

# Application Programmer's Interface

***Gentner.***

# T able of Contents

---

<b>1. Introduction</b>	<b>5</b>
<b>2. Getting Started</b>	<b>6</b>
<b>3. Format</b>	<b>7</b>
<b>4. Parameter Names and Values</b>	<b>9</b>
A. Audio	9
1. Audio Quality	9
2. Audio Input	9
3. Ringer Volume	9
4. Line Out Source	10
5. Volume Buttons	10
6. Input Mic Level	10
7. Input Line Level	10
8. Output Mix Level	11
9. Speaker Level	11
10. Line Out Level	11
11. Auto Gain Control	11
12. Full Duplex	11
13. Echo Cancellation	12
14. Echo Output Level	12
B. Video	12
1. Video Format	12
2. Video Input 1	13
3. Video Input 2	13
4. Video Output Cable	13
5. Video Input 1 Type	13
6. Video Input 2 Type	14
7. Video Quality	14
C. ISDN	14
1. Network Type	14
2. Switch Vendor	15
3. Line 1 SPID 1	15
4. Line 1 SPID	15
5. Line 2 SPID 1	15

6. Line 2 SPID 2	16
7. Line 3 SPID 1	16
8. Line 3 SPID 2	16
9. Line 1 Local Number 1	16
10. Line 1 Local Number 2	16
11. Line 2 Local Number 1	17
12. Line 2 Local Number 2	17
13. Line 3 Local Number 1	17
14. Line 3 Local Number 2	17
15. Terminate Line	18
16. Emulate Switch	18
17. Line 2 Status	18
18. Line 3 Status	18
D. Data Ports	19
1. Camera Port Mode	19
2. Camera Port Baud Rate	19
3. Camera Port Flow Control	19
4. Data Port Mode	20
5. Data Port Baud Rate	20
6. Data Port Flow Control	20
7. Aux Data Port Mode	21
8. Aux Data Port Baud Rate	21
9. Data Port Flow Control	22
10. Allow Remote Access	22
11. Mirror Camera Control	22
E. Advanced	22
1. Language	22
2. Auto Answer	23
3. Companion Mode	23
4. Auto Dial	23
5. System Name	24
6. Software Update Channel 1	24
7. Software Update Channel 2	24
8. Standby Mode	24
9. Reset System	24
F. Ethernet	25
1. IP Address Method	25
2. IP Address	25
3. Subnet Mask	25
4. Default Gateway	26
5. Web Username	26
6. Web Password	26

G.: V.35	26
1. Dialer Type	26
2. Call Profile Type	27
3. External FIFO	27
4. Call Profiles	27
<b>5. Phonebook</b>	<b>29</b>
A. Phonebook Entry Index	29
B. Call Type (Bandwidth)	30
C. Name	30
D. Prefix	30
E. Channel 1	31
F. Channel 2	31
G. Phonebook Entry Submit	31
H. Dial Phonebook Entry Index	31
<b>6. Dialing</b>	<b>32</b>
A. Call Type	32
B. Prefix	33
C. Channel 1	33
D. Channel 2	33
E. Dial	34
F. Redial	34
<b>7. Call Control</b>	<b>35</b>
A. Volume Up	35
B. Volume Down	35
C. Mute	35
D. Picture-in-Picture	36
E. Hang Up	36
F. Camera Select	36
G. Mic Status	37
<b>8. Call Diagnostics</b>	<b>38</b>
A. Transfer Rate	38
B. Encoded Audio Mode	38
C. Encoded Video Mode	38
D. Decoded Audio Mode	38
E. Decoded Video Mode	39
F. Local LSD/MLP Bandwidth	39
G. Remote LSD/MLP Bandwidth	39
H. Last Cause Code/Call Event	39
I. Line 1	40
J. Line 2	40

K. Line 3	40
L. B-Channel 1	40
M. B-Channel 2	41
N. B-Channel 3	41
O. B-Channel 4	41
P. B-Channel 5	41
Q. B-Channel 6	41

## **9. Camera Control 43**

A. FECC (Control far end camera)	43
B. Zoom Out	43
C. Zoom In	44
D. Pan Left	44
E. Pan Right	44
F. Tilt Up	45
G. Tilt Down	45
H. Get FECC Mode	45
I. Select Preset	45
J. Save Preset	46

## **10. General Information 47**

A. Software Version	47
B. Serial Number	47
C. Remote Software Version	47
D. Remote Serial Number	47
E. Company Name	48
F. Company Website	48
G. Company Phone Number	48
H. Company Tech Support	48
I. Hardware Type	49

## **11. Controlling the Menu Interface 50**

A. Remote Control Mapping	50
B. Direct Menu Placement	51

Appendix A: Wiring Status Codes	53
---------------------------------	----

Appendix B: Channel Status Codes	54
----------------------------------	----

Appendix C: Controlling the AP400	55
-----------------------------------	----

A. Dialing and Hanging Up	55
---------------------------	----

B. AP400 Status	55
-----------------	----

## Chapter 1 Introduction

This document serves as the API (Application Programmers Interface) to the APV200-IP version 5.0 (dated Aug. 19) or greater. This document will describe both the format and the values that need to be used in order to make the API work properly.

This API is intended to allow the APV200-IP UI to be manipulated from an external device. The API is ASCII string oriented and consists of two low level operations:

1. A GET operation obtains the current value of a parameter
2. A POST operation sets the current value of a parameter

This document does not serve as a manual for the APV200-IP settings. This document assumes that the developer of the programmable device understands the context and definitions of the APV200-IP settings. If not, an online manual can be obtained from [www.gentner.com](http://www.gentner.com). For other technical assistance, technical support can be contacted at 1.800.283.5936 (USA) or 1.801.974.3760.

## Chapter 2 Getting Started

In order for your programmable device to interface properly with the APV200-IP, one of the serial ports on the APV200-IP must be set to the following:

Mode = Data

The Baud Rate and Flow Control must match that of your programmable device. Your programmable device must also be set to 8 bits, no parity and one stop bit.

One thing missing in this version of API is taking and showing snapshots. In the current version of APV200-IP software, the snapshots are so tightly integrated with the menu system, that the work to pull the snapshot support out of the menu system would not be done in time for this document.

## Chapter 3 Format

The parameter names and values are exchanged with the APV200-IP through multiple mechanisms:

1. HTML

APV200-IP's which support HTML access exchange these strings through HTML GET and POST actions. This will not be described in detail here.

2. AT Commands

An APV200-IP serial port in DATA mode implements an extended AT command set.

Two proprietary AT commands support the UI API:

### **AT.name\_string?**

Get user interface parameter value for the parameter 'name\_string'. No whitespace allowed in the command. Strings are case sensitive (strings described later). This command needs to be put into the format:

```
"AT.name_string?<CR><LF>"
```

The <CR> (\r) and <LF> (\n) are return and newline characters (required). name\_string parameters are described later in this document. This command will return the following:

If the parameter name is unknown or other command errors occur:

```
<CR><LF>ERROR<CR><LF>
```

If parameter is known:

```
<CR><LF>GET name_string=value_string<CR><LF>
```

In the above case name\_string is the parameter name and value\_string is the parameter value. For example:

```
"AudioQuality=G.728"
```

It is the responsibility of the developer to parse the incoming messages from the serial port properly.

### **AT.name\_string=value\_string**

Set the user interface parameter value for the parameter 'name\_string' to the value 'value\_string'. No whitespace allowed in the command. Strings are case sensitive (string described later). This command needs to be put into the format:

```
"AT.name_string=value_string<CR><LF>"
```

The <CR> (\r) and <LF> (\n) are return and newline characters (required). This command will return the following:



If parameter is unknown, if parameter value is unknown, or other command errors occur:

```
<CR><LF>ERROR<CR><LF>
```

If parameter change is accepted.

```
<CR><LF>OK<CR><LF>
```

The parameter names and possible values are described in the next section in more specific terms.

## Chapter 4 Parameter Names and Values

### SECTION A: Audio

This section describes all the parameters you can get/post that are related to audio. These parameters can be set individually or one after another (as a page submit).

#### 1. Audio Quality

This parameter sets the audio algorithm of the next call

name\_string = "AudioQuality"

value\_string = "Auto"  
"G.728"  
"G.711"  
"G.722 56K"  
"G.722 48K"  
"Off"

Example GET: "AT.AudioQuality?<CR><LF>"

Example POST: "AT.AudioQuality=G.728<CR><LF>"

#### 2. Audio Input

This parameter sets the source of the audio input

name\_string = "AudioInput"

value\_string = "Mic"  
"Line"  
"Mix"

Example GET: "AT.AudioInput?<CR><LF>"

Example POST: "AT.AudioInput=Line<CR><LF>"

#### 3. Ringer Volume

This parameter adjusts the ringer volume (0-100%)

name\_string = "RingerVolume"

value\_string = "0" to "100"

Example GET: "AT.RingerVolume?<CR><LF>"

Example POST: "AT.RingerVolume=50<CR><LF>"

#### 4. Line Out Source

This parameter sets how you mix the output audio you hear

```
name_string = "LineOutSource"
```

```
value_string = "Remote"  
              "Remote w/mic"  
              "Remote w/line"
```

Example GET: "AT.LineOutSource?<CR><LF>"

Example POST: "AT.LineOutSource=Remote w/mic<CR><LF>"

#### 5. Volume Buttons

This parameter selects which audio output the volume buttons (+, -) control

```
name_string = "VolumeButtons"
```

```
value_string = "Speaker"  
              "Line out"  
              "Speaker/Line out"  
              "None"
```

Example GET: "AT.VolumeButtons?<CR><LF>"

Example POST: "AT.VolumeButtons=Speaker<CR><LF>"

#### 6. Input Mic Level

This parameter adjusts the input microphone level (0-100%)

```
name_string = "InputMicLevel"
```

```
value_string = "0" to "100"
```

Example GET: "AT.InputMicLevel?<CR><LF>"

Example POST: "AT.InputMicLevel=50<CR><LF>"

#### 7. Input Line Level

This parameter adjusts the input line level (0-100%)

```
name_string = "InputLineLevel"
```

```
value_string = "0" to "100"
```

Example GET: "AT.InputLineLevel?<CR><LF>"

Example POST: "AT.InputLineLevel=50<CR><LF>"

### 8. Output Mix Level

This parameter adjusts the level that the input is mixed with the output (0-100%)

name\_string = "OutputMixLevel"

value\_string = "0" to "100"

Example GET: "AT.OutputMixLevel?<CR><LF>"

Example POST: "AT.OutputMixLevel=50<CR><LF>"

### 9. Speaker Level

This parameter adjusts the speaker output volume (0-100%)

name\_string = "SpeakerLevel"

value\_string = "0" to "100"

Example GET: "AT.SpeakerLevel?<CR><LF>"

Example POST: "AT.SpeakerLevel=50<CR><LF>"

### 10. Line Out Level

This parameter adjusts the line output volume (0-100%)

name\_string = "LineOutLevel"

value\_string = "0" to "100"

Example GET: "AT.LineOutLevel?<CR><LF>"

Example POST: "AT.LineOutLevel=50<CR><LF>"

### 11. Auto Gain Control

This parameter adjusts so that the audio levels are consistent

name\_string = "AutoGainControl"

value\_string = "Off"  
"On"

Example GET: "AT.AutoGainControl?<CR><LF>"

Example POST: "AT.AutoGainControl=On<CR><LF>"

### 12. Full Duplex

This parameter, if set, allows you to hear and speak at the same time

name\_string = "EchoDuplex"

```
value_string = "Off"  
              "On"
```

Example GET: "AT.EchoDuplex?<CR><LF>"

Example POST: "AT.EchoDuplex=On<CR><LF>"

### 13. Echo Cancellation

Prevents the audio you hear from the speakers to be going out the microphone input

```
name_string = "EchoCancel"
```

```
value_string = "Off"  
              "On"
```

Example GET: "AT.EchoCancel?<CR><LF>"

Example POST: "AT.EchoCancel=On<CR><LF>"

### 14. Echo Output Level

This parameter adjusts the audio output level (0-100%)

```
name_string = "EchoOutputLevel"
```

```
value_string = "0" to "100"
```

Example GET: "AT.EchoOutputLevel?<CR><LF>"

Example POST: "AT.EchoOutputLevel=50<CR><LF>"

## SECTION B: Video

This section describes all the parameters you can get/post that are related to video. These parameters can be set individually or one after another (as a page submit).

### 1. Video Format

This parameter sets the video quality you wish to send on the next call

```
name_string = "VideoFormat"
```

```
value_string = "H.261"  
              "H.263"  
              "Auto"
```

Example GET: "AT.VideoFormat?<CR><LF>"

Example POST: "AT.VideoFormat=H.261<CR><LF>"

## 2. Video Input 1

This parameter sets the type of cable coming from video input 1

name\_string = "MainInputCable"

value\_string = "Composite"  
"S-Video"

Example GET: "AT.MainInputCable?<CR><LF>"

Example POST: "AT.MainInputCable=S-Video<CR><LF>"

## 3. Video Input 2

This parameter sets the type of cable coming from video input 2

name\_string = "AuxInputCable"

value\_string = "Composite"  
"S-Video"

Example GET: "AT.AuxInputCable?<CR><LF>"

Example POST: "AT.AuxInputCable=S-Video<CR><LF>"

## 4. Video Output Cable

This parameter sets the type of output cable going to the output monitor

name\_string = "VideoOutputCable"

value\_string = "Composite"  
"S-Video"

Example GET: "AT.VideoOutputCable?<CR><LF>"

Example POST: "AT.VideoOutputCable=Composite<CR><LF>"

## 5. Video Input 1 Type

This parameter sets the type of input (video, doc, etc.) for video input 1

name\_string = "MainInputType"

value\_string = "None"  
"Video camera"  
"Document"  
"VCR"

Example GET: "AT.MainInputType?<CR><LF>"

Example POST: "AT.MainInputType=Video camera<CR><LF>"

## 6. Video Input 2 Type

This parameter sets the type of input (video, doc, etc.) for video input 2

name\_string = "AuxInputType"

value\_string = "None"  
"Video camera"  
"Document"  
"VCR"

Example GET: "AT.AuxInputType?<CR><LF>"

Example POST: "AT.AuxInputType=Document<CR><LF>"

## 7. Video Quality

This parameter sets the preferred video quality for the next call

name\_string = "VideoQuality"

value\_string = "CIF"  
"QCIF"

Example GET: "AT.VideoQuality?<CR><LF>"

Example POST: "AT.VideoQuality=CIF<CR><LF>"

## SECTION C: ISDN

This section describes all the parameters you can get/post that are related to ISDN settings. These parameters can be set individually or one after another (as a page submit).

### 1. Network type

This parameter sets the network type of the ISDN line (for 3 BRI line system, all are the same)

name\_string = "NetworkType"

value\_string = "Custom"  
"N. ISDN 1"  
"NTT Japan"  
"ETSI"  
"1TR6 Germany"  
"VN4 France"

"Australia"

Example GET: "AT.NetworkType?<CR><LF>"

Example POST: "AT.NetworkType=N. ISDN 1<CR><LF>"

## 2. Switch vendor

This parameter sets switch provider of the ISDN line (for 3 BRI line system, all are the same)

name\_string = "SwitchVendor"

value\_string = "Unknown"

"AT&T"

"Nor. Telecom"

"Siemens"

Example GET: "AT.SwitchVendor?<CR><LF>"

Example POST: "AT.SwitchVendor=AT&T<CR><LF>"

## 3. Line 1 SPID 1

This parameter sets the SPID for line 1, b-channel 1

name\_string = "Line1Spid1"

value\_string = "X" (where X is the SPID given to you by your ISDN provider)

Example GET: "AT.Line1Spid1?<CR><LF>"

Example POST: "AT.Line1Spid1=61282819191111<CR><LF>"

## 4. Line 1 SPID 2

This parameter sets the SPID for line 1, b-channel 2

name\_string = "Line1Spid2"

value\_string = "X" (where X is the SPID given to you by your ISDN provider)

Example GET: "AT.Line1Spid2?<CR><LF>"

Example POST: "AT.Line1Spid2=61282819201111<CR><LF>"

## 5. Line 2 SPID 1

This parameter sets the SPID for line 2, b-channel 1

name\_string = "Line2Spid1"

value\_string = "X" (where X is the SPID given to you by your ISDN provider)



Example GET: "AT.Line2Spid1?<CR><LF>"

Example POST: "AT.Line2Spid1=6128281921111<CR><LF>"

### 6. Line 2 SPID 2

This parameter sets the SPID for line 2, b-channel 2

name\_string = "Line2Spid2"

value\_string = "X" (where X is the SPID given to you by your ISDN provider)

Example GET: "AT.Line2Spid2?<CR><LF>"

Example POST: "AT.Line2Spid2=61282819221111<CR><LF>"

### 7. Line 3 SPID 1

This parameter sets the SPID for line 3, b-channel 1

name\_string = "Line3Spid1"

value\_string = "X" (where X is the SPID given to you by your ISDN provider)

Example GET: "AT.Line3Spid1?<CR><LF>"

Example POST: "AT.Line3Spid1=61282819231111<CR><LF>"

### 8. Line 3 SPID 2

This parameter sets the SPID for line 3, b-channel 2

name\_string = "Line3Spid2"

value\_string = "X" (where X is the SPID given to you by your ISDN provider)

Example GET: "AT.Line3Spid2?<CR><LF>"

Example POST: "AT.Line3Spid2=61282819241111<CR><LF>"

### 9. Line 1 Local Number 1

This parameter sets the local telephone number for line 1, b-channel 1

name\_string = "Line1Local1"

value\_string = "X" (where X is the local number of your telephone line)

Example GET: "AT.Line1Local1?<CR><LF>"

Example POST: "AT.Line1Local1=8281919<CR><LF>"

### 10. Line 1 Local Number 2

This parameter sets the local telephone number for line 1, b-channel 2

name\_string = "Line1Local2"

value\_string = "X" (where X is the local number of your telephone line)

Example GET: "AT.Line1Local2?<CR><LF>"

Example POST: "AT.Line1Local2=8281920<CR><LF>"

### 11. Line 2 Local Number 1

This parameter sets the local telephone number for line 2, b-channel 1

name\_string = "Line2Local1"

value\_string = "X" (where X is the local number of your telephone line)

Example GET: "AT.Line2Local1?<CR><LF>"

Example POST: "AT.Line2Local1=8281921<CR><LF>"

### 12. Line 2 Local Number 2

This parameter sets the local telephone number for line 2, b-channel 2

name\_string = "Line2Local2"

value\_string = "X" (where X is the local number of your telephone line)

Example GET: "AT.Line2Local2?<CR><LF>"

Example POST: "AT.Line2Local2=8281922<CR><LF>"

### 13. Line 3 Local Number 1

This parameter sets the local telephone number for line 3, b-channel 1

name\_string = "Line3Local1"

value\_string = "X" (where X is the local number of your telephone line)

Example GET: "AT.Line3Local1?<CR><LF>"

Example POST: "AT.Line3Local1=8281923<CR><LF>"

### 14. Line 3 Local Number 2

This parameter sets the local telephone number for line 3, b-channel 2

name\_string = "Line3Local2"

value\_string = "X" (where X is the local number of your telephone line)

Example GET: "AT.Line3Local2?<CR><LF>"

Example POST: "AT.Line3Local2=8281924<CR><LF>"

### 15. Terminate Line

This parameter sets the line termination (set this for factory testing, not for normal use)

name\_string = "TerminateLine"

value\_string = "No" (normal use)  
"Yes"

Example GET: "AT.TerminateLine?<CR><LF>"

Example POST: "AT.TerminateLine=No<CR><LF>"

### 16. Emulate Switch

This parameter sets the switch emulation (set this for factory testing, not for normal use)

name\_string = "EmulateSwitch"

value\_string = "No" (normal use)  
"Yes"

Example GET: "AT.EmulateSwitch?<CR><LF>"

Example POST: "AT.EmulateSwitch=No<CR><LF>"

### 17. Line 2 Status

This parameter sets the status of the 2nd BRI line (3 line systems only)

name\_string = "Line2Status"

value\_string = "Enabled"  
"Disabled"

Example GET: "AT.Line2Status?<CR><LF>"

Example POST: "AT.Line2Status=Enabled<CR><LF>"

### 18. Line 3 Status

This parameter sets the status of the 3rd BRI line (3 line systems only)

name\_string = "Line3Status"

value\_string = "Enabled"  
"Disabled"

Example GET: "AT.Line3Status?<CR><LF>"

Example POST: "AT.Line3Status=Enabled<CR><LF>"

## SECTION D: Data Ports

This section describes all the parameters you can get/post that are related to data port settings. These parameters can be set individually or one after another (as a page submit).

### 1. Camera Port Mode

This parameter sets the type of connection for the camera port (serial port 1)

```
name_string = "CamMode"

value_string = "Sony camera"
               "Canon camera"
               "Panasonic camera"
               "VCR type 1"
               "Debug"
               "Data"
               "Null modem"
               "Test" (not for normal use)
```

Example GET: "AT.CamMode?<CR><LF>"

Example POST: "AT.CamMode=Panasonic camera<CR><LF>"

### 2. Camera Port Baud Rate

This parameter sets the baud rate for the camera port (serial port 1)

```
name_string = "CamBaudRate"

value_string = "300"
               "1200"
               "2400"
               "4800"
               "9600"
               "19200"
```

Example GET: "AT.CamBaudRate?<CR><LF>"

Example POST: "AT.CamBaudRate=9600<CR><LF>"

### 3. Camera Port Flow Control

This parameter sets the flow control for the camera port (serial port 1)

```
name_string = "CamFlowControl"

value_string = "None"
               "X-On/X-Off"
               "RTS/CTS"
```

Example GET: "AT.CamFlowControl?<CR><LF>"

Example POST: "AT.CamFlowControl=None<CR><LF>"

#### 4. Data Port Mode

This parameter sets the type of connection for the data port (serial port 2)

name\_string = "DataMode"

value\_string = "Sony camera"  
"Canon camera"  
"Panasonic camera"  
"VCR type 1"  
"Debug"  
"Data"  
"Null modem"  
"Test" (not for normal use)  
"AP400" (for connection to Gentner AP400 audio device)

Example GET: "AT.DataMode?<CR><LF>"

Example POST: "AT.DataMode=Data<CR><LF>"

#### 5. Data Port Baud Rate

This parameter sets the baud rate for the data port (serial port 2)

name\_string = "DataBaudRate"

value\_string = "300"  
"1200"  
"2400"  
"4800"  
"9600"  
"19200"  
"28800"  
"57600"  
"115200"

Example GET: "AT.DataBaudRate?<CR><LF>"

Example POST: "AT.DataBaudRate=19200<CR><LF>"

#### 6. Data Port Flow Control

This parameter sets the flow control for the data port (serial port 2)

name\_string = "DataFlowControl"

```
value_string = "None"
               "X-On/X-Off"
               "RTS/CTS"
```

Example GET: "AT.DataFlowControl?<CR><LF>"

Example POST: "AT.DataFlowControl=RTS/CTS<CR><LF>"

### 7. Aux Data Port Mode

This parameter sets the type of connection for the aux data port (serial port 3)

```
name_string = "AuxMode"
```

```
value_string = "Sony camera"
               "Canon camera"
               "Panasonic camera"
               "VCR type 1"
               "Debug"
               "Data"
               "Null modem"
               "Test" (not for normal use)
               "AP400" (for connection to Gentner AP400 audio device)
```

Example GET: "AT.AuxMode?<CR><LF>"

Example POST: "AT.AuxMode=Debug<CR><LF>"

### 8. Aux Data Port Baud Rate

This parameter sets the baud rate for the aux data port (serial port 3)

```
name_string = "AuxBaudRate"
```

```
value_string = "300"
               "1200"
               "2400"
               "4800"
               "9600"
               "19200"
               "28800"
               "57600"
               "115200"
```

Example GET: "AT.AuxBaudRate?<CR><LF>"

Example POST: "AT.AuxBaudRate=19200<CR><LF>"

### 9. Data Port Flow Control

This parameter sets the flow control for the aux data port (serial port 3)

name\_string = "AuxFlowControl"

value\_string = "None"  
"X-On/X-Off"  
"RTS/CTS"

Example GET: "AT.AuxFlowControl?<CR><LF>"

Example POST: "AT.AuxFlowControl=None<CR><LF>"

### 10. Allow Remote Access

This parameter allows the remote site to control the local camera

name\_string = "AllowRemoteAccess"

value\_string = "No"  
"Send only"  
"Send/Receive"

Example GET: "AT.AllowRemoteAccess?<CR><LF>"

Example POST: "AT.AllowRemoteAccess=Send/Receive<CR><LF>"

### 11. Mirror Camera Control

This parameter mirrors the pan left/right of the local camera (switches the direction)

name\_string = "MirrorCam"

value\_string = "No"  
"Yes"

Example GET: "AT.MirrorCam?<CR><LF>"

Example POST: "AT.MirrorCam=No<CR><LF>"

## SECTION E: Advanced

This section describes all the parameters you can get/post that are related to advanced settings. These parameters can be set individually or one after another (as a page submit).

### 1. Language

This parameter sets the language of the APV200-IP on-screen menu system (currently English only)

```
name_string = "Language"
```

```
value_string = "English"
```

```
Example GET: "AT.Language?<CR><LF>"
```

```
Example POST: "AT.Language=English<CR><LF>"
```

## 2. Auto Answer

This parameter sets the auto answer mode of the APV200-IP

```
name_string = "AutoAnswer"
```

```
value_string = "Yes"
```

```
"No"
```

```
"Video"
```

```
"Yes (Audio muted)"
```

```
Example GET: "AT.AutoAnswer?<CR><LF>"
```

```
Example POST: "AT.AutoAnswer=Yes<CR><LF>"
```

## 3. Companion Mode

This parameter allows 2 systems to connect together via SCSI

```
name_string = "CompanionMode"
```

```
value_string = "None"
```

```
"Master"
```

```
"Slave"
```

```
Example GET: "AT.CompanionMode?<CR><LF>"
```

```
Example POST: "AT.CompanionMode=None<CR><LF>"
```

## 4. Auto Dial

This parameter sets the phonebook entry to automatically dial when the APV200-IP starts up

```
name_string = "AutoDial"
```

```
value_string = "None"
```

```
"1"
```

```
"2"
```

```
"3"
```

```
Example GET: "AT.AutoDial?<CR><LF>"
```

```
Example POST: "AT.AutoDial=None<CR><LF>"
```



## 5. System Name

This parameter sets the system name for MCUs to see

name\_string = "SysName"

value\_string = "NAME" (where NAME is the name you wish to set the system name to)

Example GET: "AT.SysName?<CR><LF>"

Example POST: "AT.SysName=APV200-IP<CR><LF>"

## 6. Software Update Channel 1

This parameter sets channel 1 for the software update server

name\_string = "SfUpdUpdateCh1"

value\_string = "X" (where X is the first number of the software update server)

Example GET: "AT.SfUpdUpdateCh1?<CR><LF>"

Example POST: "AT.SfUpdUpdateCh1=6128281919<CR><LF>"

## 7. Software Update Channel 2

This parameter sets channel 2 for the software update server

name\_string = "SfUpdUpdateCh2"

value\_string = "X" (where X is the second number of the software update server)

Example GET: "AT.SfUpdUpdateCh2?<CR><LF>"

Example POST: "AT.SfUpdUpdateCh2=6128281920<CR><LF>"

## 8. Standby Mode

This parameter sets the idle mode of operation (screen saver mode)

name\_string = "StandbyMode"

value\_string = "Local video"  
"Active" (floating screen saver)

Example GET: "AT.StandbyMode?<CR><LF>"

Example POST: "AT.StandbyMode=Active<CR><LF>"

## 9. Reset System

This parameter sends a "reset system" to the APV200-IP

```
name_string = "SubmitReset"
```

```
value_string = "" (IGNORED. Just put in an empty string)
```

```
Example GET: NONE
```

```
Example POST: "AT.SubmitReset=<CR><LF>"
```

## SECTION F: Ethernet

This section describes all the parameters you can get/post that are related to ethernet settings. These parameters can be set individually or one after another (as a page submit).

### 1. IP Address Method

This parameter sets the addressing method (auto/manual) for IP settings

```
name_string = "IPAddressMethod"
```

```
value_string = "User entered"  
              "Automatic DHCP"
```

```
Example GET: "AT.IPAddressMethod?<CR><LF>"
```

```
Example POST: "AT.IPAddressMethod=Automatic DHCP<CR><LF>"
```

### 2. IP Address

This parameter sets the IP address for the APV200-IP

```
name_string = "IPAddress"
```

```
value_string = "XXX.XXX.XXX.XXX" (this uses dot '.' notation)
```

```
Example GET: "AT.IPAddress?<CR><LF>"
```

```
Example POST: "AT.IPAddress=208.40.136.95<CR><LF>"
```

### 3. Subnet Mask

This parameter sets the subnet mask for the APV200-IP

```
name_string = "NetMask"
```

```
value_string = "XXX.XXX.XXX.XXX" (this uses dot '.' notation)
```

```
Example GET: "AT.NetMask?<CR><LF>"
```

```
Example POST: "AT.NetMask=255.255.255.128<CR><LF>"
```

#### 4. Default Gateway

This parameter sets the default gateway for the APV200-IP

```
name_string = "DefaultGateway"
```

```
value_string = "XXX.XXX.XXX.XXX" (this uses dot '.' notation)
```

```
Example GET: "AT.DefaultGateway?<CR><LF>"
```

```
Example POST: "AT.DefaultGateway=208.40.136.1<CR><LF>"
```

#### 5. Web Username

The username to access this system through web pages

```
name_string = "WebUserName"
```

```
value_string = "USERNAME" (where USERNAME is replaced by desired web username)
```

```
Example GET: "AT.WebUserName?<CR><LF>"
```

```
Example POST: "AT.WebUserName=GENTNER<CR><LF>"
```

#### 6. Web Password

The password to access this system through web pages

```
name_string = "WebUserPassWord"
```

```
value_string = "PASSWORD" (where PASSWORD is replaced by desired web password)
```

```
Example GET: "AT.WebUserPassWord?<CR><LF>"
```

```
Example POST: "AT.WebUserPassWord=VIDEO<CR><LF>"
```

## SECTION G: V.35

This section describes all the parameters you can get/post that are related to V.35 settings (if applicable). These parameters will only work properly if you have a V.35 system.

#### 1. V.35 Dialer Type

This parameter sets the dial type of the APV200-IP to match that of the IMUX.

```
name_string = "V35DialType"
```

```
value_string = "Leased line"  
              "RS-366"  
              "DTR"
```

Example GET: "AT.V35DialType?<CR><LF>"

Example POST: "AT.V35DialType=RS-366<CR><LF>"

## 2. V.35 Call Profile Type

This parameter sets the profile type (prefix / suffix) used in the IMUX. Call profile types of type "Prefix" automatically put the profile before each number dialed. "Suffix" puts the profile after each number dialed.

name\_string = "V35ProfileType"

value\_string = "Prefix"

"Suffix"

Example GET: "AT.V35ProfileType?<CR><LF>"

Example POST: "AT.V35ProfileType=Prefix<CR><LF>"

## 3. V.35 External FIFO

This parameter sets the External FIFO parameter in the APV200-IP system.

name\_string = "V35ExtFifo"

value\_string = "Yes"

"No"

Example GET: "AT.V35ExtFifo?<CR><LF>"

Example POST: "AT.V35ExtFifo=Yes<CR><LF>"

## 4. Call Profiles

The following is a list of call profiles that you can set. Call profiles are numbers that the IMUX uses to identify a particular call type (instead of the call type that ISDN uses). First is an example of a call profile for GET/POST, then the list of different call profiles that can be set. Just substitute the "name\_string" to manipulate a difference call profile.

name\_string = "V35CP64K2ChBond"

value\_string = "XXX" (Where XXX is the number of the particular call profile)

Example GET: "AT.V35CP64K2ChBond?<CR><LF>"

Example POST: "AT.V35CP64K2ChBond=#12<CR><LF>"

List of possible call profiles:

"V35CPVoice"

"V35CP56K1ChH221"  
"V35CP64K1ChH221"  
"V35CP56K2ChH221"  
"V35CP64K2ChH221"  
"V35CP56K2ChBond"  
"V35CP64K2ChBond"  
"V35CP56K3ChBond"  
"V35CP64K3ChBond"  
"V35CP56K4ChBond"  
"V35CP64K4ChBond"  
"V35CP56K5ChBond"  
"V35CP64K5ChBond"  
"V35CP56K6ChBond"  
"V35CP64K6ChBond"

## Chapter 5 Phonebook

The phonebook entries can be manipulated in several ways. They can be retrieved (GET), edited (POST), or dialed (POST). This chapter explains how to retrieve and edit phonebook entries. Dialing phonebook entries is explained in Chapter 6, "Dialing." The phonebook entries are indexed 1 through 45. The developer of the programmable device needs to keep track of the "current" entry we are viewing (or editing).

Before doing any GET/POST operations, the APV200-IP needs to know which phonebook entry to perform the operation. This is why the developer needs to keep track of the phonebook entry "index" we are on. That index will be sent before each GET/POST action. Examples will follow as will parameter names/values.

Ex. Retrieving the first phonebook entry:

```
POST: "AT.PhonebookEntry=1<CR><LF>"      (Set the phonebook entry to 1)
GET:  "AT.PBEntryBandwidth?<CR><LF>"      (Get the call type of entry 1)
GET:  "AT.PBEntryName?<CR><LF>"          (Get the name of entry 1)
GET:  "AT.PBEntryPrefix?<CR><LF>"         (Get the prefix of entry 1)
GET:  "AT.PBEntryChannel1?<CR><LF>"       (Get channel 1 of entry 1)
GET:  "AT.PBEntryChannel2?<CR><LF>"       (Get channel 2 of entry 1)
```

Ex. Changing phonebook entry 8 to new values:

```
POST: "AT.PhonebookEntry=8<CR><LF>"      (Set the phonebook entry to 8)
POST: "AT.PBEntryBandwidth=2Ch 64K (128)<CR><LF>" (Set call type)
POST: "AT.PBEntryName=Joe<CR><LF>"        (Set the name)
POST: "AT.PBEntryPrefix=612<CR><LF>"       (Set the prefix)
POST: "AT.PBEntryChannel1=8281919<CR><LF>" (Set channel 1)
POST: "AT.PBEntryChannel2=8281920<CR><LF>" (Set channel 2)
POST: "AT.PBEntrySubmit=<CR><LF>"        (Save entry)
```

The above are just examples of how one can manipulate each phonebook entry. The parameter names and values are explained below.

### A. Phonebook Entry Index

This sets the index (1 – 45) of the phonebook entry you want to view/edit/dial.

```
name_string = "PhonebookEntry"
```

```
value_string = "X" (where X is the number of the phonebook entry you wish to modify)
```

```
Example GET: NONE
```

```
Example POST: "AT.PhonebookEntry=10<CR><LF>" (The 10th entry)
```

### B. Call Type (Bandwidth)

The call type of the phonebook entry

```
name_string = "PEntryBandwidth"
```

```
value_string = "Voice"
               "1Ch 56K (56)"
               "1Ch 64K (64)"
               "2x56"
               "2x64"
               "2Ch 56K (112)"
               "2Ch 64K (128)"
               "3Ch 56K (168)"
               "3Ch 64K (192)"
               "4Ch 56K (224)"
               "4Ch 64K (256)"
               "5Ch 56K (280)"
               "5Ch 64K (320)"
               "6Ch 56K (336)"
               "6Ch 64K (384)"
```

Example GET: "AT.PEntryBandwidth?<CR><LF>"

Example POST: "AT.PEntryBandwidth=6Ch 64K (384)<CR><LF>"

### C. Name

The name of the phonebook entry

```
name_string = "PEntryName"
```

```
value_string = "NAME" (where NAME is the name string for this phonebook entry)
```

Example GET: "AT.PEntryName?<CR><LF>"

Example POST: "AT.PEntryName=Joe<CR><LF>"

### D. Prefix

The prefix of the phonebook entry

```
name_string = "PEntryPrefix"
```

```
value_string = "PREFIX" (where PREFIX is the prefix string for the phonebook entry)
```

Example GET: "AT.PEntryPrefix?<CR><LF>"

Example POST: "AT.PEntryPrefix=612<CR><LF>"

**E. Channel 1**

The first channel of the phonebook entry

name\_string = "PEntryChannel1"

value\_string = "CHANNEL1" (where CHANNEL1 is the first channel of the phonebook entry)

Example GET: "AT.PEntryChannel1?<CR><LF>"

Example POST: "AT.PEntryChannel1=8281919<CR><LF>"

**F. Channel 2**

The second channel of the phonebook entry

name\_string = "PEntryChannel2"

value\_string = "CHANNEL2" (where CHANNEL2 is the second channel of this entry)

Example GET: "AT.PEntryChannel2?<CR><LF>"

Example POST: "AT.PEntryChannel2=8281920<CR><LF>"

**G. Phonebook Entry Submit**

This saves the phonebook entry to the APV200-IP memory

name\_string = "SubmitPEntry"

value\_string = "" (IGNORED – Just put an empty string)

Example GET: NONE

Example POST: "AT.SubmitPEntry=<CR><LF>"

**H. Dial Phonebook Entry Index**

This allows you to dial a specific index in the phonebook (1-48)

name\_string = "PhonebookEntryIndexDial"

value\_string = "X" (Where X is the index into the phonebook you wish to dial)

Example GET: NONE

Example POST: "AT.PhonebookEntryIndexDial=1<CR><LF>"



## Chapter 6 Dialing

Manually dialing using the POST API is quite easy. There are some small steps to go through, but the dialing process happens by a series of "POST" commands to the APV200-IP. There are no GET commands for manually dialed calls. Follow the following procedure:

Keep track locally of the following things (all parameters for these described below):

- Call type
- Prefix
- Channel 1
- Channel 2

These are all strings (values will be shown later) that get sent to the APV200-IP when the user hits a "dial" button on the programmable device. The user should have the option to set up a manually dialed call in the following manner:

- Set the call type
- Enter a prefix if necessary
- Enter the number for channel 1
- Enter the number for channel 2
- Press or submit a "dial" command.

When the user is setting the call type, prefix, etc. DO NOT do a post each time. It is only necessary to post this information when the user actually hits "dial." There should be five POST operation when the user presses dial. The following is an example:

User presses DIAL:

```
POST Call Type:      "AT.CallType= 2Ch 64K ( 128)<CR><LF>"
POST Prefix:         "AT.Prefix=612<CR><LF>"
POST Channel 1:     "AT.Channel1=8281919<CR><LF>"
POST Channel 2:     "AT.Channel2=8281920<CR><LF>"
POST DIAL:          "AT.SubmitDial=<CR><LF>"
```

By doing these in order, the APV200-IP fills in the correct information into a local structure and when it sees the "SubmitDial", it uses that information to dial the call. The next section shows the parameter names/values for making a manually dialed call.

### A. Call Type

The call type of the manually dialed call (post only)

```
name_string = "CallType"
```

```
value_string = "Voice"
               "1Ch 56K (56)"
               "1Ch 64K (64)"
               "2x56"
               "2x64"
               "2Ch 56K (112)"
               "2Ch 64K (128)"
               "3Ch 56K (168)"
               "3Ch 64K (192)"
               "4Ch 56K (224)"
               "4Ch 64K (256)"
               "5Ch 56K (280)"
               "5Ch 64K (320)"
               "6Ch 56K (336)"
               "6Ch 64K (384)"
```

Example GET: NONE

Example POST: "AT.CallType=6Ch 64K (384)<CR><LF>"

### B. Prefix

The prefix of the manually dialed call (post only)

```
name_string = "Prefix"
```

```
value_string = "PREFIX" (where PREFIX is the prefix string for the manually dialed call)
```

Example GET: NONE

Example POST: "AT.Prefix=612<CR><LF>"

### C. Channel 1

The first channel of the manually dialed call (post only)

```
name_string = "Channel1"
```

```
value_string = "CHANNEL1" (where CHANNEL1 is the first channel of the call)
```

Example GET: NONE

Example POST: "AT.Channel1=8281919<CR><LF>"

### D. Channel 2

The second channel of the manually dialed call (post only, not needed for all BONDING call types)

```
name_string = "Channel2"
```

value\_string = "CHANNEL2" (where CHANNEL2 is the second channel of the call)

Example GET: NONE

Example POST: "AT.Channel2=8281920<CR><LF>"

### **E. Dial**

This dials the manually dialed call (need to POST all above parameters FIRST)

name\_string = "SubmitDial"

value\_string = "" (IGNORED – Just put an empty string here)

Example GET: NONE

Example POST: "AT.SubmitDial=<CR><LF>"

### **F. Redial**

This redials the last call dialed

name\_string = "SubmitDial"

value\_string = "" (IGNORED – Just put an empty string here)

Example GET: NONE

Example POST: "AT.Redial=<CR><LF>"

### **Note: Phonebook Entry Dialing**

Phonebook dialing is the same to manually dialed calls. Just do a POST with all of the information from the phonebook entry you are viewing, and do a "SubmitDial". This will dial all of the information you just sent the APV200-IP (call type, channels, etc.).

## Chapter 7 Call Control

Call control is an important part of the API. With it, you can send commands to the APV200-IP while on a call like mute, picture-in-picture and camera select. Here are their parameters and values:

### A. Volume Up

This increases the speaker/line out volume

```
name_string = "SubmitVolUp"
```

```
value_string = "" (IGNORED – Just put an empty string here)
```

Example GET: NONE

```
Example POST: "AT.SubmitVolUp=<CR><LF>"
```

#### NOTE:

To continue to increase the volume, the programmable device must continue to send volume up commands. One command only changes the value in small increments.

### B. Volume Down

This decreases the speaker/line out volume

```
name_string = "SubmitVolDown"
```

```
value_string = "" (IGNORED – Just put an empty string here)
```

Example GET: NONE

```
Example POST: "AT.SubmitVolDown=<CR><LF>"
```

#### NOTE:

To continue to decrease the volume, the programmable device must continue to send volume down commands. One command only changes the value in small increments.

### C. Mute

This toggles the mute of the input

```
name_string = "SubmitMute"
```

```
value_string = "" (IGNORED – Just put an empty string here)
```

Example GET: NONE

```
Example POST: "AT.SubmitMute=<CR><LF>"
```

**NOTE:**

This will turn on/off the input source. An indication on the video screen will appear as to the state of the microphone (there will be a mic muted icon on the screen if the audio is muted).

**D. Picture-in-Picture**

POST - This moves the PIP image around the screen (and off)

GET – This asks the APV200-IP the current position of the PIP

Values are (GET ONLY):

"LR" – Lower Right

"UR" – Upper Right

"LL" – Lower Left

"UL" – Upper Left

"Off" - Off

name\_string = "SubmitPip"

value\_string = "" (IGNORED – Just put an empty string here)

Example GET: "AT.SubmitPip?<CR><LF>"

Example POST: "AT.SubmitPip=<CR><LF>"

**NOTE:**

This will move the PIP image around the screen. The pip has the following values:

-Upper Right

-Lower Right

-Lower Left

-Upper Left

-Off

-Local Video

Continually pressing the PIP button will move through these values one at a time.

**E. Hang Up**

This hangs up the current call

name\_string = "SubmitHangUp"

value\_string = "" (IGNORED – Just put an empty string here)

Example GET: NONE

Example POST: "AT.SubmitHangUp=<CR><LF>"

**F. Camera Select**

This switches the input from main camera to aux camera and vica versa (POST)

This asks which video input source we are on (GET)

-Returns a "1" for input source 1

-Return a "2" for input source 2

name\_string = "SelectCamera"

value\_string = "" (IGNORED – Just put an empty string here)

Example GET: "AT.SelectCamera?<CR><LF>"

Example POST: "AT.SelectCamera=<CR><LF>"

### **G. Mic Status**

This is a GET command which gets the status of the microphone

-Returns "Off" if the microphone is off

-Return "On" if the microphone is on

name\_string = "MicStatus"

value\_string = "" (IGNORED – Just put an empty string here)

Example GET: "AT.MicStatus?<CR><LF>"

Example POST: NONE

## Chapter 8 Call Diagnostics

Call diagnostics play an important role in that it allows the user to see all of the things that are happening while on a call, such as transfer rate and connection status. Here are their parameters and values.

NOTE:

There are only GET functions for the call diagnostics, as they are just for display purposes only.

### A. Transfer Rate

This tells the programmable device the transfer rate of the current call (or idle)

name\_string = "TransferRate"

value\_string = NONE (GET only)

Example GET: "AT.TransferRate?<CR><LF>"

Example POST: NONE

### B. Encoded Audio Mode

This tells the programmable device the encoded (send) audio mode of the current call

name\_string = "EncAudioMode"

value\_string = NONE (GET only)

Example GET: "AT.EncAudioMode?<CR><LF>"

Example POST: NONE

### C. Encoded Video Mode

This tells the programmable device the encoded (send) video mode of the current call

name\_string = "EncVideoMode"

value\_string = NONE (GET only)

Example GET: "AT.EncVideoMode?<CR><LF>"

Example POST: NONE

### D. Decoded Audio Mode

This tells the programmable device the decoded (receive) audio mode of the current call

name\_string = "DecAudioMode"

value\_string = NONE (GET only)

Example GET: "AT.DecAudioMode?<CR><LF>"

Example POST: NONE

### **E. Decoded Video Mode**

This tells the programmable device the decoded (receive) video mode of the current call

name\_string = "DecVideoMode"

value\_string = NONE (GET only)

Example GET: "AT.DecVideoMode?<CR><LF>"

Example POST: NONE

### **F. Local LSD/MLP Bandwidth**

This tells the programmable device the local (send) LSD/MLP Bandwidth of the current call

name\_string = "LocalLsdMlpBandwidth"

value\_string = NONE (GET only)

Example GET: "AT.LocalLsdMlpBandwidth?<CR><LF>"

Example POST: NONE

### **G. Remote LSD/MLP Bandwidth**

This tells the programmable device the remote (receive) LSD/MLP Bandwidth of the current call

name\_string = "RemoteLsdMlpBandwidth"

value\_string = NONE (GET only)

Example GET: "AT.RemoteLsdMlpBandwidth?<CR><LF>"

Example POST: NONE

### **H. Last Cause Code/Call Event**

This tells the programmable device the last cause code or call event that happened in the current call

name\_string = "LastCauseCode"

value\_string = NONE (GET only)

Example GET: "AT.LastCauseCode?<CR><LF>"

Example POST: NONE



NOTE:: See Appendix A for Wiring Status Codes

### **I. Wiring Status for Line 1**

This tells the programmable device the status of ISDN line 1

name\_string = "WiringState1"

value\_string = NONE (GET only)

Example GET: "AT.WiringState1?<CR><LF>"

Example POST: NONE

### **J. Wiring Status for Line 2**

This tells the programmable device the status of ISDN line 2

name\_string = "WiringState2"

value\_string = NONE (GET only)

Example GET: "AT.WiringState2?<CR><LF>"

Example POST: NONE

### **K. Wiring Status for Line 3**

This tells the programmable device the status of ISDN line 3

name\_string = "WiringState3"

value\_string = NONE (GET only)

Example GET: "AT.WiringState3?<CR><LF>"

Example POST: NONE

NOTE: See Appendix B for Channel Status Codes

### **L. Channel Status B-Channel 1**

This tells the programmable device the channel status of ISDN b-channel 1

name\_string = "Channel1State"

value\_string = NONE (GET only)

Example GET: "AT. Channel1State?<CR><LF>"

Example POST: NONE

**M. Channel Status B-Channel 2**

This tells the programmable device the channel status of ISDN b-channel 2

name\_string = "Channel2State"

value\_string = NONE (GET only)

Example GET: "AT.Channel2State?<CR><LF>"

Example POST: NONE

**N. Channel Status B-Channel 3**

This tells the programmable device the channel status of ISDN b-channel 3

name\_string = "Channel3State"

value\_string = NONE (GET only)

Example GET: "AT.Channel3State?<CR><LF>"

Example POST: NONE

**O. Channel Status B-Channel 4**

This tells the programmable device the channel status of ISDN b-channel 4

name\_string = "Channel4State"

value\_string = NONE (GET only)

Example GET: "AT.Channel4State?<CR><LF>"

Example POST: NONE

**P. Channel Status B-Channel 5**

This tells the programmable device the channel status of ISDN b-channel 5

name\_string = "Channel5State"

value\_string = NONE (GET only)

Example GET: "AT.Channel5State?<CR><LF>"

Example POST: NONE

**Q. Channel Status B-Channel 6**

This tells the programmable device the channel status of ISDN b-channel 6

name\_string = "Channel6State"

value\_string = NONE (GET only)

Example GET: "AT. Channel6State?<CR><LF>"

Example POST: NONE

## Chapter 9 Camera Control

The local camera can be controlled through the serial port if is setup properly using one of the data ports. The camera's mode must be set to 9600 baud and no flow control to work properly. The mode set (ex. Sony Camera) must also match the camera. If the camera control cable is hooked up properly it will work.

### QUICK TEST:

If the camera can be run from the APV200-IP remote control, then it can be run from the programmable device as well.

Here are the parameters and values for camera control.

### A. FECC (Control far end camera)

Tell the APV200-IP to control the far end camera (toggles between far to near)

```
name_string = "LocalRemoteCamera"
```

```
value_string = "" (IGNORED – Just put an empty string here)
```

Example GET: NONE

Example POST: "AT. LocalRemoteCamera=<CR><LF>"

### NOTE:

The video screen will display an icon to say we are in "far" camera mode. The "LocalRemoteCamera" command toggles the FECC mode from local (no icon) to remote (icon) modes.

### B. Zoom Out

Zoom the local / remote camera out (depends on which FECC mode we are in)

```
name_string = "CameraZoomOut"
```

```
value_string = "" (IGNORED – Just put an empty string here)
```

Example GET: NONE

Example POST: "AT. CameraZoomOut=<CR><LF>"

### NOTE:

The developer of the programmable device is responsible for sending the APV200-IP continuous commands (for continuous motion). One command only moves the camera in small increments.

### C. Zoom In

Zoom the local / remote camera in (depends on which FECC mode we are in)

```
name_string = "CameraZoomIn"
```

```
value_string = "" (IGNORED – Just put an empty string here)
```

Example GET: NONE

Example POST: "AT. CameraZoomIn=<CR><LF>"

NOTE:

The developer of the programmable device is responsible for sending the APV200-IP continuous commands (for continuous motion). One command only moves the camera in small increments.

### D. Pan Left

Pan the local / remote camera left (depends on which FECC mode we are in and mirror camera mode)

```
name_string = "CameraPanLeft"
```

```
value_string = "" (IGNORED – Just put an empty string here)
```

Example GET: NONE

Example POST: "AT. CameraPanLeft=<CR><LF>"

NOTE:

The developer of the programmable device is responsible for sending the APV200-IP continuous commands (for continuous motion). One command only moves the camera in small increments.

### E. Pan Right

Pan the local / remote camera right (depends on which FECC mode we are in and mirror camera mode)

```
name_string = "CameraPanRight"
```

```
value_string = "" (IGNORED – Just put an empty string here)
```

Example GET: NONE

Example POST: "AT. CameraPanRight=<CR><LF>"

NOTE:

The developer of the programmable device is responsible for sending the APV200-IP continuous commands (for continuous motion). One command only moves the camera in small increments.

### F. Tilt Up

Tilt the local / remote camera up (depends on which FECC mode we are in)

```
name_string = "CameraTiltUp"
```

```
value_string = "" (IGNORED – Just put an empty string here)
```

Example GET: NONE

Example POST: "AT.CameraTiltUp=<CR><LF>"

NOTE:

The developer of the programmable device is responsible for sending the APV200-IP continuous commands (for continuous motion). One command only moves the camera in small increments.

### G. Tilt Down

Tilt local / remote camera down (depends on which FECC mode we are in)

```
name_string = "CameraTiltDown"
```

```
value_string = "" (IGNORED – Just put an empty string here)
```

Example GET: NONE

Example POST: "AT.CameraTiltDown=<CR><LF>"

NOTE:

The developer of the programmable device is responsible for sending the APV200-IP continuous commands (for continuous motion). One command only moves the camera in small increments.

### H. Get FECC Mode

Tells the programmable device whether we are in local or remote camera control mode

```
name_string = "CameraGetFeccMode"
```

Example GET: "AT.CameraGetFeccMode?<CR><LF>"

Example POST: NONE

NOTE:

This will tell the programmable device whether we are in local or remote camera control mode. This is useful for camera control as well as taking snapshots.

### I. Select Preset

Selects a preset (1-9) stored in the APV200-IP (local or remote depending on camera control)

```
name_string = "CameraSelectPreset"
```

value\_string = "X" (where X is a value 1 through 9 for preset select)

Example GET: NONE

Example POST: "AT.CameraSelectPreset=X<CR><LF>"

### **J. Save Preset**

Saves a preset (1-9) in the APV200-IP memory (local or remote depending on camera control)

name\_string = "CameraSavePreset"

value\_string = "X" (where X is a value 1 through 9 for preset save)

Example GET: NONE

Example POST: "AT.CameraSavePreset=X<CR><LF>"

## Chapter 10 General Information

There is some general information that can be retrieved from the APV200-IP. The parameters and values are described below.

### A. Software Version

This tells the programmable device the software version of the APV200-IP

name\_string = "SoftwareVersion"

value\_string = NONE (GET only)

Example GET: "AT. SoftwareVersion?<CR><LF>"

Example POST: NONE

### B. Serial Number

This tells the programmable device the serial number of the APV200-IP

name\_string = "SerialNumber"

value\_string = NONE (GET only)

Example GET: "AT. SerialNumber?<CR><LF>"

Example POST: NONE

### C. Remote Software Version

This tells the programmable device the software version of the remote APV200-IP. If not on a call, the string returned will be empty. This can only return the Software Version of a GENTNER product.

name\_string = "SoftwareVersionRem"

value\_string = NONE (GET only)

Example GET: "AT. SoftwareVersionRem?<CR><LF>"

Example POST: NONE

### D. Remote Serial Number

This tells the programmable device the serial number of the remote APV200-IP. If not on a call, the string returned will either be empty or the SN of the last system dialed. This can only return the Serial Number of a Gentner product.

name\_string = "SerialNumberRem"



value\_string = NONE (GET only)

Example GET: "AT. SerialNumberRem?<CR><LF>"

Example POST: NONE

### **E. Company Name**

This returns the name of the company. This will be either Gentner information, or specific OEM information.

name\_string = "CompanyName"

value\_string = NONE (GET only)

Example GET: "AT. CompanyName?<CR><LF>"

Example POST: NONE

### **F. Company Website**

This returns the website of the company. This will be either Gentner information, or specific OEM information.

name\_string = "CompanyWebsite"

value\_string = NONE (GET only)

Example GET: "AT. CompanyWebsite?<CR><LF>"

Example POST: NONE

### **G. Company Phone**

This returns the phone number of the company. This will be either Gentner information, or specific OEM information.

name\_string = "CompanyPhone"

value\_string = NONE (GET only)

Example GET: "AT. CompanyPhone?<CR><LF>"

Example POST: NONE

### **H. Company Tech Support**

This returns tech support information for the company. This can be a phone number, server information, or other information. This will be either Gentner information, or specific OEM information.

name\_string = "CompanyTechSupport"

value\_string = NONE (GET only)

Example GET: "AT. CompanyTechSupport?<CR><LF>"

Example POST: NONE

### I. Hardware Type

This returns the Hardware Type of the APV200-IP system. This can be either an ISDN type or a V.35 type.

Possible return values:

"ISDN (ST)"

"ISDN (U)"

"V.35 setup"

name\_string = "HardwareType"

value\_string = NONE (GET only)

Example GET: "AT. HardwareType?<CR><LF>"

Example POST: NONE

## Chapter 11 Controlling the Menu Interface

In some applications, the end user needs to be able to control the menu interface instead of just blindly doing actions. An example may be a user with a device such as a Crestron that can send commands to the APV200-IP, but has no screen for feedback. In these cases, it is imperative that the user see what is going on.

The API provides 2 ways to manipulate the APV200-IP menu system for these special cases:

### 1. Remote Control Mapping

Each remote control key on the APV200-IP remote has been mapped to a serial port command so that the user can easily go through the menu system as if he were using the remote control. If a developer wanted to make an application that had a design of the APV200-IP remote control, this would be the sort of application he would use.

### 2. Direct Menu Placement

For users who want to go to menus directly instead of having to go through them one at a time, the API provides direct menu placement of any menu on the APV200-IP software system. This system needs to be used in conjunction with the Remote Control Mapping system so the user can manipulate menu items once he has reached the menu of choice.

Parameters and values for both systems are described below.

#### A. Remote Control Mapping

The following is a list of keys that emulates the APV200-IP remote control. They are used conjunction with the following parameter/value set:

name\_string = "RemoteControlCode"

value\_string = CODE (where CODE is the remote control you wish to simulate)

Example GET: NONE

Example POST: "AT.RemoteControlCode=VOLUME\_MINUS<CR><LF>"

Remote Control	Value String
DIAL	"DIAL"
STAT	"STAT"
END	"END"
FECC	"FECC"
A/B	"CAM_AB"
PIP	"PIP"
- "VOLUME_MINUS"	

MUTE	"MUTE"
+	"VOLUME_PLUS"
PREV	"PREV"
ENTER	"ENTER"
UP ARROW	"UP"
DOWN ARROW	"DOWN"
LEFT ARROW	"LEFT"
RIGHT ARROW	"RIGHT"
CONFIG	"CONFIG"
1	"1"
2	"2"
3	"3"
4	"4"
5	"5"
6	"6"
7	"7"
8	"8"
9	"9"
STAR	"*"
0	"0"
POUND	"#"
TAKE	"TAKE"
SHOW	"SHOW"
VIDEO	"VIDEO"

### B. Direct Menu Placement

You can direct most menus directly by direct menu placement. Sending a POST to the APV200-IP with the "MenuSelect" parameter allows you to directly access the menu in the value field. The menu parameters and values are described below. Here is an example:

```
name_string = "MenuSelect"
```

```
value_string = MENU (where MENU is the menu you wish to select)
```

Example GET: NONE

```
Example POST: "AT.MenuSelect=MENU_MAIN<CR><LF>"
```

The above example selects the Main Menu. Here is the list of each menu's value.

Menu	Value String
None	"MENU_NONE"
Configure	"MENU_CONFIGURE"
Audio	"MENU_AUDIO"
Advanced Audio	"MENU_ADVANCED_AUDIO"

Video	"MENU_VIDEO"
ISDN	"MENU_ISDN"
Advanced ISDN	"MENU_ADVANCED_ISDN"
Line 1 Settings	"MENU_LINE1"
Line 2 Settings	"MENU_LINE2"
Line 3 Settings	"MENU_LINE3"
Data Ports	"MENU_DATA_PORTS"
Ethernet	"MENU_ETHERNET"
Security	"MENU_SECURITY"
Change Password	"MENU_CHANGE_PASSWORD"
Advanced	"MENU_ADVANCED"
Software Update	"MENU_SOFTWARE_UPDATE"
Manual Dial	"MENU_MANUALDIAL"
AP400 Status	"MENU_AP400_STATUS"
Phonebook	"MENU_PHONEBOOK"
Information	"MENU_INFORMATION"
Company Information	"MENU_COMPANY_INFORMATION"
System Information	"MENU_SYSTEM_INFORMATION"
Echo Cancel	"MENU_ECHO_CANCEL"
Camera Port	"MENU_MAIN_SERIAL_PORT"
Data Port	"MENU_AUX1_SERIAL_PORT"
Aux Data Port	"MENU_AUX2_SERIAL_PORT"
Edit Speed Dials	"MENU_EDIT_SPEED_DIAL"
Status	"MENU_CALL_STATUS"
Main	"MENU_MAIN"
Speed Dials	"MENU_SPEED_DIALS"

## Appendix A

### Wiring Status Codes

The wiring status of the ISDN lines can have the following states:

1. Starting

This state occurs when the ISDN lines have not yet completed initialization.

2. Ready

ISDN lines are plugged in, properly initialized, and ready for use.

3. Offline

The ISDN lines are set to "Offline" in the Line 2 or Line 3 settings.

4. Error

There is an error in the configuration of the ISDN line. Contact your ISDN phone provider for further assistance on configuring your ISDN line properly.

5. Bad SPID

The SPID number for that ISDN b-channel is typed incorrectly.

6. Wiring not activated

ISDN layer 1 is not activated properly. Either the line is plugged in incorrectly or it is not configured correctly. Contact your ISDN phone line provider for further assistance on configuring your ISDN line properly.

## Appendix B

### Channel Status Codes

The channel status of the ISDN b-channels can have the following states:

1. Reset

This state occurs when the ISDN lines are set to "disabled" in Line 2 or Line 2 settings.

2. Inactive

The ISDN lines are either not connected, or idle. Check your wiring status for more information.

3. Wait, proceeding

4. Proceeding

The call (either dialing or answering) is in the middle of negotiating with the remote side.

5. Need digits

The incorrect number of digits is being dialed. This can also happen if dialing to an international site where the switch doesn't know the number of digits that needed to be called.

6. Ringing local

This b-channel is ringing locally (answering).

7. Ringing distant

This b-channel is ringing on the remote side.

8. Busy

This b-channel is busy on the remote side.

9. Connected

This b-channel is connected properly.

## Appendix C

### Controlling the AP400

The Gentner AP400 can be controlled using API commands. This includes dialing and hanging up. The following are the commands necessary to dial and hangup the AP300 properly.

#### Configuring

In order for the dialing to work properly, you must have an AP400 connected (with appropriate cable) to either the 2nd or 3rd data port (data or aux data). Also, the data mode for that port must be set to "AP400." After setting this, reboot the APV200-IP system and the AP400 should be ready for use.

#### A. Dialing and Hanging Up

Dialing using the AP400 is just like dialing using ISDN. The only difference between the two is that to make an AP400 call, you need to set the call type to "Voice."

Call Type:        "AT.CallType=Voice<CR><LF>"

All other things for POSTing the dial are explained in Chapter 6 on "Dialing."

Hanging up the AP400 call is slightly different. Instead of doing this:

```
"AT.SubmitHangUp=<CR><LF>"
```

You need to do this instead:

```
"AT.SubmitHangUpAp400=<CR><LF>"
```

This will hang up the AP400 call instead of the ISDN call.

#### APV400 Call Status

There are two things you can receive from the APV200-IP for AP400 call status.

1. AP400 Connected (Yes or No)
2. AP400 Call Connected (Yes or No)

The following explains how to get this information



**AP400 Connected**

This tells the programmable device whether the AP400 is connected or not

name\_string = "AP400Connected"

value\_string = NONE (GET only)

Example GET: "AT.AP400Connected?<CR><LF>"

Example POST: NONE

If AP400 is connected,

"AP400Connected=Yes"

else

"AP400Connected=No"

**AP400 Call Connected**

This tells the programmable device whether the AP400 is on a call

name\_string = "AP400CallConnected"

value\_string = NONE (GET only)

Example GET: "AT.AP400CallConnected?<CR><LF>"

Example POST: NONE

If AP400 is on a call,

"AP400CallConnected=Yes"

else

"AP400CallConnected=No"