

VIDEO REQUEST

Communication Protocol Guide

Version 1.1.2

IR, RS-232, and Ethernet Interface Guide



Diagram illustrating the connection of a Sony DVP-CX777ES VCR to a Zektor HDS4.2 w/RS-232 and a Key Digital KD-SW4x1 Component Switcher.

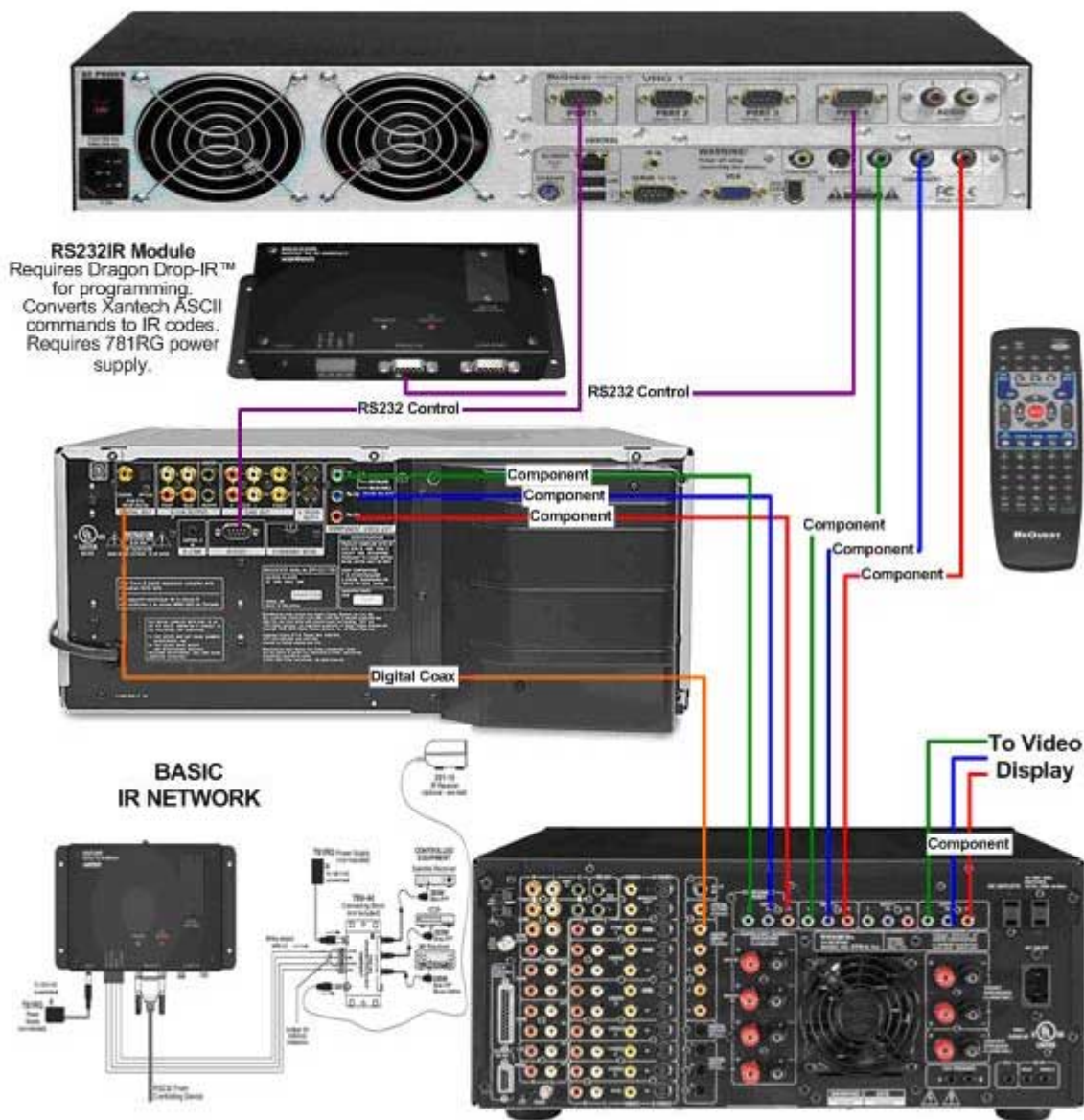
Connections:

- RS232 CONTROL:** Connects the Sony DVP-CX777ES to the Zektor HDS4.2 w/RS-232.
- Component Switcher Recommended:** Zektor HDS4.2 w/RS-232 (shown) and Key Digital KD-SW4x1.
- Output 1:** Connects the Zektor HDS4.2 w/RS-232 to the Sony DVP-CX777ES.
- Digital Coax:** Connects the Sony DVP-CX777ES to the Zektor HDS4.2 w/RS-232.
- Optical:** Connects the Sony DVP-CX777ES to the Zektor HDS4.2 w/RS-232.

The diagram shows the VCR's rear panel with various ports (RS232, GREE, BLUE, RED, Y OUT2, DIGITAL OUT, S-VIDEO OUT) and the switcher's front panel with corresponding inputs and outputs. A screenshot of the switcher's menu is also included.

1. Connect the VideoReQuest Video output to **Input 1** of the switch.
2. Connect the Audio and Video from DVD Changers 1-3 to **Inputs 2-4**.
3. Connect the RS-232 cable from the DVD changers to **Ports 1-3** of the VideoReQuest.
4. Connect **Port 4** of the VideoReQuest to the RS-232 Port of the switch.
5. Set the Video Switch setting on the VideoReQuest to the switch you have by going to **Menu→Configuration→Video Options→Video Switch** on the VideoReQuest GUI.

VIDEO REQUEST with XANTECH CONTROL



VideoReQuest with Xantech Control

1. Connect the VideoReQuest Video output to the receiver (Can be Composite, S-Video, or Component).
2. Connect the Audio and Video from DVD Changers 1-3 to **Inputs 2-4**.
3. Connect the RS-232 cable from the DVD changers to **Ports 1-3** of the VideoReQuest.
4. Connect **Port 4** of the VideoReQuest to the Xantech RS232IR module.
5. Set the Video Switch setting on the VideoReQuest to **Xantech IR Blaster** by going to **Menu→Configuration→Video Options→Video Switch** on the VideoReQuest GUI.
6. Connect the IR emitter or plug to the receiver.
7. Using Xantech Dragon Drop-IR software, load receiver codes. First code is for VideoReQuest. Code 2- 4 is for DVD changer 1-3.

VIDEO REQUEST PRONTO CONTROL



VideoReQuest with Pronto

1. Connect the VideoReQuest Audio and Video to the receiver (Can be Composite, S-Video, or Component).
2. Connect the Audio and Video from DVD Changers 1-3 to **Inputs 2-4**.
3. Connect the RS-232 cable from the DVD changers to **Ports 1-3** of the VideoReQuest.
4. Program the Pronto with the VideoReQuest sample program. Program the VRQ, DVD, Play, Stop, and Pause command to the appropriate receiver command.

TABLE OF CONTENTS

Table of Contents.....	I
Revision History.....	II
Introduction	1
Conventions Used	1
Part 1: IR Control.....	2
Rear IR Pinout	2
Learning IR Codes into Other Systems.....	2
List of Advanced IR Commands.....	2
Part 2: Serial/Ethernet Protocol	3
Rear Serial Port Pinout	3
Serial Cable Pinout.....	3
COM Settings	3
Using Ethernet	3
Communicating with the VideoReQuest	4
Sending Commands	4
Breakdown of Each Command Portion:	4
Marker:	4
Command Type:.....	4
Command Subtype:	4
Flags:	5
Data Size:	5
Header Check Sum:	5
Engine:	5
Mode:	5
Data:	5
Data Check Sum:	6
Initializing the Connection	6
Lines:	6
Calculating Checksum	6
Data Commands:.....	7
Examples of Strings to Send:	8
Subtype 01h – Cursor Up:	8
Subtype 02h – Letter 'A':	8
Subtype 03h – Number '3':.....	8
Subtype 04h – Requesting Player Detail "Genres":	8
Calculating the 4 Byte Number Argument:.....	8
Handling Feedback.....	9
Possible Subtypes:	9
Special Data Formats:	10
Appendix A – The GUI	13

REVISION HISTORY

Revision Number	Changes
1.1.2	Initial Version

INTRODUCTION

This document describes how to communicate with the VideoReQuest via IR, Serial, or Ethernet. For Serial and Ethernet, we will also explain how data feedback is formatted. Serial and Ethernet protocol is identical in terms of commands sent to the VideoReQuest and feedback received from the VideoReQuest. For the purpose of simplicity, Serial and Ethernet communication will be referred to as Serial in this document.

Conventions Used

- 1) Throughout this document, we will use the lowercase 'H' (*h*) notation when we wish to denote a Hexadecimal number. Multi-Byte strings will be written as "XX*h*, YY*h*, ...".

Examples:

- The Hex number 39 will be written as 39*h*, which is 57 in ASCII, or the number 9
 - The Hex string 4AC5 will be written as 4A*h*, C5*h*.
- 2) **VRQ** and **VideoReQuest** will be used synonymously to describe the VideoReQuest line of products.

PART 1: IR CONTROL

There are 2 ways to control the VideoReQuest via IR. The simplest method is to use a handheld remote aimed at the IR receiver on the front of the unit. For more advanced installations, there is a 1/8th inch mini jack on the rear of the unit for integration with IR based control systems. No matter which method you choose, this portion of the document will describe the commands available.

Rear IR Pinout

The rear IR jack is designed for a 2 conductor 1/8th inch mini plug from a powered connecting block or control system. You CAN NOT plug a dinkylink directly into the back of the VideoReQuest.

2 Conductor Pinout	
Tip	Signal
Ring	Ground
Sleeve	Not Connected

Learning IR Codes into Other Systems

For any advanced commands (2 button key combinations), the proper method for teaching the system the VideoReQuest IR codes is as follows:

1. Aim the VideoReQuest Remote **AWAY** from the learner
2. Press **AND HOLD** the **Alt** or **Shift** button (depending on command)
3. Aim the remote **AT** the learner
4. Press the second button
5. Release **BOTH** buttons

Most advanced commands are written on the IR remote in GREEN text. To access these commands, press and hold the **Alt** button prior to pressing the desired command button. Other advanced commands not written on the remote are listed in the table below.

List of Advanced IR Commands

Action	Description	Alt/Shift	2 nd Button
Refresh DVD Details	Performs a lookup on the current DVD	ALT	EDIT

PART 2: SERIAL/ETHERNET PROTOCOL

This section describes how to communicate with the VideoReQuest via Ethernet or Serial. Ethernet and Serial protocols are identical in terms of sending commands and receiving feedback. For simplicity, we will refer to Ethernet and Serial communication as Serial, though it can be used for Ethernet as well.

Rear Serial Port Pinout

Pin	Purpose
1	Not Used
2	Receive
3	Transmit
4	Not Used
5	Ground
6	Not Used
7	RTS
8	CTS
9	Not Used

Serial Cable Pinout

Pin (VRQ Side)	Pin (Control Side)
2	3
3	2
5	5
7	8
8	7

COM Settings

The **rear serial port** is used for integrating the VideoReQuest with a control system. These settings can be found by going to **Menu→Configuration→Control Port** on the VideoReQuest.

Parameter	Value	Value	Value	Value
Baud	9600	19200	38400	57600
Data Bits	8	8	8	8
Stop Bits	1	1	1	1
Parity	None	None	None	None
Flow Control	None	HW	HW	HW

Using Ethernet

The default port used to communicate with the VideoReQuest via Ethernet is port **3663**. If you open a connection to the VideoReQuest's IP address at port 3663, you can communicate with the unit just as if you connected serially.

Communicating with the VideoReQuest

Before you can communicate with the VideoReQuest, you must initialize the connection. A "Start Communications" command is similar to all other commands, with a few exceptions. See **Initializing the Connection** on page 6 for more information.

Sending Commands

Commands are sent to the VideoReQuest in strings of Hexadecimal (or Hex) numbers. Depending on the command, these strings are anywhere from 13 bytes to 17 or more bytes.

All commands are in the following format:

HEADER									DATA			
Marker		Command Type	Command Subtype	Flags		Data Size		Chk Sum	Engine	Mode	Data	Chk Sum
EC _h	A0 _h	0A _h	1 Byte	00 _h	00 _h	Byte 1	Byte 2	00 _h	FF _h	FF _h	1+ Bytes	00 _h

Breakdown of Each Command Portion:

Marker:

The **Marker** bytes are ALWAYS **FCh**, **A0h**. These bytes tell the VideoReQuest the data that follows is valid.

Command Type:

The **Command Type** will almost always be **0Ah** when sending commands to the VideoReQuest. All possible command types are as follows:

Command Type	Description
05h	Start Communications (see Initializing the Connection on page 6 for more information)
0Ah	Command (most commands sent to the VRQ)
0Fh	Data (not typically used)
14h	Acknowledgement (used only if ACK flag is turned ON)
19h	End Communications (used to terminate connection)

Command Subtype:

The 4 possible subtypes are as follows:

Command Subtype	Description	Data Length (bytes)*
01h	1 byte command, no arguments	4
02h	2 bytes (1 for command, 1 for character argument)	5
03h	5 bytes (1 for command, 4 for integer argument)	8
04h	Multi-byte string argument	TBD

* The data length includes the **Engine**, **Mode**, and **Data Checksum** bytes.

Flags:

The 2 flag bytes are used to tell the VideoReQuest whether you do or do not want to use certain features. Setting both to 00h turns off all options. Each byte is broken up into 8 bits. Each bit is a flag for a certain feature. The possible options are as follows:

Byte 1								Byte 2							
Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
														ACK	Chk Sum

Example:

To turn on the use of Check Sums, but not Acknowledgments, send **00h, 01h** for the flag bytes. This should be done on initialization of the connection as well as for any subsequent command sent.

Data Size:

2 byte value that contains the length (in bytes) of the data portion of the command string. For **Command Subtypes 1, 2, or 3**, use the **Subtype** table on page 4 to determine the data length. For **Command Subtype 4**, the following example illustrates how to calculate the **Data Size**:

Assume the length of the string is 25 bytes (25 characters, including spaces and punctuation). This means that the full **Data Size** is 25 bytes PLUS 3 bytes for the **Engine, Mode, and Checksum**, or 28 bytes total.

$$\text{BYTE 1} = \frac{28}{255} = 0, \text{ or } 00h$$

$$\text{BYTE 2} = 28 \text{ MOD } 255 = 28, \text{ or } 1Ch$$

Header Check Sum:

If you are not using Check Sum (based on your Flag settings), then set this byte to **00h**. Otherwise, see **Calculating Checksum** on page 6 for more information.

Engine:

The Engine is simply the page that the VideoReQuest is currently on. Setting the Engine byte to **FFh** sends the command to whatever the current Engine is. Setting the Engine byte to a specific engine (see table) sends the command to that engine ONLY.

Engine	Description
00h	Sends command to Browse mode
01h	Sends command to Player mode
02h	Sends command to DVD player
FFh	Sends command to default (current) engine

Mode:

The Mode is reserved for future use, and should always be set to **FFh**.

Data:

These byte(s) are the actual command and possible arguments that you are sending to the VideoReQuest. See **Data Commands** on the next page for a complete list of commands.

Data Check Sum:

If you are not using Check Sum (based on your Flag settings), then set this byte to **00h**. Otherwise, see **Calculating Checksum** below for more information.

Initializing the Connection

Before the VideoReQuest will respond to commands OR send you feedback, you must initialize the connection. This tells the VideoReQuest a number of things, including whether or not you wish to calculate Check Sum, how many lines of feedback you want at a time, etc. The format for a "Start Communications" command is as follows:

HEADER									DATA		
Marker		Command Type	Command Subtype	Flags		Data Size		Chk Sum	Cmd	Lines	Chk Sum
FCh	A0h	05h	01h	Byte 1	Byte 2	00h	03h	00h	01h	06h	00h

Currently, the only variable in this string is **Flags**, which you can set to whatever you want depending on whether or not you want to use Acknowledgements or Check Sum (see **Flags** on the previous page for more information). To simplify programming, it is recommended that you use **00h, 00h** for the flag bytes, which turns OFF both Acknowledgements and Check Sum calculation.

Lines:

The **Lines** portion of the data section is a 1 byte value that tells the VideoReQuest how many lines of text you want to receive at once on the Browse and Chapters pages. Currently, the only supported value is **06h**.

Calculating Checksum

The Checksum can be calculated as follows:

1. Take the value of each byte in the string, up to but NOT including the checksum, and multiply it by the position in the string from left to right (the leftmost byte is byte 1)
2. Add all the values calculated in step 1
3. Convert the result to Hex
4. The rightmost Hex byte is the checksum

This is done the same way for both the Header and the Data checksums.

Example Header Checksum:

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9
Incoming String	FC	A0	05	01	00	00	00	03	67
Integer value	252	160	5	1	0	0	0	3	n/a
Position Multiplier	1	2	3	4	5	6	7	8	9
Result	252	320	15	4	0	0	0	24	n/a

Sum of all Results = 252+320+15+4+0+0+0+0+24 = 615 = **02h, 67h**

The *rightmost* byte of the result is **67h**, which is the calculated checksum. As we can see, that matches the incoming checksum, so no data was lost.

Data Commands:

Command	Hex String	Description	Cmd Subtype
Cursor Down	00h	Moves cursor down 1 line at a time	01h
Cursor Left	01h	Moves cursor left or moves back 1 level on Browse page. Acts as Rewind in DVD mode.	01h
Cursor Right	02h	Moves cursor right or moves down 1 level on Browse page. Acts as Fast Forward in DVD mode.	01h
Cursor Up	03h	Moves cursor up 1 line at a time	01h
Refresh	04h	Requests all data from VideoReQuest	01h
Page Up	05h	Moves up 6 lines at a time	01h
Page Down	06h	Moves down 6 lines at a time	01h
All Movies	07h	Jumps to All Movies list	01h
Genres	08h	Jumps to Genres list	01h
Ratings	09h	Jumps to Ratings list	01h
Now Playing	0Ah	Jumps to Player, displays currently playing movie info	01h
Enter	0Bh	Issues Enter command	01h
Home	0Ch	Returns to top of Browse page (Now Playing, All Movies, etc)	01h
VRQ Mode	0Dh	When in DVD mode, returns to VRQ mode	01h
Edit	0Eh	Not currently supported	01h
Eject	0Fh	Not currently supported	01h
Power Toggle	10h	Toggles soft power ON/OFF	01h
Power ON	11h	If unit is soft powered OFF, turns unit ON	01h
Power OFF	12h	If unit is soft powered ON, turns unit OFF	01h
Number	13h	Sends the 4 byte number in the argument portion	03h
DVD Menu	14h	Not currently supported	01h
Play	15h	Plays the current selection or unpauses playback	01h
Pause Toggle	16h	Toggles playback between Pause and Unpause	01h
Pause ON	17h	Pauses playback	01h
Pause OFF	18h	Resumes playback from a Paused state	01h
Stop	19h	Stops playback, returns to VRQ mode	01h
VRQ Menu	1Ah	Not currently supported	01h
DVD Mode	1Bh	When in VRQ mode, switches to DVD mode	01h
Next Chapter	1Ch	Plays the next chapter	01h
Previous Chapter	1Dh	Resumes current chapter or plays previous chapter	01h
Goto Top	1Eh	Jumps to top of current list in Browse page	01h
Goto Bottom	1Fh	Jumps to bottom of current list in Browse page	01h
Search	20h	Not currently supported	01h
Move to Line	21h	Moves to the line number indicated by 4 byte integer argument	03h
Letter	22h	Sends the letter indicated by the 1 byte character argument	02h
Backspace	23h	Deletes the last character entered in text entry	01h
Audio	24h	Toggles the Audio options in DVD playback	01h
Subtitles	25h	Toggles subtitle options in DVD playback	01h
Angle	26h	Toggles angle options in DVD playback on enabled DVDs	01h
View	27h	Not currently supported	01h
Queue	28h	Not currently supported	01h
Repeat	29h	Not currently supported	01h
Shuffle	2Ah	Not currently supported	01h
Enter Line	2Bh	Issues an Enter command on the line number indicated by the 4 byte integer argument	03h
Themes	2Ch	Not currently supported	01h
Info	2Dh	Not currently supported	01h
Directors	2Eh	Jumps to Directors list	01h
Actors	2Fh	Jumps to Actors list	01h
Mode	30h	Not currently supported	01h
Player Detail Request	37h	Requests a certain Player Detail, based on string argument	04h

Examples of Strings to Send:

Subtype 01h – Cursor Up:

The command for Cursor Up is **03h**. Send the following:

Marker		Cmd Type	Sub type	Flags		Data Size		Chk Sum	Engine	Mode	Data	Chk Sum
FCh	A0h	0Ah	01h	00h	00h	00h	04h	00h	FFh	FFh	03h	00h

Subtype 02h – Letter 'A':

The command for any Letter is **22h**. Send the following:

Marker		Cmd Type	Sub type	Flags		Data Size		Chk Sum	Engine	Mode	Data		Chk Sum
FCh	A0h	0Ah	02h	00h	00h	00h	05h	00h	FFh	FFh	22h	41h	00h

Subtype 03h – Number '3':

The command for any Number is **13h**. Send the following:

Marker		Cmd Type	Sub type	Flags		Data Size		Chk Sum	Engine	Mode	Data*					Chk Sum
FCh	A0h	0Ah	03h	00h	00h	00h	08h	00h	FFh	FFh	13h	00h	00h	00h	03h	00h

* See below for more information on calculating the 4 byte number in the Data Argument

Subtype 04h – Requesting Player Detail "Genres":

The command for Player Detail Request is **37h**. Send the following:

Marker		Cmd Type	Sub type	Flags		Data Size**		Chk Sum	Engine	Mode	Data		Chk Sum
FCh	A0h	0Ah	03h	00h	00h	00h	0Ah	00h	FFh	FFh	37h	Genres	00h

** Data Size for Subtype **04h** is a variable that depends on the length of the argument

Calculating the 4 Byte Number Argument:

- 1) Assume that the number you wish to send is an ASCII value, NOT the actual number
- 2) Determine the Hex value of the number you wish to send (see table below for examples)
 - a. For example, the number 3 would be **03h**, NOT **33h**
 - b. If the number is less than 4 bytes (i.e. less than 4311810304), append **00h** to the beginning of the number to complete the 4 byte requirement

Num	Hex
0	00
1	01
2	02
3	03
4	04
5	05
6	06
7	07
8	08
9	09
10	0A
11	0B
12	0C
13	0D
14	0E
15	0F

Num	Hex
16	10
17	11
18	12
19	13
20	14
21	15
22	16
23	17
24	18
25	19
26	1A
27	1B
28	1C
29	1D
30	1E
31	1F

Num	Hex
32	20
33	21
34	22
35	23
36	24
37	25
38	26
39	27
40	28
41	29
42	2A
43	2B
44	2C
45	2D
46	2E
47	2F

Num	Hex
48	30
49	31
50	32
51	33
52	34
53	35
54	36
55	37
56	38
57	39
58	3A
59	3B
60	3C
61	3D
62	3E
63	3F

Num	Hex
64	40
65	41
66	42
67	43
68	44
69	45
70	46
71	47
72	48
73	49
74	4A
75	4B
76	4C
77	4D
78	4E
79	4F

Num	Hex
79	50
80	51
81	52
82	53
83	54
84	55
85	56
86	57
87	58
88	59
89	5A
90	5B
91	5C
92	5D
93	5E
94	5F

Before the VideoReQuest will send feedback data, you must send a command to the VideoReQuest that initializes communication. See **Initializing the Connection** on Page 6 for more information.

HEADER									DATA	
Marker		Command Type	Command Subtype	Flags		Data Size		Chk Sum	Data	Chk Sum
FC _h	A0 _h	0A _h	1 Byte	00 _h	00 _h	Byte 1	Byte 2	00 _h	1+ Bytes	00 _h

For help understanding what each **Subtype** is, see **Appendix A**.

Hex	Integer	Description	Data Size
80	128	Browse window title ("Welcome", "Total: 46", etc)	TBD
81	129	Browse List Title (Home, Genres, etc)*	TBD
82	130	Browse Column 1 Header (Total)	TBD
83	131	Browse Column 2 Header (Genre, Rating, etc)	TBD
84	132	Browse Column 1 Data	TBD
85	133	Browse Column 2 Data	TBD
86	134	Browse List Size, Page UP/DOWN Flags and Cursor Position*	8
8A	138	Browse Long Description	TBD
8B	139	Engine Mode 0: On Browse Page 1: On Player Page 2: DVD Pass-thru (Playing a movie)	2
8C	140	View Mode, Changer Number, Slot Number*	5
90	144	Browse Text Line 1*	TBD
91	145	Browse Text Line 2*	TBD
...
9F	159	Browse Text Line 16*	TBD
A0	160	Player Long Description (not used)	TBD
A1	161	Player List Size, Page UP/DOWN Flags and Cursor Position (which chapter is selected)*	8
A2	162	Player Movie Title	TBD
A3	163	Player State 0: Stopped 1: Playing 2: Paused	2
A5	165	Player Window Title ("More Info", "Chapters")	TBD
A6	166	Player Detail Headers*	TBD
A7	167	Player Detail Text*	TBD
A8	168	Player Mode (on Chapters or on Details) 0: Details 1: Chapters	2
A9	169	Player List Size, Page UP/DOWN Flags and Now Playing Cursor Position (which chapter is playing)*	8
B0	176	Player Chapter Line 1 Text	TBD
B1	177	Player Chapter Line 2 Text	TBD
...
BF	191	Player Chapter Line 16 Text	TBD

- 9 -

Example:

Let's say you received the following string from the VideoReQuest:

FCh, A0h, 0Ah, 84h, 00h, 00h, 00h, 07h, 00h, Horror, 00h

We know from the format explained on the previous page that **84h** is the **Command Subtype**. From the table of subtypes on the previous page, we know that **84h** indicates that you are receiving the **Browse Column 1 Data**. Therefore, we extract **Horror** from the data portion and save that in our **Browse Column 1 Data** variable.

Special Data Formats:
Browse List Title:

The **Browse List Title** is sent in chunks, delimited by a NULL character (**00h**). For example, if you are inside the Genres list, the data portion of the string sent by the VideoReQuest will look like this: **Home[00h]Genres[00h]**. This allows you to use just the final portion to show the current list title, or the entire thing to show the full hierarchy.

List Size, Page UP/DOWN Flags, and Cursor Position:

The **List Size**, **Page UP/DOWN Flags** and **Cursor Position** for the Browse and Player pages are sent by the VideoReQuest in 1 string. The first 4 bytes of the string are the **List Size**, the next byte is the **Page UP/DOWN Flag**, and the final 2 bytes are for the **Cursor Position**.

List size is calculated as follows:

$$256^3 \times \text{BYTE1} + 256^2 \times \text{BYTE2} + 256 \times \text{BYTE3} + \text{BYTE4}$$

Page UP/DOWN flags could be one of the following:

Value	Description
00h	No more lines exist above or below the current view
01h	More lines exist below the current view
02h	More lines exist above the current view
03h	More lines exist above AND below the current view

The 2 bytes for **Cursor Position** are calculated as follows:

$$256 \times \text{BYTE1} + \text{BYTE2} = \text{RESULT}$$

The following values of **RESULT** are possible:

Value	Cursor on Line
1	1
2	2
4	3
8	4
16	5
32	6
64	7
128	8

Value	Cursor on Line
256	1
512	2
1024	3
2048	4
4096	5
8192	6
16384	7
32768	8

View Mode, Changer Number, and Slot Number:

The **View Mode**, **Changer Number**, and **Slot Number** are sent by the VideoReQuest in 1 string. The first byte is the **View Mode**, followed by 1 byte for the **Changer Number** (1, 2, 3, or 4), followed by 2 bytes for the **Slot Number** (1-400).

Possible **View Modes** are as follows:







View Mode	Description
1	In VRQ Mode. Should display VRQ video output.
2	In DVD Mode. Should display DVD video output.

Slot Number is calculated in the same manner as **Cursor Position**:

$$256 \times \text{BYTE1} + \text{BYTE2} = \text{Slot Number}$$

Browse Line Text:

The first byte of the **Browse Line Text** is a 1 byte flag indicating which icon should be displayed on that line. Possible values are as follows:

Value	Description	Icon
00h	No Icon	N/A
01h	Now Playing	
02h	All Movies	
03h	Genres	
04h	Ratings	
05h	Actors	
06h	Directors	

Player Detail Headers:

The **Player Detail Headers** are sent by the VideoReQuest in 1 string, separated by the NULL character (**00h**). The data portion of the string would look similar to the following:

Genres[00h]Cast[00h]Directors[00h]Plot Summary[00h]

Player Detail Text:

The VideoReQuest will not send any **Player Detail Text** until it is requested to do so. Therefore, when you receive the **Player Detail Headers**, you should send one or more of the following commands to receive the **Player Detail Text**:

Detail	Cmd Subtype	Data Length	Data Command	Data Argument
GENRES	04h	10	37h	Genres
CAST	04h	8	37h	Cast
DIRECTORS	04h	13	37h	Directors
PLOT SUMMARY	04h	16	37h	Plot Summary

For example, to request the **Genres**, send the following:

FCh, A0h, 0Ah, 04h, 00h, 00h, 00h, 0Ah, 00h, FFh, FFh, 37h, Genres, 00h

The VideoReQuest will respond by sending you the **Header** you requested, followed by the NULL character (**00h**), followed by the **Detail** for that header.

For example, if you requested the **Genres**, the VideoReQuest will respond with:

FCh, A0h, 0Ah, A7h, 00h, 00h, 00h, 0Eh, 00h, Genres, 00h, Horror, 00h

Appendix A – The GUI

This section shows what each portion of feedback is and where it is located on the TV Out GUI.



Figure 1 – The Browse Page



Figure 2 – The Chapters Page

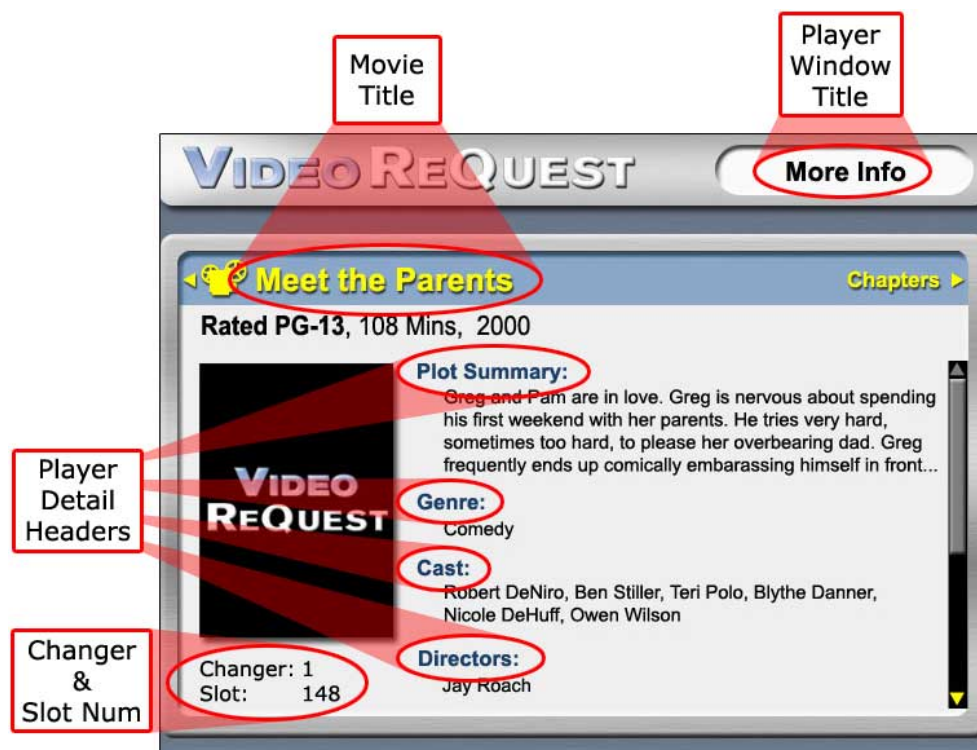


Figure 3 – The Player Details Page