>If device is available:</p> <pre>< GET 4 BATT_HEALTH > < REP 4 BATT_HEALTH 098 ></pre> <p>If device is registered, but unavailable:</p> <pre>< GET 1 BATT_HEALTH > < REP 1 BATT_HEALTH UNKNOWN ></pre>

UNIT_AVAILABLE

Description	Indicates device is available
Supported Commands	GET and REP
Indexing	<p>Seat Number</p> <p>Note:</p> <p>There are 4000 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."</p> <p>The GET command with an index of 0 will GET ALL seat numbers of registered devices.</p>
Values	<p>AVAILABLE</p> <p>OFFLINE</p> <p>NOT_REGISTERED</p>
Examples	<p>If device is available:</p> <pre>< GET 1 UNIT_AVAILABLE > < REP 1 UNIT_AVAILABLE AVAILABLE ></pre> <p>If device becomes registered and is available:</p> <pre>< REP 2 UNIT_AVAILABLE AVAILABLE ></pre>

	<p>If registered device goes offline:</p> <pre>< REP 3 UNIT_AVAILABLE OFFLINE ></pre> <p>If device is not registered:</p> <pre>< GET 4 UNIT_AVAILABLE > < REP 4 UNIT_AVAILABLE NOT_REGISTERED ></pre> <p>If registered device becomes deregistered:</p> <pre>< GET 5 UNIT_AVAILABLE > < REP 5 UNIT_AVAILABLE AVAILABLE ></pre> <p>*The unit associated with seat 5 is deregistered from the access point.* *No REP will be returned to indicate a previously registered OFFLINE/ONLINE device has become deregistered.*</p> <p>NOT_REGISTERED will not be returned unless command targets an unregistered seat number:</p> <pre>< GET 0 UNIT_AVAILABLE > < REP 1 UNIT_AVAILABLE AVAILABLE > < REP 2 UNIT_AVAILABLE AVAILABLE > < REP 3 UNIT_AVAILABLE OFFLINE ></pre>
--	--

AUDIO_METER_RATE

Description	Set and view audio metering rate	
Supported Commands	GET, SET, and REP	
Indexing	None	
Values	<p>Where rate is from 0= off, 100-99999= interval between reported metering samples in milliseconds.</p> <p>Audio sampling messages are sent periodically at the rate specified in the AUDIO_METER_RATE message.</p>	
Examples	<p>Audio metering:</p> <pre>< 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 ></pre>	<p>Where auxInPeak is the aux input peak audio level.</p> <p>Where auxInRms is the aux input RMS audio level.</p> <p>Where slot[N]Peak is the uplink slot [n] peak audio level.</p>

		<p>Where slot[N]Rms is the uplink slot [n] RMS audio level.</p> <p>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</p>
--	--	--

RF_METER_RATE

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

AUX_INPUT_MUTE

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

Examples	<pre>< GET 1 AUX_INPUT_MUTE > < REP 1 AUX_INPUT_MUTE ON > < SET AUX_INPUT_MUTE OFF > < REP AUX_INPUT_MUTE OFF ></pre>
-----------------	--

AUX_OUTPUT_MUTE

Description	Retrieve and set aux output mute
Supported Commands	GET, SET, and REP
Indexing	<p>Aux Output</p> <p><i>Note: Using the GET and SET commands with an index value of 0 or 1 will GET and SET the aux output.</i></p>
Values	<p>OFF</p> <p>ON</p>
Examples	<pre>< GET 1 AUX_OUTPUT_MUTE > < REP 1 AUX_OUTPUT_MUTE ON > < SET 1 AUX_OUTPUT_MUTE OFF > < REP 1 AUX_OUTPUT_MUTE OFF ></pre>

MODEL

Description	Retrieve model number
Supported Commands	GET
Indexing	None
Values	<p>Format: Fixed string of 32 characters for REP</p> <p>Characters from the set: A-Z and 0-9</p>
Examples	<pre>< GET MODEL > < REP MODEL {CD2BEEEEEE13911DF81930015C5F3F612 ></pre>

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	<pre>< 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 ></pre> <p>If the voting configuration does not exist:</p> <pre>< SET START_VOTE 10 > < REP ERR ></pre>

COMPLETE_VOTE

Description	Complete voting session
Supported Commands	SET
Indexing	None
Values	TRUE
Examples	<pre>< 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 ></pre> <p>If voting state is inactive or complete:</p> <pre>< SET COMPLETE_VOTE TRUE > < REP ERR ></pre>

PAUSE_VOTE

Description	Pause voting session
Supported Commands	SET
Indexing	None
Values	TRUE
Examples	<pre> < 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 > </pre>

RESUME_VOTE

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

```

< 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	<pre> < 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 > </pre> <p>If voting state is inactive or complete:</p> <pre> < SET CANCEL_VOTE TRUE > < REP ERR > </pre>

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. {1234567890123456789012345678901} Fixed character size: 31
Examples	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} >

Note: 31 characters total occur in the brackets, "{}." Not all character spaces are shown in these examples due to size limitations of the document.

VOTING_BUTTON_NAME

Description	Retrieve labels of voting buttons
Supported Commands	GET and REP

Indexing	Voting Button
Values	<p>Voting button name</p> <p>UTF8 characters are allowed except those used to encapsulate TPCI: {,},<,></p> <p>e.g. {1234567890123456789012345678901}</p> <p>Fixed character size: 31</p>
Examples	<p>Get all voting button labels for 2-button voting:</p> <pre>< GET 1 0 VOTING_BUTTON_NAME > < REP 1 1 VOTING_BUTTON_NAME {Yes} > < REP 1 2 VOTING_BUTTON_NAME {No} ></pre> <p>Get all voting button labels for 3-button voting:</p> <pre>< 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} ></pre> <p>Get all voting button labels for 5-button voting:</p> <pre>< 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 {--} ></pre>

Note: 31 characters total occur in the brackets, "{}." Not all character spaces are shown in these examples due to size limitations of the document.

VOTING_STATE

Description	Indicates state of the current voting session
Supported Commands	GET and REP
Indexing	None
Values	<p>INACTIVE</p> <p>PAUSE</p> <p>ACTIVE</p> <p>COMPLETE</p>
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_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	<p>Non-secret voting session:</p> <pre>< 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 ></pre> <p>Secret voting session:</p> <pre>< SET START_VOTE 4 > < REP VOTING_CONFIGURATION 04 > < REP VOTING_STATE ACTIVE > < SET COMPLETE_VOTE TRUE > < REP VOTING_STATE COMPLETE ></pre>

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	<p>Get last voting sessions result after 3 Button Voting:</p> <pre>< GET 0 FINAL_VOTING_RESULT > < REP 1 FINAL_VOTING_RESULT 5 > < REP 2 FINAL_VOTING_RESULT 1 > < REP 3 FINAL_VOTING_RESULT 2 ></pre> <p>Get last voting session result after voting cancelled:</p>

```
< 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	<p>Sharing non-secret voting results:</p> <pre>< 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 ></pre> <p><i>Note: Text between asterisks, "*"*, " explains actions taken by the system. Only text within angle brackets, "<>," is displayed.</i></p> <p>*Non-secret voting results are automatically shared*</p> <pre>< REP SHARE_VOTING_RESULTS TRUE ></pre> <p>Sharing secret voting results:</p> <pre>< 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 ></pre>

CLOSE_VOTING_RESULTS

Description	
Supported Commands	SET and REP
Indexing	None
Values	TRUE

	FALSE
<p>Examples</p> <p><i>Note: Text between asterisks, "*"*, " explains actions taken by the system. Only text within angle brackets, "<>," is displayed.</i></p>	<p>Sharing and closing non-secret voting results:</p> <pre>< 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 ></pre> <p>*SET close voting results to TRUE for closing the voting results screen and returning to the home screen*</p> <pre>< SET CLOSE_VOTING_RESULTS TRUE > < REP CLOSE_VOTING_RESULTS TRUE ></pre> <p>Sharing and closing secret voting results:</p> <pre>< 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 ></pre> <p>*SET close voting results to TRUE for closing the voting results screen and returning to the home screen*</p> <pre>< SET CLOSE_VOTING_RESULTS TRUE > < REP CLOSE_VOTING_RESULTS TRUE ></pre> <pre>< SET CLOSE_VOTING_RESULTS TRUE ></pre> <p>*No response is given when the results have already been closed the 1st time on the same voting session.*</p>