ADTEC PRODUCTIONS, INC.

SOLOIST CONTROL INTERFACE

PRELIMINARY SPECIFICATIONS

Version 7.5

March 10,1997

2231-1 Corporate Square Blvd. Jacksonville, FL 32216 1-904-720-2003

NOTICE TO USER

DISCLAIMER

This document is not currently complete (March 10,1997) and is an advance copy .

OVERVIEW

The Soloist Professional MPEG Player is a standalone digital video player capable of playing MPEG1 video from a variety of disc mediums, including CD-ROM, removable hard drives, and fixed hard drives (both internal and docking station varieties). As a standalone product, all necessary hardware and firmware are self contained, eliminating the need for either a PC or PC Software for functionality. To increase usability, a serial interface and protocol has been developed to allow for sophisticated external control. It is this interface and protocol which will be covered in this document.

FEATURES

Features of the Soloist Professional MPEG Player include:

- **Guaranteed uninterrupted MPEG1 program stream playback at up to 4Mbs**
- **Line doubling decoding, converting SIF 352*240 into 704*480 effective resolution**
- **4 IDE ATA/ATAPI drive interface ports (up to 4 master mode drives)**
- **1 SCSI drive interface port (up to 7 SCSI drives)**
- **31 Mbit video buffer (approximately 10 seconds @ 3Mbps)**
- **1 RS-232 control port with loop through capability**
- **1 RS-422 control port**
- **1 4 bit parallel control port**
- **2 composite video outputs**
- **1 sync input (vertical synchronization with an external source)**
- **CD quality (44.1 Khz) stereo audio outputs**
- **OPTION- S Video output and balanced audio outputs**
- **OPTION- Real Time Clock**

CONTROL INTERFACE

There are 3 control ports on the Soloist: 1) 4 bit unidirectional parallel control, 2) Full duplex loop through RS-232 serial control, and 3) Full duplex RS-422 serial control. Each type will be described below.

4 bit unidirectional parallel control-

Allows for control using legacy equipment, i.e. Lite-Ning and Active. This interface is not supported for third party development or any new applications. It will not be discussed any further in this document.

Full Duplex loop through RS-232 serial control-
Allows for control using a single standard RS-232 serial port at baud rates up to 38400. Loop through allows for an almost unlimited number of Soloists to be connected to a single host port. Addressability is a feature of the command protocol.

Full Duplex RS-422 serial control-
Allows for control using a single standard RS-422 serial port at baud rates up to 38400. Loop through is not supported, so a control relationship using this port is strictly one to one.

The recommended control port for third party development is the RS-232 serial control port, since nearly every PC has an RS-232 serial port and loop through control is available. If a one to one control situation in an electrically noisy environment exists, it may be desirable to use the RS-422 serial control port. Either way, both ports support identical baud rates and control protocols.

CONTROL PROTOCOL

All control is achieved using a documented protocol. This protocol is ASCII based and is designed to allow for simple interface in a terminal like manner. This protocol is recommended in any direct connection applications, since it is fast, simple and allows for minimal error detection and recovery, which is commensurate with most direct connection operating environments. In addition, a secure protocol extension is available which packetizes each command line with crc-16 values and stop text markers. This protocol is recommended in any telephony applications, since it allows easy error detection and retransmission of corrupted data, which is commensurate with the noisy nature of telephony. These two protocols will herein be referred to as "Terminal" and "Packetized" respectively. The Soloist is capable of automatically detecting either protocol and responding appropriately, on a command by command basis.

TERMINAL PROTOCOL

Terminal protocol is designed to allow the Soloist to be connected directly to a terminal and be controlled by simple ASCII text commands. These commands are of the form Name, Command, and any necessary Arguments.

<NAME><SEP>**<COMMAND>**<SEP>**[ARGUMENT(S)]** [SEP]**<CR>**[LF]

<>= Required, []= Optional

where

NAME = Soloist name, up to 20 ASCII characters. Any non-white ASCII characters are valid.

SEP = Separator characters, SPACE (" "), COMMA (","), or COLON ("

COMMAND = Command, ASCII text in either verbose or terse form. Verbose form is the command in complete word form- i.e. "PLAY", "STOP", "PLAYSPOT", etc. Terse form is the command in abbreviated form- i.e. "PL", "ST", "PS", etc. Whichever form the command uses, any responses will be in a like form. Generally, verbose form is commonly used for manual terminal control, whereas programmatic control may find terse mode more suitable.

ARGUMENTS = Argument list, ASCII text describing any arguments in either verbose or terse form. Valid arguments will be listed with each command's description.

CR = ASCII Carriage Return character, 0x0D.

LF = ASCII Line Feed character, 0x0A.

Depending on the command, the Soloist will respond with an appropriate ASCII message. These messages will be explained under each command's description.

PACKETIZED PROTOCOL

Packetized protocol is designed to allow the Soloist to be connected remotely through a telephony link and be controlled by simple ASCII text commands. These commands are of the form Name, Command, any necessary Arguments, Cyclic Redundancy Code, and End Text.

<NAME><SEP>**<COMMAND><**SEP>**[ARGUMENT(S)]** <SEP>**<CRC-16><ETX>**

<>= Required, []= Optional

where

NAME = Soloist name, up to 20 ASCII characters. Any non-white ASCII characters are valid.

SEP = Separator characters, SPACE (" "), COMMA (","), or COLON ("

COMMAND = Command, ASCII text in either verbose or terse form. Verbose form is the command in complete word form- i.e. "PLAY", "STOP", "PLAYSPOT", etc. Terse form is the command in abbreviated form- i.e. "PL", "ST", "PS", etc. Whichever form the command uses, any responses will be in a like form. Generally, verbose form is commonly used for manual terminal control, whereas programmatic control may find terse mode more suitable.

ARGUMENTS = Argument list, ASCII text describing any arguments in either verbose or terse form. Valid arguments will be listed with each command's description.

CRC-16 = 16 bit cyclic redundancy code, generated from all prior ASCII characters and sent as ASCII representation of the 16 bit value in decimal format.

ETX = ASCII End Text character, 0x03.

Upon receipt of a packetized command, the Soloist will validate reception of the command by parsing off the CRC-16 portion of the command and seeing if it confirms the validity of the transmitted text. If it is validated, the Soloist will immediately respond by sending the <ACK> character (0x06) and processing the command in the exact same manner, with the exact same responses, as if it had been issued using the terminal protocol. If it is not validated, the Soloist will immediately respond by sending the <NAK> character (0x15) and discard the entire command. It would then be possible for the host to try and re-transmit the packetized command or abandon the communiqué. Notice that the packetized protocol is the same form as the terminal protocol, with the exception that the <CR> terminator is replaced with the CRC-16 text and the <ETX> character and an <ACK> character precedes the normal Soloist responses.

If packetized protocol is used, the Soloist will respond in kind (using packetized responses). If terminal protocol is used, the Soloist will respond in kind (using terminal responses).

NOTE: Comands can be concatenated on a single line by using the PIPE character (|) as a field seperator. For example, a command line could be:

<NAME> LIST CLEAR|LIST ADD "NESTLE"|PLAY

This command would clear out the list, then add the spot named “NESTLE” to the list, and finally begin playing, all with one command line.

COMMANDS

PLAY

Description- Plays the current spot. If the playlist has any entries, the current spot will be from the list. If the playlist is empty, all available inventory will be treated as the playlist. If there is any inventory in the Soloist, a spot is guaranteed to be played. You may optionally argue how many consecutive spots you would like to play.

Relevant Flags- Repeat, Errorlimit
Required Arguments- None
Optional Arguments- Number of consecutive spots to play, 1 - 2,147,483,648
Verbose Example- Command- `<NAME> PLAY`
Response- `OK`
Terse Example- Command- `<NAME> PL 4`
Response- `0`
Errors- (Terse, Verbose) `1, ERROR- No spots, unit is empty`

STOP

Description- Stops the currently playing spot and cues up the next spot.

Relevant Flags- Blank
Required Arguments- None
Optional Arguments- None
Verbose Example- Command- `<NAME> STOP`
Response- `OK`
Terse Example- Command- `<NAME> ST`
Response- `0`
Errors- (Terse, Verbose) `1, ERROR- No spots, unit is empty`

NEXT

Description- Go to the next spot, rolling over if at the end of the list. If the playlist has any entries, the next spot will be from the list. If the playlist is empty, all available inventory will be treated as the playlist. If there is any inventory in the Soloist, a next spot is guaranteed. If the Soloist is currently stopped, it will remain stopped. Likewise, it will continue playing if currently playing.

Relevant Flags- Repeat, Shuffle
Required Arguments- None
Optional Arguments- None
Verbose Example- Command- `<NAME> NEXT`
Response- `OK`
Terse Example- Command- `<NAME> NE`
Response- `0`
Errors- (Terse, Verbose) `1, ERROR- No spots, unit is empty`

PREVIOUS

Description- Go to the previous spot, rolling under if at the beginning of the list. If the playlist has any entries, the previous spot will be from the list. If the playlist is empty, all available inventory will be treated as the playlist. If there is any inventory in the Soloist, a previous spot is guaranteed. If the Soloist is currently stopped, it will remain stopped. Likewise, it will continue playing if currently playing.

Relevant Flags- Repeat, Shuffle
Required Arguments- None
Optional Arguments- None
Verbose Example- Command- <NAME> PREVIOUS
Response- OK
Terse Example- Command- <NAME> PR
Response- 0
Errors- (Terse, Verbose) 1, ERROR- No spots, unit is empty

PAUSE

Description- Pause the currently playing spot, if any.
Relevant Flags- None
Required Arguments- None
Optional Arguments- None
Verbose Example- Command- <NAME> PAUSE
Response- OK
Terse Example- Command- <NAME> PA
Response- 0
Errors- (Terse, Verbose) 1, ERROR- No spots, unit is empty

REWIND

Description- Rewind the current spot. If a spot is playing, go back to the beginning of that spot and play it again. If no spot is playing, go back to the beginning of the list. If there is no playlist, go back to the first spot in the system.
Relevant Flags- Blank
Required Arguments- None
Optional Arguments- None
Verbose Example- Command- <NAME> REWIND
Response- OK
Terse Example- Command- <NAME> RW
Response- 0
Errors- (Terse, Verbose) 1, ERROR- No spots, unit is empty

CUESPOT

Description- Cue up a spot and get it ready to play. You must specify the spot name, or you may optionally specify the full drive, disc, and spot name or spot number.
Relevant Flags- None
Required Arguments- Spot Name
Optional Arguments- Drive Number, Disc Number, Spot Name or Spot Number
Verbose Example- Command- <NAME> CUESPOT 0 0 “NESTLE”
Response- OK
Response- ERROR- Spot does not exist
Terse Example- Command- <NAME> CS “NESTLE”
Response- 0
Errors- (Terse, Verbose) 1, ERROR- No spots, unit is empty
2, ERROR- Bad Arguments
3, ERROR- Spot does not exist
4, ERROR- Transport in use,no drive access allowed

5, ERROR- Playlist has entries, no cueing allowed

PLAYSPOT

Description- Play a specified spot. You must specify the spot name, or you may optionally specify the full drive, disc, and spot name or spot number.

Relevant Flags- None

Required Arguments- Spot Name

Optional Arguments- Drive Number, Disc Number, Spot Name or Spot Number

Verbose Example-
Command- <NAME> PLAYSPOT 0 0 "NESTLE"
Response- OK

Terse Example-
Command- <NAME> PS "NESTLE"
Response- 0

Errors- (Terse, Verbose)
1, ERROR- No spots, unit is empty
2, ERROR- Bad Arguments
3, ERROR- Spot does not exist

TRANSPORT

Description- Displays the current transport status. The response format is: Status, DriveNumber, DiscNumber, Spot Number, SpotName, Size, Length (Hr:Min:Sec.Frame), BitRate, Date, TimeStamp (Hr:Min), TimeCode (Hr:Min:Sec.Frame), PercentCompleted.

Relevant Flags- None

Required Arguments- None

Optional Arguments- None

Verbose Example-
Command- <NAME> TRANSPORT
Response- OK
IDLING 12, 0, 0 NESTLE 11.286 Mbytes 0:00:30.01 3.0 Mbps 8/08/96 6:20 0:00:00.00 0.00%

Terse Example-
Command- <NAME> TR
Response- 0
01 12 00 000 NESTLE 0011286 00 00 30 01 030 08 08 96 06 20 00 00 00 00 00 00

Status Codes - (Terse, Verbose)

0, UNIT NOT READY - No disc(s) are present, there are no spots available.
1, IDLING - Unit is not currently playing and the video buffer is full.
2, IDLE CUEING - Unit is not currently playing and is currently cueing the next spot.
3, IDLE NEXT - Unit is not playing, has completely buffered next spot, and is cueing up the following spot
4, STOPPING - Unit is stopping
5, PLAYING - Unit is playing
6, BUFFERED - Present playing spot is completely loaded into the video buffer
7, WARNING - Read error occured and Errorlimit threshold is checked.
8, FATAL - Errorlimit has been exceeded, play is abandoned.
9, NEXT - Next spot is currently being fetched.
10, PREVIOUS - Previous spot is currently being fetched.
11, PAUSED - Unit is paused
12, PAUSE CUEING - Unit is paused and the and is currently cueing the next spot.
13, PAUSE NEXT - Unit is paused, has completely buffered next spot, and is cueing up the following spot
14, REWINDING - Unit is rewinding

INDEX

Description- Jump forward or backwards from the current position of a spot by an absolute or relative amount. Plus (+) and Minus (-) signs indicate a relative offset from the current point, while unsigned values specify an absolute position within the spot. If a value is entered that is outside the bounds of the spot, the minimum or maximum value is used, respectively.

Relevant Flags- None

Required Arguments- FRAME
Optional Arguments- [+] (HR:MIN:SEC:FRAME), [-] (HR:MIN:SEC:FRAME)
Verbose Example- Command- <NAME> INDEX + 00:00:02.00 (jump forward 2 sec)
Response- OK
Verbose Example- Command- <NAME> INDEX 00:00:02.00 (goto 2 sec past begin)
Response- OK
Terse Example- Command- <NAME> IX + 2.00
Response- 0
Terse Example- Command- <NAME> IX 2.00
Response- 0
Errors- (Terse, Verbose) 1, ERROR- No spots, unit is empty
2, ERROR- Bad Arguments

TIMECODE

Description- Displays the timecode position of the current spot.
Relevant Flags- None
Required Arguments- None
Optional Arguments- None
Verbose Example- Command- <NAME> TIMECODE
Response- OK
0:00:00:00
Terse Example- Command- <NAME> TC
Response- 0
00 00 00 00

STARTUP

Description- Determines whether or not the Soloist starts playing immediately after power-up or reset with no intervention.
Relevant Flags- None
Required Arguments- None
Optional Arguments- OFF (0), ON (1)
Verbose Example- Command- <NAME> STARTUP
Response- OK
StartUp= OFF
Verbose Example- Command- <NAME> STARTUP ON
Response- OK
Terse Example- Command- <NAME> SU
Response- 0
0
Terse Example- Command- <NAME> SU 1
Response- 0
Errors- (Terse, Verbose) 2, ERROR- Bad Arguments

SHUFFLE

Description- Displays or changes whether or not spots are to be shuffled in the current playlist. If shuffle is ON, any rollover or rollunder or additions to the list will cause a re-shuffle to occur. Any adjacencies created by shuffling the list will be resolved by a post-processor which works to maximize the variation in a shuffled list.

Relevant Flags- None
Required Arguments- None
Optional Arguments- OFF (0), ON (1)

Verbose Example-
Command- `<NAME> SHUFFLE`
Response-
```
OK
Shuffle= OFF
```

Verbose Example-
Command- `<NAME> SHUFFLE ON`
Response-
```
OK
```

Terse Example-
Command- `<NAME> SH`
Response-
```
0
0
```

Terse Example-
Command- `<NAME> SH 1`
Response-
```
0
```

Errors- (Terse, Verbose)
```
2, ERROR- Bad Arguments
```

REPEAT

Description- Determines how the next spot is fetched when the current spot finishes playing. OFF= stop when done with current spot; ONE= repeat current spot; ALL= cycle through the list, or the inventory if there is no list.

Relevant Flags- None
Required Arguments- None
Optional Arguments- OFF (0), ONE (1), ALL (2)

Verbose Example-
Command- `<NAME> REPEAT`
Response-
```
OK
Repeat= OFF
```

Verbose Example-
Command- `<NAME> REPEAT ONE`
Response-
```
OK
```

Terse Example-
Command- `<NAME> RP`
Response-
```
0
0
```

Terse Example-
Command- `<NAME> RP 1`
Response-
```
0
```

Errors- (Terse, Verbose)
```
2, ERROR- Bad Arguments
```

BLANK

Description- Displays or changes the status of the video output when spots are not playing. NOVIDEO= total absence of video signal and sync; BLACK= black video signal; HOLD= the last frame from the spot is held

Relevant Flags- None
Required Arguments- None
Optional Arguments- NOVIDEO (0), BLACK (1), HOLD (2)

Verbose Example-
Command- `<NAME> BLANK`
Response-
```
OK
Blank= NOVIDEO
```

Verbose Example-
Command- `<NAME> BLANK BLACK`
Response-
```
OK
```

Terse Example-
Command- `<NAME> BL`
Response-
```
0
0
```

Terse Example- Command- `<NAME> BL 1`
Response- `0`
Errors- (Terse, Verbose) `2, ERROR- Bad Arguments`

VIDEO

Description- Displays or changes the current video output format between NTSC and PAL.
Relevant Flags- None
Required Arguments- None
Optional Arguments- NTSC (0), PAL (1)
Verbose Example- Command- `<NAME> VIDEO`
Response- `OK`
`Video= NTSC`
Verbose Example- Command- `<NAME> VIDEO PAL`
Response- `OK`
Terse Example- Command- `<NAME> VI`
Response- `0`
`0`
Terse Example- Command- `<NAME> VI 1`
Response- `0`
Errors- (Terse, Verbose) `2, ERROR- Bad Arguments`

ERRORLIMIT

Description- Displays or changes the number of allowable read errors from the disk before the playing spot is terminated.
Relevant Flags- None
Required Arguments- None
Optional Arguments- Number of attempts (0-1000)
Verbose Example- Command- `<NAME> ERRORLIMIT`
Response- `OK`
`ErrorLimit= 0`
Verbose Example- Command- `<NAME> ERRORLIMIT 10`
Response- `OK`
Terse Example- Command- `<NAME> EL`
Response- `0`
`0000`
Terse Example- Command- `<NAME> EL 10`
Response- `0`
Errors- (Terse, Verbose) `2, ERROR- Bad Arguments`

LIST

Description- Displays or changes the current playlist in the Soloist. Volatile lists are stored in the Soloists DRAM memory and are lost at power down. Non-volatile lists are stored in the Soloists permanent memory area and are retained at power down.
Relevant Flags- None
Required Arguments- None
Optional Arguments- ADD spot name (optionally Drive Number, Disc Number, Spot Name or Spot Number)
CLEAR (clears the list from volatile memory)

ERASE (clears the volatile as well as the non-volatile lists)
LOAD (copies a list from non-volatile to volatile memory)
PTR [number] (display or changes the pointer to the current list event in volatile memory)
SAVE (saves the volatile list to non-volatile memory)

Verbose Example- Command- `<NAME> LIST`
Response- `OK`
`1- 0, 0, 0 NESTLE 11.286 Mbytes 0:00:30.04 3.0 Mbps 10/22/96 4.58`
`2- 0, 0, 1 HAINES 11.319 Mbytes 0:00:30.01 3.0 Mbps 8/08/96 6.23`
`Current program is 2, 2 program(s) in the list`

Verbose Example- Command- `<NAME> LIST ADD "HEINZ"`
Response- `OK`

Verbose Example- Command- `<NAME> LIST CLEAR` (clears volatile schedule only)
Response- `OK`

Verbose Example- Command- `<NAME> LIST ERASE` (clears volatile schedule only)
Response- `OK`

Verbose Example- Command- `<NAME> LIST LOAD` (loads schedule into volatile memory)
Response- `OK`

Verbose Example- Command- `<NAME> LIST PTR 0003` (select third list event as next)
Response- `OK`

Verbose Example- Command- `<NAME> LIST SAVE` (save a schedule to non-volatile memory)
Response- `OK`

Terse Example- Command- `<NAME> LI`
Response- `0`
`0002`
`NESTLE`
`HAINES`

Terse Example- Command- `<NAME> LI ADD "HEINZ"`
Response- `0`

Terse Example- Command- `<NAME> LI CLEAR`
Response- `0`

Terse Example- Command- `<NAME> LI ERASE`
Response- `0`

Terse Example- Command- `<NAME> LI LOAD`
Response- `0`

Terse Example- Command- `<NAME> LI PTR 0003`
Response- `0`

Terse Example- Command- `<NAME> LI SAVE`
Response- `0`

Errors- (Terse, Verbose)
`2, ERROR- Bad Arguments`
`6, ERROR- Load failed- EEPROM missing or bad`
`7, ERROR- Save failed- EEPROM missing or bad`
`8, ERROR- Erase failed- EEPROM missing or bad`
`11, ERROR- List event not added`

SCHEDULE

Description- Displays or changes the current schedule in the Soloist. Volatile schedules are stored in the Soloists DRAM memory and are lost at power down. Non-volatile schedules are stored in the Soloists permanent memory area and are retained at power down. Multiple commands can be entered as a single schedule event by using the PIPE character to connect them. (Example: TH 01/12/97 – 12 00 LI CLEAR|LI ADD "NESTLE"|PL) Wildcard fields are entered as dashes --. (In the example, the event will be executed at 12 minutes after every hour on Thursday, Jan 12, 1997)

NOTE: Only multiple commands can be entered with the PIPE character, multiple schedule events can NOT be added on a single line with the pipe character.

Relevant Flags- None

Required Arguments- None

Optional Arguments- ADD event (WKDAY MO DATE YEAR HR MIN SEC <CMD>[SEP][ARG]|<CMD>[SEP][ARG])
CLEAR (clears the schedule from volatile memory)
DEBUG [OFF(0), COM1(1), COM2(2), COMBOTH(3)] (sends status out when events are executed)
ERASE (clears the volatile as well as the non-volatile schedules)
LOAD (copies a schedule from non-volatile to volatile memory)
RUN [OFF(0), ON(1)] (changes the schedule from inactive to active or vice-versa)
SAVE (saves the schedule to non-volatile memory)

Verbose Example- Command- <NAME> SCHEDULE
Response- OK
Run= OFF
Debug= OFF
1- WE 01/12/97 12:00:00 STOP
1 event(s) in the schedule.

Verbose Example- Command- <NAME> SCHEDULE ADD MO -- -- -- 12 00 00 PS "HAINES"
Response- OK

Verbose Example- Command- <NAME> SCHEDULE CLEAR (clears volatile schedule only)
Response- OK

Verbose Example- Command- <NAME> SCHEDULE DEBUG COM1 (sends debug info out on com1)
Response- OK

Verbose Example- Command- <NAME> SCHEDULE ERASE (clears volatile & non-volatile schedules)
Response- OK

Verbose Example- Command- <NAME> SCHEDULE LOAD (loads schedule into volatile memory)
Response- OK

Verbose Example- Command- <NAME> SCHEDULE RUN ON (begin execution of a schedule)
Response- OK

Verbose Example- Command- <NAME> SCHEDULE SAVE (save a schedule to non-volatile memory)
Response- OK

Terse Example- Command- <NAME> SC
Response- 0
03 01 12 97 12 00 00 STOP

Terse Example- Command- <NAME> SC ADD MO -- -- -- 12 00 00 PS "HAINES"
Response- 0

Terse Example- Command- <NAME> SC CLEAR
Response- 0

Terse Example- Command- <NAME> SC DEBUG 1
Response- 0

Terse Example- Command- <NAME> SC ERASE
Response- 0

Terse Example- Command- <NAME> SC LOAD
Response- 0

Terse Example- Command- <NAME> SC RUN 1
Response- 0

Terse Example- Command- <NAME> SC SAVE
Response- 0

Errors- (Terse, Verbose)

```
2, ERROR- Bad Arguments
6, ERROR- Load failed- EEPROM missing or bad
7, ERROR- Save failed- EEPROM missing or bad
8, ERROR- Erase failed- EEPROM missing or bad
10, ERROR- Schedule event not added
```

NAME

Description- Displays or changes the name used to address the Soloist. It can be up to 20 alpha-numeric characters in length.

Relevant Flags- None
Required Arguments- None
Optional Arguments- UnitName (20 alpha-numeric characters)

Verbose Example-
Command-
```
<NAME> NAME
```
Response-
```
OK
SOLOIST
```

Verbose Example-
Command-
```
<NAME> NAME COMEDY
```
Response-
```
OK
```

Terse Example-
Command-
```
<NAME> NA
```
Response-
```
0
SOLOIST
```

Terse Example-
Command-
```
<NAME> NA COMEDY
```
Response-
```
0
```

COM1

Description- Displays or changes the configuration for the RS-232 port. NOTE: If this parameter is changed from the RS-232 port, you will obviously have to change the settings in your controller or terminal program to match afterwards.

Relevant Flags- None
Required Arguments- None
Optional Arguments- Baud, DataBits, StopBits, Parity

Verbose Example-
Command-
```
<NAME> COM1
```
Response-
```
OK
38400 Baud, 8 Data Bits, 1 Stop Bit(s), NO parity
```

Verbose Example-
Command-
```
<NAME> COM1 19200 8 1 N
```
Response-
```
OK <NONE IF CHANGE IS MADE FROM RS-232 PORT>
```

Terse Example-
Command-
```
<NAME> C1
```
Response-
```
0
038400 8 1 N
```

Terse Example-
Command-
```
<NAME> C1 19200 8 1 N
```
Response-
```
0 <NONE IF CHANGE IS MADE FROM RS-232 PORT>
```

Errors- (Terse, Verbose)

```
2, ERROR- Bad Arguments
```

COM2

Description- Displays or changes the configuration for the RS-422 port. NOTE: If this parameter is changed from the RS-422 port, you will obviously have to change the settings in your controller or terminal program to match afterwards.

Relevant Flags- None
Required Arguments- None
Optional Arguments- Baud, DataBits, StopBits, Parity

Verbose Example- Command- `<NAME> COM2`
Response- `OK`
`38400 Baud, 8 Data Bits, 1 Stop Bit(s), NO parity`
Verbose Example- Command- `<NAME> COM2 19200 8 1 N`
Response- `OK <NONE IF CHANGE IS MADE FROM RS-422 PORT>`
Terse Example- Command- `<NAME> C2`
Response- `0`
`038400 8 1 N`
Terse Example- Command- `<NAME> C2 19200 8 1 N`
Response- `0 <NONE IF CHANGE IS MADE FROM RS-422 PORT>`
Errors- (Terse, Verbose) `2, ERROR- Bad Arguments`

TIME

Description- Displays or changes the current time (if the unit is equipped with an add-in real time clock module).
Relevant Flags- None
Required Arguments- None
Optional Arguments- Weekday, Month/Date/Year, Hour/Minute/Second
Verbose Example- Command- `<NAME> TIME`
Response- `OK`
`MO 1/01/97 2:02:32`
Verbose Example- Command- `<NAME> TIME TH 02/09/97 10:05:16`
Response- `OK`
Terse Example- Command- `<NAME> TI`
Response- `0`
`MO 1/01/97 2:02:32`
Terse Example- Command- `<NAME> TI TH 02/09/97 10:05:16`
Response- `0`
Errors- (Terse, Verbose) `2, ERROR- Bad Arguments`
`9, ERROR- Clock failed- RTC missing or bad`

CONFIGURATION

Description- Displays the current configuration for the Soloist. The configuration for the Soloist is loaded from non-volatile memory to volatile memory at power up or reset.
Relevant Flags- None
Required Arguments- None
Optional Arguments- CLEAR (clears the configuration to the default conditions)
ERASE (clears the volatile as well as the non-volatile schedules)
LOAD (copies the configuration from non-volatile to volatile memory)
SAVE (saves the volatile configuration to non-volatile memory)
Verbose Example- Command- `<NAME> CONFIGURATION`
Response- `OK`
`Name= SOLOIST`
`Video= NTSC`
`Talk= OFF`
`StartUp= ON`
`Shuffle= OFF`
`Repeat= ALL`

```
Blank= BLACK
ErrorLimit= 3
Attenuation= 0
Com1= Baud-38400, Data Bits-8, Stop Bits-1, Parity- NONE
Com2= Baud-38400, Data Bits-8, Stop Bits-1, Parity- NONE
Modem=
```

Verbose Example- Command- `<NAME> CONFIGURATION CLEAR`
Response- `OK`

Verbose Example- Command- `<NAME> CONFIGURATION ERASE`
Response- `OK`

Verbose Example- Command- `<NAME> CONFIGURATION LOAD`
Response- `OK`

Verbose Example- Command- `<NAME> CONFIGURATION SAVE`
Response- `OK`

Terse Example- Command- `<NAME> CF`
Response- `0`
`SOLOIST 000 000 001 000 002 001 0003 000`

Terse Example- Command- `<NAME> CF CLEAR`
Response- `0`

Terse Example- Command- `<NAME> CF ERASE`
Response- `0`

Terse Example- Command- `<NAME> CF LOAD`
Response- `0`

Terse Example- Command- `<NAME> CF SAVE`
Response- `0`

Errors- (Terse, Verbose)

```
2, ERROR- Bad Arguments
6, ERROR- Load failed- EEPROM missing or bad
7, ERROR- Save failed- EEPROM missing or bad
8, ERROR- Erase failed- EEPROM missing or bad
```

DIRECTORY

Description- Displays the current directory listing for all drives in the Soloist. Optionally, a single drive or single disc directory can be displayed. The optional REFRESH command can be used to update the directory in the event of a disc swap or other discrepancies

Relevant Flags- None
Required Arguments- None
Optional Arguments- Drive Number, Disc Number
R or REFRESH

Verbose Example- Command- `<NAME> DIRECTORY`
Response- `OK`

```
Directory for Drive 0, Disc 0, Spots 2, Size 22 MB
0, 0, 0 NESTLE 11.286 Mbytes 0:00:30.04 3.0 Mbps 10/22/96 4.58
0, 0, 1 HAINES 11.319 Mbytes 0:00:30.01 3.0 Mbps 8/08/96 6.23
2 spot(s) on 1 disc(s)using 22 MB
```

Verbose Example- Command- `<NAME> DIRECTORY REFRESH`
Response- `OK`

```
Directory for Drive 0, Disc 0, Spots 2, Size 22 MB
0, 0, 0 NESTLE 11.286 Mbytes 0:00:30.04 3.0 Mbps 10/22/96 4.58
0, 0, 1 HAINES 11.319 Mbytes 0:00:30.01 3.0 Mbps 8/08/96 6.23
```

Terse Example- Command- `<NAME> DI`

Response- 0
00 00 000 NESTLE 0011286 00 00 30 04 030 10 22 96 04 58
00 00 001 HAINES 0011319 00 00 30 01 030 08 08 96 06 23

Terse Example- Command- <NAME> DI R
Response- 0
00 00 000 NESTLE 0011286 00 00 30 04 030 10 22 96 04 58
00 00 001 HAINES 0011319 00 00 30 01 030 08 08 96 06 23

Errors- (Terse, Verbose) 2, ERROR- Bad Arguments

INVENTORY

Description- Displays a condensed listing of all spots on all drives in the Soloist. Optionally, a single drive or single disc inventory can be displayed. The optional REFRESH command can be used to update the inventory in the event of a disc swap or other discrepancies

Relevant Flags- None
Required Arguments- None
Optional Arguments- Drive Number, Disc Number
R or REFRESH

Verbose Example Command- <NAME> INVENTORY
Response- OK
NESTLE 0:00:30.04
HAINES 0:00:30.01

Verbose Example Command- <NAME> INVENTORY REFRESH
Response- OK
NESTLE 0:00:30.04
HAINES 0:00:30.01

Terse Example- Command- <NAME> IN
Response- 0
NESTLE
HAINES

Terse Example- Command- <NAME> IN R
Response- 0
NESTLE
HAINES

Errors- (Terse, Verbose) 2, ERROR- Bad Arguments

BANNER

Description- Displays the message header. This header is normally sent on the COM PORT enabled by the TALK parameter (COM1, COM2, or COMBOTH) or is disabled if TALK is set to OFF.

Relevant Flags- None
Required Arguments- None
Optional Arguments- None
Verbose Example- Command- <NAME> BANNER
Response- OK
Soloist Professional MPEG Player
Version 7 Revision 5 3-10-97
Adtec Productions, Inc © 1997

Terse Example- Command- `<NAME> BA`
Response- `0`
`07 05 03 10 97`

MANUFACTURE

Description- Displays the manufacturing information. (ProductName, PartNumber, Revision, RevisionDate, Model, Serial, ManufactureDate, SymphonyNetID)
Relevant Flags- None
Required Arguments- None
Optional Arguments- None
Verbose Example- Command- `<NAME> MANUFACTURE`
Response-
```
OK
Product= SOLOIST
Part Number=
Revision=
Revision Date=
Model=
Serial=
Manufacture Date=
SymphonyNet ID= 0000000
```
Terse Example- Command- `<NAME> MF`
Response-
```
0
SOLOIST 0000000
```
Errors- (Terse, Verbose) `2, ERROR- Bad Arguments`

QUERY

Description- In a multi-user network environment, this command will search for units and the units will respond with their SymphonyNetIDs (The Ids are 20 alpha-numeric characters in length). It is the users responsibility to echo each valid ID after it is received, in order to cause the device to go quiet. All devices will be identified in this manner until the network goes silent for more than 2 seconds. After each device is found, a unique NAME can be assigned to each unit by using the SymphonyNetID to address them independantly.
Relevant Flags- None
Required Arguments- None
Optional Arguments- None
Verbose Example- Command- `<NAME> QUERY`
Response-
```
000000 <HOST MUST ECHO TO SILENCE DEVICE>
000100 <HOST MUST ECHO TO SILENCE DEVICE>
<etc.>
```
Terse Example- Command- `<NAME> QU`
Response-
```
000000 <HOST MUST ECHO TO SILENCE DEVICE>
000100 <HOST MUST ECHO TO SILENCE DEVICE>
```

MEMORY

Description- Displays the current memory allocation for the Soloist.
Relevant Flags- None
Required Arguments- None
Optional Arguments- None
Verbose Example- Command- `<NAME> MEMORY`
Response- `OK`

```
Total Memory= 4194304 bytes
Application Code & Data= 120030 bytes
Directory Buffer= 8000 bytes
FAT Buffer= 131072 bytes
Video Buffer= 3899392 bytes
System Stack= 35776 bytes
```

Terse Example- Command- `<NAME> ME`

Response- `0`

0004194304 0000120030 0000008000 0000131072 0003899392 0000035776

UNITS

Description- Displays the status of all possible drives in the Soloist.

Relevant Flags- None

Required Arguments- None

Optional Arguments- None

Verbose Example- Command- `<NAME> UNITS`

Response- `OK`

```
UNITS
Drive- 0, Discs- 4, Type-ATAPI
Drive- 1, Discs- 0, Type-None
Drive- 2, Discs- 0, Type-None
Drive- 3, Discs- 0, Type-None
Drive- 4, Discs- 1, Type-ATA
Drive- 5, Discs- 0, Type-None
Drive- 6, Discs- 0, Type-None
Drive- 7, Discs- 0, Type-None
Drive- 8, Discs- 0, Type-None
Drive- 9, Discs- 0, Type-None
Drive-10, Discs- 0, Type-None
Drive-11, Discs- 0, Type-None
Drive-12, Discs- 1, Type-SCSI
Drive-13, Discs- 0, Type-None
Drive-14, Discs- 0, Type-None
3 drive(s) total containing 6 disc(s) total
```

Terse Example- Command- `<NAME> UN`

Response- `0`

0401 0000 0000 0000 0102 0000 0000 0000 0000 0000 0000 0000 0103 0000 0000 03 006

MODEM

Description- Displays or changes the current modem initialization string. If the modem string starts with an ‘AT’, then it is used to initialize a modem at power up or reset.

Relevant Flags- None

Required Arguments- None

Optional Arguments- Configuration string for modem initialization

Verbose Example- Command- `<NAME> MODEM`

Response- `OK`

```
Modem= ATE
```

Verbose Example- Command- `<NAME> MODEM ATE0V0X0S0=2`
Response- `OK`
Terse Example- Command- `<NAME> MO`
Response- `0`
`ATE`
Terse Example- Command- `<NAME> MO ATE0V0X0S0=2`
Response- `0`

TALK

Description- Determines whether or not the Soloist sends messages when not explicitly addresssed. OFF= Silent except when being addressed; ON= sends update messages when transport conditions change and when addressed.
Relevant Flags- None
Required Arguments- None
Optional Arguments- OFF (0), ON (1)
Verbose Example- Command- `<NAME> TALK`
Response- `OK`
`Talk= OFF`
Verbose Example- Command- `<NAME> TALK ON`
Response- `OK`
Terse Example- Command- `<NAME> TA`
Response- `0`
`0`
Terse Example- Command- `<NAME> TA 1`
Response- `0`
Errors- (Terse, Verbose) `2, ERROR- Bad Arguments`

ATTENUATE

Description- Displays or changes the current audio attenuation setting. The attenuation level is set in increments of 2 dB up to a maximum of 126dB. Odd input values are rounded down. (e.g. 3 would be saved as 2)
Relevant Flags- None
Required Arguments- None
Optional Arguments- Attenuation level (0-126 dB)
Verbose Example- Command- `<NAME> ATTENUATE`
Response- `OK`
`Attenuate= 0 dB`
Verbose Example- Command- `<NAME> ATTENUATE 64`
Response- `OK`
Terse Example- Command- `<NAME> AT`
Response- `0`
`000`
Terse Example- Command- `<NAME> AT 64`
Response- `0`
Errors- (Terse, Verbose) `2, ERROR- Bad Arguments`

RESET

Description- Reset the Soloist.
Relevant Flags- None

Required Arguments-	None	
Optional Arguments-	None	
Verbose Example-	Command-	<NAME> RESET
	Response-	OK
Terse Example-	Command-	<NAME> RS
	Response-	0

ERROR MESSAGES

TERSE-MODE	**VERBOSE-MODE**
1	**No spots, unit is empty**
2	**Bad Arguments**
3	**Spot does not exist**
4	**Transport in use, no drive access allowed**
5	**Playlist has entries, no cueing allowed**
6	**Load failed- EEPROM missing or bad**
7	**Save failed- EEPROM missing or bad**
8	**Erase failed- EEPROM missing or bad**
9	**Clock failed- RTC missing or bad**
10	**Schedule event not added**
11	**List event not added**

RESPONSE MESSAGES

. = Busy
: = Initializing
? = Unknown Command

Responses from the Soloist are terminated with a blank line (<CR><LF>). In some cases, there may be a delay from the time of the acknowledgement message (OK or 0) to actual return of data. An example would be a Directory or Inventory Refresh, where the drives may take up to 10 or 20 seconds to respond, worst-case. Hence, a return message from the Soloist is not complete until the blank line <CR><LF> is received.