kaleidescape

# Control Protocol Reference Manual

## for Kaleidescape Systems

October 2018

# Contents

USING THIS DOCUMENT 6

OVERVIEW 6

Basic communication 6

Types of control messages 6

QUICK START TUTORIAL 7

Connections 7

Terminal emulator 9

Control messages 12

Command example 12

Command response 13

Event messages 13

COMMUNICATION 14

Physical connection 14

Setting up a TCP/IP connection 14

Setting up an RS-232 serial connection 16

Basic control protocol 18

Control message syntax 18

Device identifier 18

Sequence number 22

Message body 22

Error detection 27

Event messages 28

Maintaining synchronization 29

COMMAND USAGE 29

Connection commands 29

Power commands and messages 30

Idle mode commands and messages 30

Verification commands 31

Protocol commands 31

Event message registration 32

Module registration 32

Friendly name commands 32

OSD control 33
Navigation 33
Menu 34
Views 34
User input 35
View-specific commands 35
Page and content details 36
Screen saver 37
OSD playback control 37
Playback control 37
Movie playback 38
Music playback 38
DVD/Blu-ray Disc navigation 39
Movie playback options 39
Blu-ray Disc playback options 40
Standalone music control (SATP and keypad) 40
Text-based music browsing interface (SATP) 42
Keypad control 42
Advanced integration 43
Lighting, screen masking, video, and audio settings 43
Scripts 44
User-defined events 44
Child user interface commands 44
CONNECTION MANAGEMENT 45
Power commands 46
Idle Mode 49
Verification 51
Protocol 54
Event message registration 57
Module registration 61
Friendly name 62
OSD CONTROL 64
Basic navigation commands 67
Kaleidescape menu commands 72

Views 74
Movie views 75
Music views 78
Store views 82
Other views 82
User input 84
View-specific commands 88
Page and content details 92
Screen saver commands 98
OSD PLAYBACK CONTROL COMMANDS 99
Playback control 102
Playback information 108
Music playback controls 114
DVD/Blu-ray Disc navigation 118
Movie playback options 122
Blu-ray Disc playback options 130
CONTEXT-SENSITIVE COMMANDS 132
STANDALONE MUSIC CONTROL (SATP AND KEYPAD) 135
Text-based music browsing interface (SATP) 136
Overview 136
Implementation and examples 146
Keypad collections and presets 161
ADVANCED INTEGRATION 166
Lighting, screen masking, and video settings 168
Scripts 182
User-defined events 183
Kaleidescape Mobile App and Kaleidescape App for iPad 185
Child user interface 188

OTHER COMMANDS 189

GETTING ADDITIONAL SUPPORT 194

APPENDIX A: COMMAND SUMMARY AND STATUS CODES 194

Commands 194

Status codes 204

APPENDIX B: REVISION HISTORY 205

# Using This Document

The simple text-based control protocol for Kaleidescape Systems can be used to issue commands from, and provide information to, control devices. Connections between control devices and Kaleidescape components can be serial links or via TCP/IP over the Ethernet network.

This document describes the low-level details of the Kaleidescape System control protocol. For templates, modules, and documentation for specific control systems, go to support.kaleidescape.com and search ‘control systems’.

> Note: The information in this document does not apply to the Kaleidescape Reader, the Kaleidescape Bulk Loader, the Kaleidescape Speed Reader, or the Modular Disc Vault. Those devices do not support communication through the control protocol.

This document revision corresponds to kOS version 10.4.2 and 8.4.2.

# Overview

## Basic communication

The connection between a controller and a Kaleidescape component is either a direct serial connection or via TCP/IP over the Ethernet network. For test purposes, a terminal emulator can be used to send commands to a Kaleidescape player. See Quick Start Tutorial on page 7 for details.

## Types of control messages

This document describes three types of control messages: commands, response messages, and event messages.

- Commands are sent from the controller (or terminal) to a Kaleidescape component.
- Response messages are sent from a Kaleidescape component in response to a command.
- Event messages are unsolicited messages sent from a Kaleidescape component to indicate a change of state.

  Event messages can also be sent as response messages to commands. This way the controller can determine the current state of a component even if there has been no change.

Kaleidescape control messages have either three or four segments, separated by slash (/) characters. See Basic control protocol on page 18 for complete information on message syntax.

See Appendix A: Command Summary and Status Codes on page 194 for a complete list of control messages.

# Quick Start Tutorial

This section provides a quick start tutorial using control messages on a computer. Use this information to learn how certain messages affect the system.

## Connections

A computer can be connected to a Kaleidescape component via TCP/IP over the existing Ethernet network or by direct serial connection.

### TCP/IP link

Use the following steps to find the IP address for a Kaleidescape player.

1. Turn on both the computer and the Kaleidescape player.
2. Verify the computer is on the same subnet as the Kaleidescape player.
3. Open the browser interface using:

   http://my-kaleidescape (Windows)
   http://my-kaleidescape.local (Mac)

4. Provide a password as required.  Select the SETTINGS tab.
5. Click on Components in the second row of tabs.
6. Record the IP address listed for the player.

### Serial link

Use the following steps to look up and change the serial port settings for a Kaleidescape player. (Only Premiere line players – with the exception of Mini players – have serial ports.)

1. Turn on both the computer and the Kaleidescape player.
2. Verify the computer is on the same subnet as the Kaleidescape player.
3. Open the browser interface using:

   http://my-kaleidescape (Windows)
   http://my-kaleidescape.local (Mac)

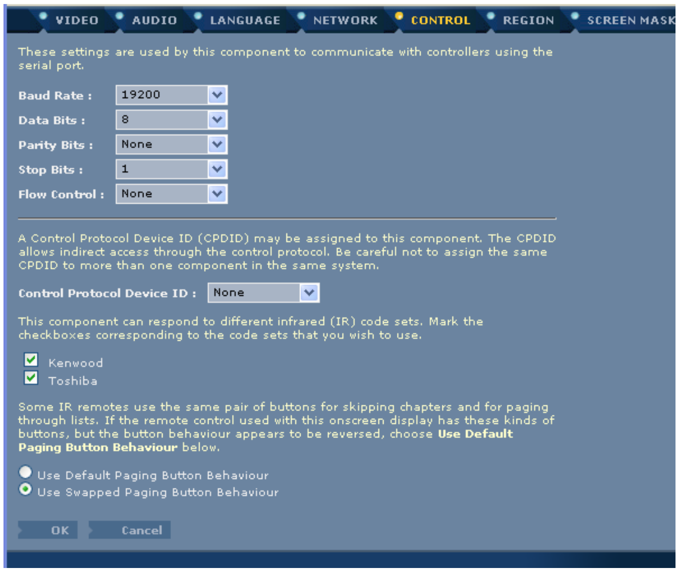
   Provide a password as required.

4. Select the SETTINGS tab.

5. Click on Components in the second row of tabs.

6. Select the Settings button for the player.

7. In the new window, click on the CONTROL tab,



These settings are used by this component to communicate with controllers using the serial port.

Baud Rate : 19200

Data Bits : 8

Parity Bits : None

Stop Bits : 1

Flow Control : None

A Control Protocol Device ID (CPDID) may be assigned to this component. The CPDID allows indirect access through the control protocol. Be careful not to assign the same CPDID to more than one component in the same system.

Control Protocol Device ID : None

This component can respond to different infrared (IR) code sets. Mark the checkboxes corresponding to the code sets that you wish to use.

☑ Kenwood
☑ Toshiba

Some IR remotes use the same pair of buttons for skipping chapters and for paging through lists. If the remote control used with this onscreen display has these kinds of buttons, but the button behaviour appears to be reversed, choose **Use Default Paging Button Behaviour** below.

○ Use Default Paging Button Behaviour
● Use Swapped Paging Button Behaviour

Figure 1: Control tab

8. Verify that Flow Control is set to *None* and make any other required changes.

9. Record the settings for the player. This information is required to set up the terminal emulation program.

10. Click OK to close the window.

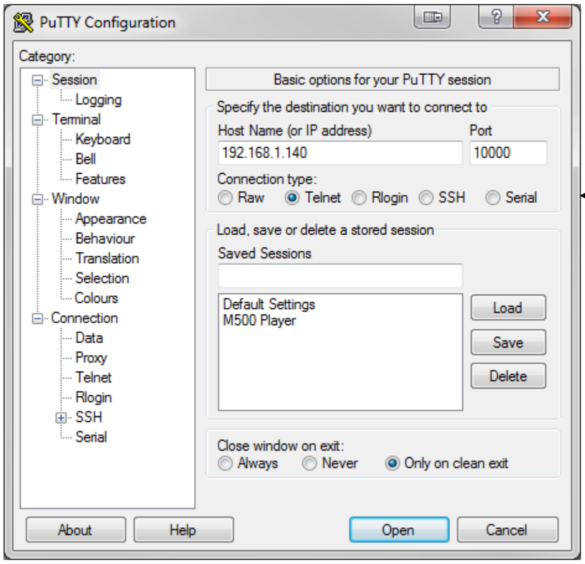
## Terminal emulator

Use the following procedures to set up a terminal emulator. Several third-party emulators are available. This document describes the procedure using PuTTY for Windows. A telnet client in the terminal window can be used for a Mac or Linux.

> Note: PuTTY is available at http://www.chiark.greenend.org.uk/~sgtatham/putty

## TCP/IP connection

1. Open PuTTY on the computer.
2. Select the Telnet radio button in the Connection type section as shown in the following figure.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.1.140

Port: 10000

Connection type: ○ Raw ● Telnet ○ Rlogin ○ SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: Default Settings, M500 Player

Close window on exit: ○ Always ○ Never ● Only on clean exit

Select Telnet

Figure 2: Telnet selection

3. Enter the IP address of the player and enter *10000* for the Port setting.
4. Click Open. A terminal session window appears.

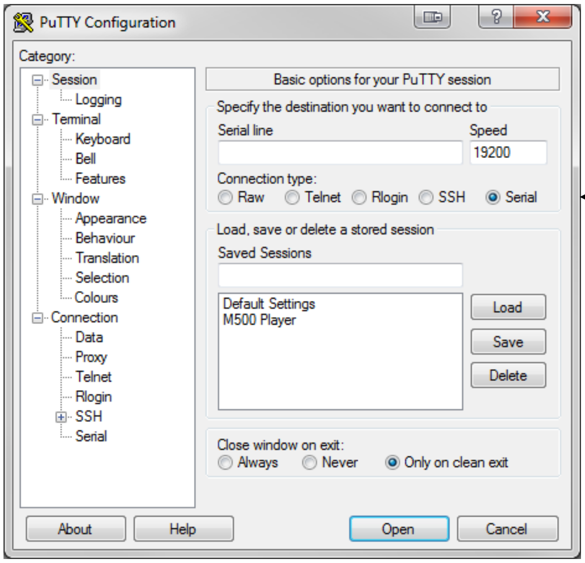
## Serial connection

If using another emulator rather than PuTTY, see Table 1 for serial parameters.

Table 1: Serial Parameters

| Serial Parameter | Player |
|---|---|
| Speed (baud) | 19200 |
| Data bits | 8 |
| Parity | None |
| Stop bits | 1 |
| Flow control | None |

1. Open PuTTY on the computer.
2. Select the Serial radio button in the Connection type section as shown in the following figure.



Specify the destination you want to connect to

Serial line:

Speed: 19200

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Load, save or delete a stored session

Saved Sessions:

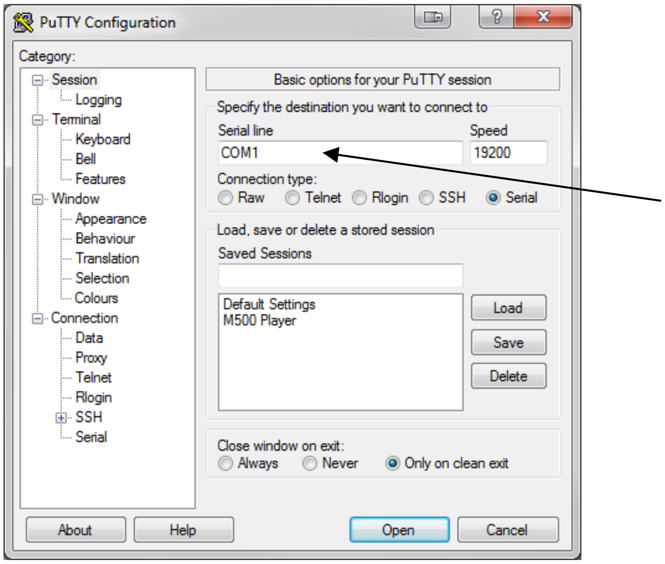
Default Settings
M500 Player

Close window on exit: ○ Always ○ Never ● Only on clean exit

Select Serial

Figure 3: Serial selection

3. Enter the name of the computer serial port in the Serial line text box. This is usually *COM1* or *COM2*; however, USB to serial adaptors usually use *COM3*.



Specify the destination you want to connect to

Serial line: COM1

Speed: 19200

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Load, save or delete a stored session

Saved Sessions

Default Settings
M500 Player

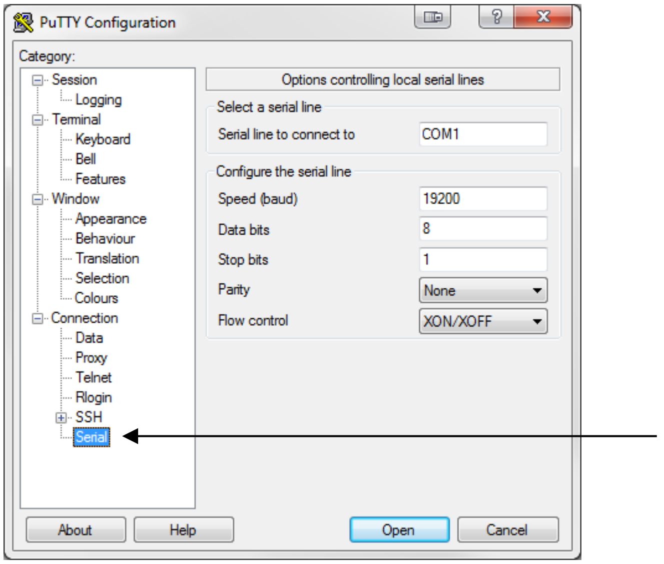
Close window on exit: ○ Always ○ Never ● Only on clean exit

Enter the computer serial port name, e.g., *COM1*

Figure 4: Serial line text box

4. Enter the baud rate found on the browser interface into the Speed text box. The default baud rate is 19200 for a player and 115200 for a server.

5. Click the Serial option under Connection in the Category section of the window as shown in the following figure.



Select a serial line

Serial line to connect to: COM1

Configure the serial line

Speed (baud): 19200

Data bits: 8

Stop bits: 1

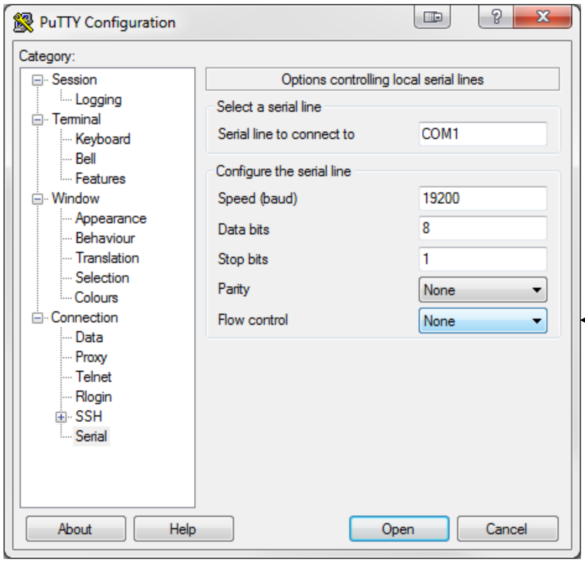
Parity: None

Flow control: XON/XOFF

Click Serial

Figure 5: Serial option

6. Verify that *None* is selected for Flow control from the drop-down menu as shown in the following figure.



Options controlling local serial lines

Select a serial line

Serial line to connect to: COM1

Configure the serial line

Speed (baud): 19200

Data bits: 8

Stop bits: 1

Parity: None

Flow control: None

Select *None*

Figure 6: Flow control setting

7. Click Open. A terminal session window appears.
8. Connect a female-to-female DB-9 serial null modem cable to a serial port or serial adapter on the computer and connect the other end of the cable to the control port on the Kaleidescape player.

## Control messages

After connecting to a Kaleidescape component and opening a terminal session window, enter a sample command for the onscreen display. The following command for controlling the onscreen display is a good example because the command response is immediately visible.

### Command example

Type the following down arrow command in the terminal session window. Remember to press the enter key to send commands in a terminal session.

```
01/1/DOWN:
```

In this example, the command has three parts.

`01` is the device ID for the component receiving the command. The value `01` is the control protocol device ID (CPDID) that identifies the component directly connected. This means that regardless of the CPDID setting for the player connected, the player will act on commands with device ID `01`.

`1` is the sequence number, which the component will send back in response to the command. The sequence number allows the controller to verify that the command was received.

`DOWN` is the message body, which indicates the action the component is to perform. In this example, the `DOWN` command moves the highlight down the list. If the screen saver was on, the screen saver disappears.

## Command response

The following text appears in the terminal session window:

```
01/1/000:/89
```

This is the response from the component to the `DOWN` command. The response to this command has four parts.

`01` is the device ID of the component that sent the response, which matches the device ID of the component that the `DOWN` command was sent to.

`1` is the sequence number. This number is the same number as the command so that responses can be matched to commands.

`000` is the message body which starts with the three-digit status code that indicates whether or not the command was handled successfully. The value `000` indicates successful handling.

`89` is the checksum that accompanies every response from a Kaleidescape component. The controller program can use this number to validate the message.

## Event messages

A `DOWN` command can trigger state changes in the Kaleidescape component that result in event messages. If the screen saver was active when the `DOWN` command was issued, the following the event message appears:

```
01/!/000:UI_STATE:01:00:00:0:/38
```

This event message provides information about the current state of the onscreen display. The message body in this example has six fields separated by colons.

`01` is the device ID of the component that sent the response, which matches the device ID of the component that the `DOWN` command was sent to.

`!` is used as the sequence number to indicate that this is an unsolicited event message.

`000` is the three-digit status code. The value `000` indicates successful handling and no errors.

`UI_STATE` is the name of the message, which indicates that the onscreen display has changed.

`01:00:00:0` are response parameters.

- `01` means that the onscreen display is on the Movie List.
- `00` means that no movie details page or menu is displayed.
- `00` means that no dialog box is displayed.
- `0` means that the screen saver is off.

`38` is the checksum.

Refer to the command sections to view commands, how the system responds, and the event messages generated. See Appendix B: Revision History on page 205 for command additions and changes.

# Communication

The Kaleidescape System communicates with a controller using a simple text-based protocol. There are two important elements for communicating with the Kaleidescape System.

- Setting up a TCP/IP link or a serial link between a controller and a Kaleidescape component
- Understanding the message syntax for the Kaleidescape control protocol

## Physical connection

Although connected to a single component, a controller can route commands to any component in the Kaleidescape System using assigned device ID numbers or component serial numbers.

### Setting up a TCP/IP connection

Each Kaleidescape component can handle up to twenty simultaneous TCP/IP control connections. The controller must specify the IP address of the component and port 10000.

#### Determining the IP address of a Kaleidescape component

Verify that the computer is on the same subnet as the Kaleidescape System, and use the following procedure to find a component IP addresses.

1. Open the browser interface using:

   http://my-kaleidescape (Windows)
   http://my-kaleidescape.local (Mac)

   Provide a password as required.

2. Select the SETTINGS tab.
3. Click on Components in the second row of tabs.

4. Record all IP addresses listed for the components in the group that the controller will connect to.

## Using a stable IP address

When using TCP/IP connections, Kaleidescape components usually obtain IP addresses via DHCP, which means that IP addresses can change.

An unstable IP address can result in the controller losing connection to the Kaleidescape System. There are two ways to make a component IP address stable.

- Reserve an IP address for the Kaleidescape component in the DHCP server. If the DHCP server supports reservations, this is the preferred option.
- Assign a static IP address to the component. This option is less desirable because the Kaleidescape component must be reconfigured if moved to a different subnet.

DHCP reservations provide a central point of management for all IP allocations at a site. With DHCP reservations, network parameters such as IP addresses, subnet mask, default gateway, and DNS servers are set in one place — at the DHCP server. If any of these parameters change, only the DHCP server has to be modified.

For most installations, the DHCP server built into the router is perfectly adequate. The procedure to create DHCP reservations is device-specific. Refer to the router or DHCP server documentation for instructions. DHCP reservations are usually made using MAC addresses. To see MAC addresses for Kaleidescape components, go to the Components tab in the browser interface.

## Setting up an RS-232 serial connection

Kaleidescape components act as an RS-232 Data Terminal Equipment (DTE) device with data rates up to 115,200 baud for servers and up to 57,600 baud for players.

The male DB-9 serial control port uses a standard DTE pinout shown in Figure 7. CTS and RTS are optional and are used only when using hardware flow control.

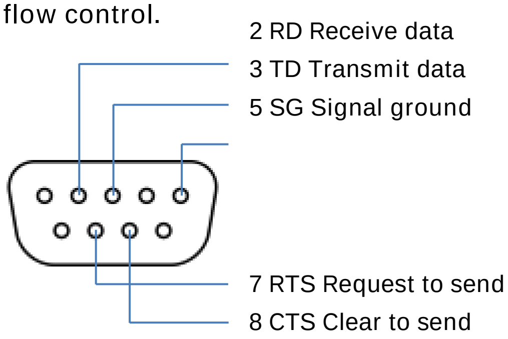


Figure 7: RS-232 pinout

Use a straight-through cable to connect to a DTE device. If connecting to another DTE device, use a null modem cable. If in doubt of which cable to use, check the pinout for the port connecting to the Kaleidescape component. The component can be connected to a standard computer serial port using a null modem cable. Most AMX and Crestron DB-9 serial ports require a null modem cable.

### Configuring the serial port

Use the following steps to configure the Kaleidescape serial port.

1. Open the browser interface using:

   http://my-kaleidescape (Windows)
   http://my-kaleidescape.local (Mac)

   Provide a password as required.

2. Select the SETTINGS tab.
3. Click on Components in the second row of tabs.
4. Click on the Settings button for the player.
5. In the new window, click on the CONTROL tab.
6. Select the appropriate serial port settings for the player from the drop-down menus. Table 2 shows the default port settings for servers and players. Server port settings cannot be changed. The port settings on the control system must be set to match.

Table 2: Default port settings for servers and players

| Serial Parameter | Server | Player |
|---|---|---|
| Speed (baud) | 115200 | 19200 |
| Data bits | 8 | 8 |
| Parity bits | None | None |
| Stop bits | 1 | 1 |
| Flow control | None | None |

### Flow control

Kaleidescape components support hardware flow control using the RTS/CTS lines, but not software flow control. Table 3 describes effects of flow control options: *None* (default) or *RTS/CTS*.

> Note: RTS and CTS signal wiring can be omitted between a component serial port and the controller if the default flow control setting is used.

Table 3: Flow control settings

| Flow Control Option | Line Status |
|---|---|
| None | Component ignores CTS. Sends data at any time, and keeps RTS asserted, indicating that component is always ready to accept data. |
| RTS/CTS | Component only sends data when the CTS line is asserted, and asserts the RTS line when the component is ready to accept data. The controller does not send data when the RTS line is clear. |

### Character echo

To avoid overloading the serial link, Kaleidescape components do not echo characters back to the sender. Error detection is handled through checksums. See Error detection on page 27 for more information.

If working with a terminal emulator, turn on local character echo to view commands. Both carriage return and line feed might also have to be set to send.

## Estimating response times

Kaleidescape components respond to all commands. To detect dropped or lost commands, the controller can expect a response within the time frames listed in Table 4. Time is based on the serial link speed.

Table 4: Response times of Kaleidescape Systems

| Speed | Response Time |
|---|---|
| 9600 | 3.2 seconds |
| 19200 | 1.6 seconds |
| 38400 | 0.8 seconds |
| 57600 | 0.5 seconds |

# Basic control protocol

Control commands for the Kaleidescape System use a simple ASCII text-based format. Each control message has up to 1024 characters followed by a carriage return or line feed character (decimal ASCII 13 and 10).

Kaleidescape components ignore multiple carriage returns or line feed characters as empty messages. This makes it easy to enter control commands in a terminal emulator while developing or testing controller programming.

## Control message syntax

Kaleidescape control messages have either three or four segments, separated by slash (/) characters:

```
device_id/seq/message_body[/checksum]
```

*device_id* identifies the Kaleidescape component sending or receiving a message. This is usually the control protocol device ID (CPDID) of the Kaleidescape component.

*seq* is a sequence number used to match a response from the component with the original command sent by the controller.

*message_body* contains the message name and other data. Message fields are delimited by colons and order dependent. See Message body on page 22.

*checksum* aids in error detection. The checksum is optional in commands but part of all response and event messages.

## Device identifier

There are two types of device identifiers.

- Control protocol device ID (CPDID)
- Serial number device identifier

A music zone identifier can be appended to a device ID with .xx, e.g., 01.02 is device 01, music zone 02.

The device ID identifies the Kaleidescape component receiving or responding to a command. The device ID allows controllers to send and receive messages to and from a Kaleidescape component while connected to another component. This is called command routing.

Command routing is a feature of the Kaleidescape control protocol that allows commands routed through a single physical connection on one Kaleidescape component to reach other Kaleidescape components on the network. This allows multiple Kaleidescape components to be controlled with only one connection to the controller.

Commands prefaced with a unique CPDID for a component or with the serial number of the component, are received by that component, no matter the physical connection.

The advantage of using a unique CPDID as opposed to a serial number device ID is that a component can be replaced without changing the control program. The advantage of using the serial number format is that CPDIDs do not have to be assigned.

> Note: The only module provided by Kaleidescape that requires CPDIDs to be set is the AMX module. Other modules communicate directly with the player being controlled.

> Note: It is not usually necessary to connect a controller to the server. This is useful only when using the server as a gateway to route commands to other components.

## Control protocol device ID (CPDID)

Control protocol device ID (CPDID) numbers are the most common device IDs. CPDIDs are two-digit numbers assigned to a component for communication.

### Numbering

CPDID numbers range from 01 to 99.

- CPDID 01 is the component directly connected to the controller, either via a TCP/IP session or a serial link.
- A specific CPDID (from 02 to 99) can be set to allow for command routing. Using this ID sends the command to the specified component.
- CPDID ?? is a special identifier that appears only in response messages when the original command was garbled and the Kaleidescape component could not determine the intended destination for the command.
- CPDID 00 is invalid.

### Assigning CPDID numbers

Use the following steps to assign CPDIDs on the installer pages of the browser interface.

1. Open the browser interface using:

   http://my-kaleidescape (Windows)
   http://my-kaleidescape.local (Mac)

   Provide a password as required.

2. Select the SETTINGS tab.

3. Click on Components in the second row of tabs.

4. Click on the Settings button for the player.

5. In the new window, click on the CONTROL tab.

6. Select the Control Protocol Device ID from the drop-down menu.

   Never assign the same CPDID (other than *None*) to two components in the same server group. In the case of duplicate CPDIDs, messages directed to that CPDID return a response status error indicating a device identifier conflict.

   - When connecting a serial controller directly to a player or Cinema One (1st generation) via the RS-232 control port, communication parameters must be set to match the serial controller.

   - The M300 Player has no serial port, but can be controlled via TCP/IP or via a controller connected by RS-232 to another player.

7. Click OK.

### Example 1

In this example, the system has a Cinema One (1st generation) and an M500 Player with a controller connected directly to the M500 Player. The Cinema One has been set to CPDID 02. The controller can send messages to the Cinema One using CPDID 02, or send messages to the M500 Player using CPDID 01.

Controller sends:

```
02/1/GET_NUM_ZONES:                  (command routed to Cinema One)
```

Kaleidescape System sends:

```
02/1/000:NUM_ZONES:01:03:/94         (response from Cinema One)
```

Controller sends:

```
01/1/GET_NUM_ZONES:                  (command sent directly to connected
                                      M500 Player)
```

Kaleidescape System sends:

```
01/1/000:NUM_ZONES:01:01:/91         (response from M500 Player)
```

Example 2

In this example, the system has an M500 Player directly connected to a controller. The M500 Player has been set to CPDID 03. The controller can send messages to the player using CPDID 01 (the local device ID) or 03 (the indirect/routable CPDID).

| | |
|---|---|
| Controller sends: | |
| `03/1/GET_NUM_ZONES:` | (command routed to M500 Player) |
| Kaleidescape System sends: | |
| `03/1/000:NUM_ZONES:01:01:/93` | (response from M500 Player) |
| Controller sends: | |
| `01/1/GET_NUM_ZONES:` | (command sent directly to connected M500 Player) |
| Kaleidescape System sends: | |
| `01/1/000:NUM_ZONES:01:01:/91` | (response from M500 Player) |

### Serial number device identifier

The device identifier can also be the serial number of the component. The serial number device identifier is specified by a pound sign followed by the serial number. Leading zeros in serial numbers may be omitted in commands. For example, to send a command to a component with serial number 0000 000144B,

```
#144B/1/DOWN:
```

The response message will also contain the serial number, identified with a pound sign, and zero-padded to 12 hexadecimals. For example,

```
#00000000144B/1/000:/30
```

By default, events always use the two-digit device ID format. To receive event messages in serial number format, use the `ENABLE_EVENTS` command. See `ENABLE_EVENTS` on page 57.

### Music zone identifier

The device identifier field can contain an optional music zone identifier. A music zone identifier has two digits and is preceded by a period. For example,

```
01.02/3/PAUSE:
```

In this example, the connected device (01) has a music zone (02), which is paused. When using serial number syntax, the music zone identifier follows the serial number. For example,

```
#144B.02/3/PAUSE:
```

Commands sent to a music zone do not specifically control the Kaleidescape onscreen display. When a music zone is specified in the device identifier, only commands that control music are allowed. These commands include `BROWSE` and music controls, such as `PAUSE`.

If the music zone identifier is omitted, the command applies to the onscreen display if applicable. If the component does not have an onscreen display, only a limited set of commands are available, e.g., network setting commands.

Examples

```
01/6/STOP:
```

This command controls the onscreen display. If the screen saver is active, the first time this command is sent, the screen saver is cleared. The second time the command is sent, the movie or music playing stops.

```
01.01/6/STOP:
```

Sending this command to an M500 Player music zone only stops the music (if playing) and does not disable the screen saver.

If the onscreen display is set to control a different music zone (via the Now Playing view or the `SET_CONTROLLED_ZONE` command), the `01/6/STOP:` command stops music in that zone, rather than the zone in the connected player. The `01.01/6/STOP:` command always stops the music for the connected player.

## Sequence number

The sequence number is a single digit from 0 to 9 for commands and response messages, or the character ! for an event message.

When a Kaleidescape component sends a response to a command from a controller, the component puts the sequence number from the message received into the response. This sequence number allows the controller to keep track of the responses to each command.

If a Kaleidescape component cannot determine the sequence number of a command, the question mark character ? is used as the sequence number in the response.

## Message body

The message body contains the name of the message and any associated data. Event messages and command responses also contain a status code in the message body. See Appendix A: Command Summary and Status Codes on page 194 for a list of status codes.

The information in the message body is delimited into fields separated by colons. The number and meaning of the fields varies with the specific message.

For commands, the first field is always the name of the message. For example, `GO_MOVIE_COVERS` or `PLAY`.

For response and event messages, the message body always begins with a status code, followed by the message name and any additional fields.

The following example has some commands that might be sent by a controller, along with the responses and event messages from the Kaleidescape component.

```
01/1/GO_MOVIE_COVERS:
   01/1/000:/89                              (response)
   01/!/000:UI_STATE:03:00:00:0:/40          (event message)
01/2/PLAY:
   01/2/000:/90                              (response)
   01/!/000:UI_STATE:03:02:00:0:/42          (event message)
   01/!/000:TITLE_NAME:Almost Famous:/34     (event message)
   …
01/3/PA.SE:
   01/3/010:/92                              (response)
01/4/~AUSE:/30
   01/4/003:/95                              (response)
```

This sequence of commands displays the Movie Covers view and begins playing the highlighted movie, Almost Famous. Note that these commands have increasing sequence numbers, and the first three have no checksums.

The responses from the Kaleidescape component have sequence numbers matching the commands and contain checksums (as do all responses sent by Kaleidescape components).

The status code `000` in the first two responses indicates that no error occurred receiving or handling the commands. Some event messages follow the responses that indicate changes in the state of the zone under control.

In the last example, the controller sent two garbled `PAUSE` commands some time after the original commands. Because there is no checksum, the component must look at the message itself to determine whether the message was sent correctly. The component does not recognize the command name and responds with a status of `010` indicating an invalid command.

The controller reacts by sending another `PAUSE` command, this time with a checksum of `30` (the correct checksum for the ungarbled message), but the message becomes garbled again. The component compares the checksum (`30`) provided by the controller to the checksum on the received message (`76`), to determine that the message was garbled. The component then responds with a status of `003`, indicating a checksum error. Checksums can be used to detect problematic RS-232 connections.

## Message status codes

All response and event messages contain a zero-padded, three-digit status code. A `000` status code means no errors; any other code indicates an error. Error messages are not intended to be displayed to the end user.

- For event messages, the status code is always `000` (no error) to simplify message parsing.
- For response messages, the status code can be a non-zero value indicating an error of some sort. For a complete list, see Status codes on page 204.

The controller must inspect the status field of any response message before taking action based on the response.

Some commands, for example the `BROWSE` command, return a brief message along with the status code if there is an error. For example,

```
01/1/BROWSE:bad::1-10::
01/1/012:Invalid node:/15
```

## Message character set

Other than message delimiters (carriage return and line feed), all characters in a control message must be printable, 8-bit ASCII characters in the ISO 8859-1 (Latin-1) character set — no raw control characters are allowed.

The characters allowed in a message range from decimal value 32 (the space character) to decimal 255 (y-umlaut, ÿ).

Certain characters from the Latin-1 character set are also supported using special escape sequences.

> Note: A controller must be able to handle escaped characters in response and event messages, and must be able to escape ASCII characters with special meaning in the message format.

To support interactive serial or TCP/IP connections where each character is transmitted as typed, the serial protocol handles ASCII 8 (BS) and 127 (DEL) by deleting the preceding character.

> Note: The `SET_PROTOCOL_SETTINGS` command can override most of this behavior.

### Special characters

To include format and control characters in a message data field, characters must be escaped with a backslash \ character using the sequences in Table 5.

If the controller cannot handle escape sequences, escaped characters can be avoided by using the `SET_PROTOCOL_SETTINGS` command.

Table 5: Special characters

| Character | Escape sequence |
|---|---|
| Line feed | `\n` |
| Carriage return | `\r` |
| Horizontal tab | `\t` |
| Slash (/) | `\/` |
| Backslash (\) | `\\` |
| Colon (:) | `\:` |
| Any Latin-1 character including characters with accents | `\dnnn`<br>where `nnn` is the zero-padded three-digit decimal value for the character |

The Kaleidescape System uses the Latin-1 character set for accented letters common in foreign films and names, and for special punctuation. To include these characters, use the character directly or use the escape sequence `\d` followed by the three-digit decimal value for the letter. For example, to represent the character a-acute (á), which has the 8-bit ASCII value 225, use the sequence `\d225`.

### Field processing algorithm

The pseudo code below shows how to process a field from a control message. The variable `field` is assumed to be a string of the field within the overall message string. The variable `field_terminated` is used to make sure the field ends with a colon character. The variable `field_text` is filled with the processed characters from the message. The variables `escaped` and `char` handle escape sequences in the loop that runs through the message until it encounters a colon character terminating the field, or the end of the message string. After the loop finishes, if a colon was not found (or a segment delimiter (/) was found), an error occurs.

```
field_text = ""
field_terminated = false
escaped = false
char = ''

for each character in field:
   if (escaped is true) then
      if (character is 'd') then
         char = (next three digit characters as a decimal value)
         (advance loop past digit characters)
      else if (character is 'n') then
         char = (newline character)
      else if (character is 'r') then
         char = (carriage return character)
      else if (character is 't') then
         char = (tab character)
      else if (character is '/') then
         char = '/'
      else if (character is ':') then
         char = ':'
      else if (character is '\') then
         char = '\'
      end if

      append char to field_text
      escaped = false
   else if (character is '\') then
      escaped = true
   else if (character is ':') then
      field_terminated = true
      break loop // a field delimiter ends scan w/success
   else if (character is '/') then
      break loop // a segment delimiter ends scan w/o a valid field
   else if (not between 32 and 126) then
      note error condition
      break loop
   else
      append character to field_text
   end if
next for

if (field_terminated is false) then
   note error condition
end if
```

## Error detection

The Kaleidescape control protocol allows for a checksum to be included with every message for reliable communication even on noisy connections.

The checksum is an optional segment in controller commands. If a checksum is provided, the Kaleidescape System uses this information to detect transmission errors before processing a command. If the checksum is omitted, the slash character before the checksum must also be omitted.

The checksum is always included in response and event messages. The controller has the option to use or ignore the checksum.

The controller programmer must decide how to handle a response or event message with an incorrect checksum. This situation indicates a problem with the communications link and must, at the least, result in a log message to identify the problem. The controller software can check the message body status field for a 000 (no error) value to decide whether or not to resend the command with the matching sequence number.

The programmer must decide if the controller will assume that the command arrived intact. This decision depends on whether a repeated command is preferable to a missed one. This decision depends on the specific message. A repeated `KALEIDESCAPE_MENU_TOGGLE` command, which flashes the Kaleidescape menu briefly onscreen, is likely to be more annoying than a missed one, where the menu simply fails to appear.

On the other hand, a repeated `STOP` command can do no harm. The simplest policy is to ignore all responses and events that fail checksum verification, on the assumption that it is better to ignore a garbled message than to act on incorrect contents.

Checksums are not applicable when using a TCP/IP connection, because the TCP protocol has built-in error handling.

### Calculating the checksum

A checksum is a zero-padded, two-digit decimal number, calculated as a straightforward sum, modulo 100, of the decimal value of each character in the message before the checksum, including the last slash character.

Use the following pseudo code to compute a checksum.

```
checksum = 0
for each character in the full message before the checksum segment
   checksum = checksum + (current character decimal value)
   checksum = checksum modulo 100
end for
```

## Event messages

Event messages are unsolicited messages sent from a Kaleidescape component to indicate a change of state. It is not necessary for a controller to query a Kaleidescape component repeatedly for information about the current state of the component. Any changes such as the state of the user interface are automatically sent to the controller.

The component sends out event messages any time relevant information changes. The controller only needs to monitor for such messages and take appropriate action.

Some event messages are nearly identical to command response messages.

For example,

```
01/!/000:UI_STATE: …
```

contains the same message parameters as the response to the `GET_UI_STATE` command.

```
01/1/000:UI_STATE: …
```

Examples

1. When a movie enters the end credits, a `MOVIE_LOCATION` event message is sent to the controller which can respond by raising the lights in the theater.
2. A `PLAY_STATUS` message indicates playback and scan speed, which the controller can use to swap a Play button with a Pause button, or to highlight the Fast Forward button when the player is scanning. This message also contains information on the currently playing title and chapter numbers, as well as the playback location within each, which the controller can display on a touch panel.
3. A `TITLE_NAME` message indicates the title of the movie currently playing.
4. For theater installations that must manage changes in aspect ratio for masking systems or installations with external scalers, the event messages `VIDEO_MODE` and `SCREEN_MASK` provide the necessary information to adjust the theater equipment automatically as the user switches between the onscreen display and movies with different aspect ratios.
5. For installations that must react to video mode changes, such as those incorporating external scalers or display devices that do not automatically detect changes in the video mode, a `VIDEO_MODE` event message indicates the current video mode for each video output of the movie zone. This information can be used by the controller to configure the downstream video display or processing devices accordingly.
6. The `UI_STATE` message provides information about which screen is visible in the Kaleidescape user interface, along with any details pages or menus, dialog boxes, or a screen saver.

7. The USER_INPUT message tells if the user is being prompted for input, what type of input, what the prompt is, and any currently entered characters. The controller can inspect this event message and change the display to show a numeric keypad or alphanumeric keyboard, display a field with the prompt string, as well as show what the user has already entered.

### Maintaining synchronization

Some control protocol messages change the behavior of a component. When the component restarts or loses connection to the controller, those messages must be resent to the component.

For example, the SET_SCREEN_MASK message sends data to the component that the component remembers and uses until a restart. When the component restarts, the component state resets to the default values. This means the SET_SCREEN_MASK message has to be sent again.

The controller can tell that a component might have restarted or temporarily lost the serial connection when the controller receives the PLAYER_RESTART, VIDEO_MODE or SCREEN_MASK messages.

# Command Usage

Understanding how commands in the Kaleidescape control protocol work together allows programmers to use all of the capabilities provided by the Kaleidescape System.

This general overview of commands divides commands into five main groups depending on usage.

- Connection
- OSD control
- Playback control
- Standalone music control (SATP and keypad)
- Advanced integration

## Connection commands

Connection commands affect all components. These commands can be divided into the following groups.

- Power
- Idle mode
- Verification
- Protocol
- Event message registration

- Module registration
- Friendly name

## Power commands and messages

### Commands

`ENTER_STANDBY`

`LEAVE_STANDBY`

`PLAYER_RESTART`

`GET_DEVICE_POWER_STATE`

When a controller connects to the Kaleidescape System for the first time, the controller must check the power state of the component with the `GET_DEVICE_POWER_STATE` command.

If a player is in standby, the controller can send a `LEAVE_STANDBY` command. Kaleidescape components other than the M-Class players and Cinema One reset the TCP/IP connection at power on, so if a `DEVICE_POWER_STATE` message does not appear within a second, the controller should drop the connection, wait at least 30 seconds, and then try to reconnect. A Kaleidescape component can take several minutes to reboot, depending on system conditions. If the controller cannot reconnect within a reasonable amount of time, the controller should display an error message to the user.

In some situations, it can be difficult to determine whether an existing TCP/IP connection is still active. In this case, a command can be sent to determine whether the connection is active. The `GET_TIME` command is a good test command because this command has no effect on the system.

When a component restarts, depending on the specific type of component, the controller either loses the TCP/IP connection to the component, receives a `DEVICE_POWER_STATE` event message indicating a change in power state, or receives a `PLAYER_RESTART` event message. The controller should handle any of these situations by reconnecting to and resynchronizing with the component. Use the `GET_UI_STATE` command to the view changes.

## Idle mode commands and messages

### Commands

`GET_SYSTEM_READINESS_STATE`

`LEAVE_IDLE_MODE`

When a controller connects to an Encore Movie Player or Kaleidescape Cinema One (2nd generation), the controller may check the idle mode using GET_SYSTEM_READINESS_STATE.

The controller can monitor idle mode while it is connected using the SYSTEM_READINESS_STATE message. The controller can remove the system from idle mode by sending LEAVE_IDLE_MODE. The controller can display a message that the Kaleidescape system is “Spinning up” when SYSTEM_READINESS_STATE = 1, and dismiss the message after the Kaleidescape system has left idle mode, SYSTEM_READINESS_STATE = 0.

A Kaleidescape Encore Movie Player or Cinema One (2nd generation) will automatically enter idle mode after a period of inactivity. The Kaleidescape system will exit idle mode after receiving LEAVE_IDLE_MODE or after any user interaction.

## Verification commands

Commands

`GET_SYSTEM_VERSION`
`GET_DEVICE_TYPE_NAME`
`GET_AVAILABLE_DEVICES`
`GET_NUM_ZONES`
`GET_TIME`
`GET_AVAILABLE_DEVICES_BY_SERIAL_NUMBER`

After a connection has been established to a component that is powered on, check the current system version with the `GET_SYSTEM_VERSION.` This command can also be used to check the protocol version as well as the kOS version.

The `GET_NUM_ZONES` command can be used to verify controller configuration. For example, if the controller has been configured to control a movie zone on a component that does not have a movie zone, the `GET_NUM_ZONES` command returns a `0` for the number of movie zones, allowing the controller to produce an error message for the user. The `GET_DEVICE_TYPE_NAME` command can be used in conjunction with the error message to make the error message more meaningful.

The `GET_AVAILABLE_DEVICES_BY_SERIAL_NUMBER` and `GET_AVAILABLE_DEVICES` commands can be used to discover all components in the system.

## Protocol commands

Commands

`GET_PROTOCOL`
`GET_ACTIVE_PROTOCOL`
`SET_PROTOCOL_SETTINGS`
`SET_SUPPORTED_PROTOCOL`

After a connection has been established to a component that is powered on, check the protocol version with the `GET_PROTOCOL` command. The controller can require a minimum protocol version to ensure that all commands are available. If the protocol version is high enough, the controller can continue with initialization.

To match controllers and device protocol versions, use the GET_ACTIVE_PROTOCOL and SET_SUPPORTED_PROTOCOL to read the current sessions protocol version and, if necessary, set the correct version for the current session that is compatible with both the control system and the Kaleidescape devices being controlled.

During startup, changing the protocol settings to enable binary delimiters allows for faster parsing and processing by the controller. Use the SET_PROTOCOL_SETTINGS command to change protocol settings.

## Event message registration

Commands

| | |
|---|---|
| ENABLE_EVENTS | GET_DEVICE_INFO |
| DISABLE_EVENTS | SEND_EVENT |

To control a movie or music zone on the component connected directly to the controller, use the GET_DEVICE_INFO command to identify the Control Protocol Device ID (CPDID) of the connected device. Even though the component responds to commands sent to CPDID 01, the component only generates event messages from the assigned CPDID. The GET_DEVICE_INFO command provides the assigned CPDID so the controller can accept event messages from the component.

If the controller is configured to control a zone other than the locally connected movie zone, the controller must use the ENABLE_EVENTS command to receive event messages from that zone. This command applies to music zones on the directly connected component as well as all zones on other components.

## Module registration

Commands

SEND_TO_SYSLOG

To assist in troubleshooting, a Kaleidescape System can send information to the diagnostic logs about the control module version. Use the SEND_TO_SYSLOG command to send a message with the module description and version. Kaleidescape recommends sending this command once a day and each time Kaleidescape control is initiated.

Note: This information is covered by the Kaleidescape privacy policy.

## Friendly name commands

Commands

GET_FRIENDLY_NAME

SET_FRIENDLY_NAME

GET_FRIENDLY_SYSTEM_NAME

The controller can display the name of the Kaleidescape movie zone or music zone being controlled. Use the GET_FRIENDLY_NAME command to discover the friendly name. Use SET_FRIENDLY_NAME to label a zone on a user interface. Use GET_FRIENDLY_SYSTEM_NAME to determine the system name that the zone is a part of.

## OSD control

The onscreen display (OSD) control commands navigate the Kaleidescape onscreen display, control movie and music playback through the onscreen display, and respond to playback events. Software modules that support these commands are sometimes referred to as OSD control modules.

All OSD control commands are sent to a movie zone, and all event messages related to OSD control are sent from a movie zone. These commands can be divided into the following groups.

- Navigation
- Menu
- Views
- User input
- View-specific commands
- Page and content details
- Screen saver

### Navigation

Commands

| | |
|---|---|
| Arrow commands | Page up/down commands |
| SELECT | POSITION_SELECT |
| CANCEL | Paging and skipping |
| CHILD_SELECT | |

The user must be provided with basic controls to navigate the onscreen display. The UP, DOWN, LEFT, RIGHT and other arrow commands are used to navigate the onscreen options, and the SELECT or CHILD_SELECT command to select. (CHILD_SELECT behaves like SELECT but also activates the child user interface.) The PAGE_UP, PAGE_DOWN and other paging commands allow the user to search through long lists quickly in the list and collections views.

The arrow commands and page up/down commands have _PRESS and _RELEASE versions to refine user navigation (for example PAGE_UP_RELEASE). If the controller can detect distinct press and release events, program the controller so that pressing the corresponding button sends the _PRESS version of the command, and releasing the button sends the _RELEASE version.

To close dialog boxes and cancel user input, use the CANCEL or STOP_OR_CANCEL command. The STOP_OR_CANCEL command is useful when there is no explicit Cancel or Exit button available.

A controller that supports a touch-sensitive video feed can use the POSITION_SELECT command to make it possible for the user to have direct touch control to interact with the Kaleidescape onscreen display. Kaleidescape refers to software that support this feature as OSD Video and software that does not as OSD No Video.

Paging and skipping commands behave like PAGE_UP or PAGE_DOWN in the user interface.

## Menu

### Commands

KALEIDESCAPE_MENU_ON

KALEIDESCAPE_MENU_OFF

KALEIDESCAPE_MENU_TOGGLE

DISC_OR_KALEIDESCAPE_MENU

The Kaleidescape user interface has three types of views for the user to access content: list, covers, and collections. These views can all be accessed through the Kaleidescape menu, which is accessed using the Kaleidescape menu commands.

If the controller cannot support both a Disc Menu button and a Kaleidescape Menu button, use the DISC_OR_KALEIDESCAPE_MENU command to provide a single button that provides both functions.

## Views

> Note: Music view commands will re-direct to the movie collection “Songs” when sent to a player which does not support music.

### Commands

GET_UI_STATE

GO_MOVIES

GO_MOVIE_LIST

GO_MOVIE_COVERS

GO_MOVIE_COLLECTIONS

GO_MOVIE_COLLECTION

GO_MOVIE_STORE

GO_SYSTEM_STATUS

GO_VAULT_SUMMARY

GO_MUSIC

GO_MUSIC_LIST

GO_MUSIC_COVERS

GO_MUSIC_COLLECTIONS

GO_MUSIC_COLLECTION

GO_NOW_PLAYING

GO_PARENTAL_CONTROL

DISC_IN_TRAY_TOGGLE

Direct access to the views on the Kaleidescape menu are available by using the GO_MOVIE_LIST, GO_MOVIE_COVERS, GO_MOVIE_COLLECTIONS, GO_MOVIE_STORE, GO_MUSIC_LIST, GO_MUSIC_COVERS, GO_MUSIC_COLLECTIONS, GO_NOW_PLAYING, GO_VAULT_SUMMARY, GO_SYSTEM_STATUS, and GO_PARENTAL_CONTROL commands. To switch between music and movie views, use the GO_MOVIES and GO_MUSIC commands.

The controller can provide feedback about the view currently active by using the GET_UI_STATE command or waiting for a UI_STATE event message when the view changes.

## User input

Commands

GET_USER_INPUT

SET_USER_INPUT_ENTRY

KEYBOARD_CHARACTER

BACKSPACE

GET_USER_INPUT_PROMPT

SELECT

CANCEL

User input is sometimes requested from the Kaleidescape onscreen display. Capable controllers should watch for unsolicited USER_INPUT event messages to display the appropriate keyboard or numeric keypad when required. These event messages can occur at any time, even during movie playback. When user input is requested, the specifics of the on-screen prompts can be determined by use of the GET_USER_INPUT_PROMPT command. Entering user input is done with KEYBOARD_CHARACTER commands send the input character by character, or SET_USER_INPUT_ENTRY for sending complete strings. The BACKSPACE command is used to delete characters and fix errors. Sometimes, input requires a SELECT command to confirm that the entry is complete. The CANCEL command is used to cancel user input.

When user input is not being requested, KEYBOARD_CHARACTER commands act differently.

## View-specific commands

Commands

FILTER_LIST

DEFAULT_LEVEL

SAFE_LEVEL

SHUFFLE_COVER_ART

ALPHABETIZE_COVER_ART

GET_USER_INPUT

CHILD_SHUFFLE_COVER_ART

KEYBOARD_CHARACTER

KEYBOARD_LITERAL

BACKSPACE

SELECT

CANCEL

GO_PARENTAL_CONTROL

Different commands are enabled depending on the user interface view. Parental controls are enabled on all movie views. To enter a parental control passcode to enable restricted content, send the numeric passcode to the component using KEYBOARD_CHARACTER commands. The BACKSPACE command can be used to fix typing errors when entering the code and the CANCEL command to return to the movie library.

As characters are entered, USER_INPUT event messages are generated containing the characters currently displayed on the screen and the passcode prompt. When entering a passcode, the digits are masked by asterisks.

Direct access to the parental control settings view can be made through the GO_PARENTAL_CONTROL command.

Lists in the Movie List, Music List, Movie Collections, and Music Collections views can be filtered using the FILTER_LIST command. This command puts the OSD in a filtering mode. To filter the list, send KEYBOARD_CHARACTER commands containing the string for filtering. As characters are entered, USER_INPUT event messages are generated containing the text, along with a prompt that can be displayed on the controller.

When filtering the list, BACKSPACE can be used to delete characters, and CANCEL to exit filtering mode.

When not in filtering mode, sending a KEYBOARD_CHARACTER command causes the display to jump quickly to movies or albums starting with the keyboard letter without entering filtering mode.

When the onscreen keyboard is displayed KEYBOARD_LITERAL is used to enter characters. Focus is placed on the DONE button after every character sent, ready for the SELECT command to complete the onscreen keyboard entry.

In the Movie Covers and Music Covers views, the SHUFFLE_COVER_ART command can be used to force the cover art shuffling.

## Page and content details

Commands

DETAILS

GET_CONTENT_DETAILS

GET_HIGHLIGHTED_SELECTION

As the user navigates the onscreen display, HIGHLIGHTED_SELECTION event messages are generated for each selection highlight change. The HIGHLIGHTED_SELECTION message supplies a handle that can be used by the GET_CONTENT_DETAILS command to get selection item details.

### Screen saver

#### Commands

GO_SCREEN_SAVER

STOP_SCREEN_SAVER

After establishing a fresh connection to a Kaleidescape component, Kaleidescape recommends sending the STOP_SCREEN_SAVER command to clear the screen saver. This saves the user a button press. The UI_STATE message is used to determine whether or not the screen saver is active and can be used to display a message on the controller when the screen saver is active.

## OSD playback control

These commands control movie and music playback through the onscreen display.

These commands can be divided into the following groups.

- Playback control
- Playback information
- Music playback controls
- DVD/Blu-ray Disc navigation
- Movie playback options
- Blu-ray Disc playback options

### Playback control

#### Commands

PLAY

PAUSE

STOP

REPLAY

NEXT and PREVIOUS

CHILD_STOP

CHILD_PLAY

Paging and skipping

Kaleidescape menu commands

INTERMISSION_ON

INTERMISSION_OFF

INTERMISSION_TOGGLE

SCAN_FORWARD and SCAN_REVERSE

CHILD_PAUSE

Basic movie and music playback control can be achieved through the PLAY, PAUSE, and STOP commands, along with the paging and skipping commands.

The CHILD_PLAY, CHILD_STOP, and CHILD_PAUSE commands function like the corresponding non-child playback commands, and also activate the child user interface if not already active.

These commands do not directly control music playback when sent to a movie zone. These commands can also have different effects depending on the current status of the onscreen display. For example, if the screen saver is active, 01/1/PLAY: first hides the screen saver before playing anything. The command 01.01/1/PLAY: simply starts music playback of whatever is in the Now Playing view.

The INTERMISSION_ commands produce an effect similar to PAUSE, but also trigger lighting events and display an Intermission screen.

## Movie playback

### Commands

SET_STATUS_CUE_PERIOD

GET_PLAY_STATUS

GET_PLAYING_TITLE_NAME

GET_UI_STATE

GET_MOVIE_LOCATION

These commands determine the state of movie playback. These commands can be used to display playback information to the user, change controls on a touch panel page, or trigger lighting events. UI_STATE and MOVIE_LOCATION messages indicate whether or not a movie is playing and are generated when movie playback stops or starts.

The PLAY_STATUS message can be used to determine how far movie playback has progressed, what chapter playback is on, or whether playback is paused, scanning, or playing. This message is normally sent during chapter changes or changes in playback. For more frequent updates, use the SET_STATUS_CUE_PERIOD command to set PLAY_STATUS messages to be generated once per second.

## Music playback

### Commands

> Note: The response code for music related commands will return “Command is not available” for products which do not support music.

MUSIC_RANDOM_ON

MUSIC_REPEAT_OFF

MUSIC_RANDOM_TOGGLE

MUSIC_REPEAT_ON

MUSIC_REPEAT_OFF

MUSIC_REPEAT_TOGGLE

GET_CONTROLLED_ZONE

SET_CONTROLLED_ZONE

GET_MUSIC_NOW_PLAYING_STATUS

GET_MUSIC_PLAY_STATUS

GET_MUSIC_TITLE

Music playback status has a different set of commands than movie playback. The MUSIC_PLAY_STATUS message is used instead of PLAY_STATUS messages to discover how far the current track playback has progressed, what track playback is on, and whether playback is paused, scanning or playing. The MUSIC_PLAY_STATUS message frequency is controlled by the SET_STATUS_CUE_PERIOD command like the PLAY_STATUS message.

The GET_MUSIC_TITLE and GET_MUSIC_NOW_PLAYING_STATUS commands request information about the music currently playing, that can be used to populate a user display. Information includes the name of the song, the artist performing the song, and the album with the song.

Various settings related to music playback can be controlled in the Now Playing view on the onscreen display. Whether the music playlist plays back randomly, whether the music repeats, and what music zone the onscreen display is controlling can all be modified from this view. These settings can also be changed directly through the control protocol using the MUSIC_RANDOM_TOGGLE and MUSIC_REPEAT_TOGGLE commands (along with _ON and _OFF variants, and the SET_CONTROLLED_ZONE and GET_CONTROLLED_ZONE commands.

## DVD/Blu-ray Disc navigation

Commands

DISC_MENU

DISC_TOP_MENU

DISC_RESUME

DISC_OR_KALEIDESCAPE_MENU

START_CHAPTER_ENTRY

START_DISC_TITLE_ENTRY

KEYBOARD_CHARACTER

DVDs and Blu-ray Discs have menus that can be accessed through the DISC_MENU command. If there is no room for discrete DISC_MENU and KALEIDESCAPE_MENU controls on the control interface, use the DISC_OR_KALEIDESCAPE_MENU command. To leave the disc menu, use the DISC_MENU command again, the DISC_RESUME command, or even the PLAY command.

It is possible to jump to specific chapters on a disc using the START_CHAPTER_ENTRY command, followed by KEYBOARD_CHARACTER commands to select the chapter.

## Movie playback options

Commands

SHOW_NAVIGATION_OVERLAY

STATUS_AND_SETTINGS

SET_FAVORITE_SCENE_START

SET_FAVORITE_SCENE_END

START_SEND_NUMBER_TO_DISC_ENTRY

ANGLE_NEXT

ANGLE_PREVIOUS

AUDIO_NEXT

`SUBTITLES_NEXT`

`GET_CAMERA_ANGLE`

There are several playback options to be adjusted during movie playback. These options are accessed onscreen through the movie overlay. This overlay can be toggled using the `STATUS_AND_SETTINGS` command.

Some of the functions in the movie overlay can be accessed directly using control protocol commands. For example, the `SET_FAVORITE_SCENE_START` and `SET_FAVORITE_SCENE_END` commands can be used to mark scenes. The `ANGLE_NEXT` and `ANGLE_PREVIOUS` commands change angles. The `AUDIO_NEXT` and `SUBTITLES_NEXT` change the audio and subtitle tracks for the disc.

### Blu-ray Disc playback options

#### Commands

Blu-ray color buttons

`GET_MOVIE_MEDIA_TYPE`

`BLURAY_SPECIAL_STOP`

`BLURAY_POPUP_MENU_TOGGLE`

Some Blu-ray Discs support color buttons in the disc menus and special features. To use these controls, use the Blu-ray color button commands, `RED`, `GREEN`, `BLUE`, and `YELLOW`. Blu-ray specific controls can be set to display only when a Blu-ray Disc is played back. The `MOVIE_MEDIA_TYPE` event message indicates what type of disc is being played.

## Standalone music control (SATP and keypad)

These commands control the Kaleidescape System when the Kaleidescape onscreen display is not available, or the user prefers not to use the OSD because these commands provide a more direct experience. These commands support simple keypads and more complex controllers with graphical displays.

> Note: The response code for music related commands will return “Command is not available” for products which do not support music.

Standalone music control commands can be divided into the following groups.

- Text-based music browsing interface (SATP)
- Keypad collections and presets

Basic playback information is used differently by OSD control, SATP (Standalone Touch Panel) control, and keypad control. The same commands that control basic playback through the onscreen display also control music playback directly when sent to a music zone.

#### Commands

`PLAY`

`PAUSE`

`REPLAY`

`STOP`

NEXT and PREVIOUS

GET_MUSIC_TITLE

SET_STATUS_CUE_PERIOD

GET_MUSIC_PLAY_STATUS

SCAN_FORWARD and SCAN_REVERSE

GET_MUSIC_NOW_PLAYING_STATUS

MUSIC_RANDOM_TOGGLE

MUSIC_REPEAT_TOGGLE

All the messages in this section are sent to and from a music zone. To receive event messages from a music zone, use the ENABLE_EVENTS command for the zone.

### SATP applications

For SATP applications, because there is no onscreen display, the controller must display information about music playback on the controller screen. The MUSIC_TITLE message provides information that can be used to display information to the user including the current song title, artist name, and album title, along with information required to find the URL for cover art.

For information about playback progress for the current track, use the SET_STATUS_CUE_PERIOD command to generate MUSIC_PLAY_STATUS event messages every second. The information in the event message about the playback mode (playing, rewinding, fast forwarding, paused) also states how far playback is into the current track.

The MUSIC_NOW_PLAYING_STATUS message is generated when random and repeat settings are changed. This message can be used to display information to the user. To change the random and repeat settings, send the MUSIC_RANDOM_TOGGLE and MUSIC_REPEAT_TOGGLE commands (along with their _ON and _OFF variants).

### Keypad applications

For keypad applications, the MUSIC_RANDOM_TOGGLE and MUSIC_REPEAT_TOGGLE commands (along with their _ON and _OFF variants) can be used to toggle the status of random and repeat. If feedback is supported, such as a simple text display or highlighted button, the MUSIC_NOW_PLAYING_STATUS message can be used to determine the current status of the random and repeat functions.

Depending on the space available for text display, the MUSIC_TITLE message can be used to provide feedback to the user about the music currently playing (the current song title, artist name, and album title, along with information required to find the URL for cover art).

For information about playback progress for the current track, use the SET_STATUS_CUE_PERIOD command to generate MUSIC_PLAY_STATUS event messages every one or three seconds. The information in the event message about the playback mode (playing, rewinding, fast forwarding, paused) also states how far playback is into the current track.

## Text-based music browsing interface (SATP)

### Commands

BROWSE

PERFORM_ACTION

GET_MUSIC_NOW_PLAYING_STATUS

The Kaleidescape System supports a text-based interface for browsing the music library. The text-based music control interface is for controllers that have a graphical display component but cannot provide video for the Kaleidescape onscreen display. Kaleidescape modules that implement these commands are sometimes referred to as Standalone Touch Panel modules, or SATP modules.

This interface is controlled through the BROWSE and PERFORM_ACTION commands. This text-based interface has a series of pages of text arranged in a hierarchical fashion. The BROWSE command is used to retrieve information stored at each level of the hierarchy. The PERFORM_ACTION command is used to begin playback, queue tracks, or similar actions. When the structure of the hierarchy or the data in that hierarchy changes, a MUSIC_NOW_PLAYING_STATUS message is generated with a new generation value to indicate the change. See Standalone music control (SATP and keypad) on page 40 for more detail on how to use this interface.

## Keypad control

### Commands

PLAY_FIRST_IN_MUSIC_COLLECTION

PLAY_NEXT_IN_MUSIC_COLLECTION

PLAY_PREVIOUS_IN_MUSIC_COLLECTION

GET_PLAYING_MUSIC_INFORMATION

ASSIGN_PLAYING_MUSIC_TO_PRESET

PLAY_MUSIC_PRESET

GET_MUSIC_PRESET_INFORMATION

Collections created by the user in the browser interface can be associated with a simple control protocol command. These commands work well as individual buttons on a keypad. To navigate a collection, use the PLAY_FIRST_IN_MUSIC_COLLECTION, PLAY_NEXT_IN_MUSIC_COLLECTION, and PLAY_PREVIOUS_IN_MUSIC_COLLECTION commands.

Music presets are used to create a simplified user interface for users with minimal controls. A controller can be preprogrammed with a set of presets to call back at will, or the controller can be programmed to allow the user to modify the presets on the fly. Presets are called using the PLAY_MUSIC_PRESET command and are stored with the ASSIGN_PLAYING_MUSIC_TO_PRESET command.

For a description of the current setting of a preset, use GET_MUSIC_PRESET_INFORMATION. This command is useful if a keypad has text display capabilities.

The MUSIC_PRESET_INFORMATION message can be used in conjunction with the PLAYING_MUSIC_INFORMATION message to discover which presets are active. For example, a keypad button can be illuminated when the associated preset is playing.

## Advanced integration

The Kaleidescape System provides information to support advanced lighting, masking, and other forms of system integration. These commands can be divided into the following groups.

- Lighting, screen masking, and video settings
- Scripts
- User-defined events

### Lighting, screen masking, video, and audio settings

Commands

GET_CINEMASCAPE_MASK

GET_CINEMASCAPE_MODE

GET_MOVIE_LOCATION

GET_SCALE_MODE

GO_CALIBRATE_MASKING

GO_CALIBRATE_MASKING_OVERSCAN

SET_CINEMASCAPE_MODE

SET_SCREEN_MASK

GET_SCREEN_MASK

GET_VIDEO_COLOR

GET_SCREEN_MASK2

The MOVIE_LOCATION event message can be used to trigger lighting events by monitoring when movie playback begins and ends, when the end credits start to roll, and whether or not the intermission function is activated.

The SCREEN_MASK message provides information on the aspect ratio of the video output, as well as more detailed masking information that can be fed into a masking system.

To reduce controller processing time, the SCREEN_MASK2 message can provide masking information specifically calibrated to match the format expected by the masking processor. The masking processor must be calibrated first using the GO_CALIBRATE_MASKING and GO_CALIBRATE_MASKING_OVERSCAN commands.

Some movies place subtitles in areas covered by the masking system. Use the SET_SCREEN_MASK command to reposition subtitles if a screen masking system is being used.

For players with output that can be distributed from a 2:35 theater to another room with a non-CinemaScape friendly display, the CINEMASCAPE_MODE message provides information on the CinemaScape mode. The CINEMASCAPE_MASK message returns the frame aspect ratio for the video format.

The SCALE_MODE event message is most useful to theaters using CinemaScape Native mode, providing information to the projector on whether image scaling is required.

### Scripts

> Note: Scripts are not supported on Encore Movie Players or Cinema One (2nd generation).

Command

PLAY_SCRIPT

Scripts can be created on the user pages of the browser interface. These scripts can be played using the PLAY_SCRIPT command. One of the steps that can be added to a script is sending a command to the controller during script execution. These steps are received in USER_DEFINED_EVENT event messages.

### User-defined events

Command

SEND_EVENT

User-defined events can be used by the controller to perform tasks based on commands sent from scripts, sent by other controllers using SEND_EVENT, volume commands from other controllers (including IR remotes), or when the Kaleidescape System requests that its input be selected using SELECT_KALEIDESCAPE_INPUT.

## Child user interface commands

ENTER_CHILD_MODE

LEAVE_CHILD_MODE

GET_CHILD_MODE_STATE

A user-defined event, SELECT_KALEIDESCAPE_INPUT, is issued informing the controller when a player enters the child user interface, or when the screen saver active over the child user interface is cleared. Using another remote to control the Kaleidescape component, or sending non-child commands, switches the component out of the child user interface.

The controller can explicitly request the child user interface with ENTER_CHILD_MODE, exit the child user interface with LEAVE_CHILD_MODE, and query if the child user interface is active with GET_CHILD_MODE_STATE.

# Connection management

Connection commands include turning components on and off, verifying system configuration. Commands are grouped with a detailed description of each command including command examples. Table 6 lists connection commands.

Table 6: Connection management command summary

| Command | Description |
|---|---|
| Power commands | |
| GET_DEVICE_POWER_STATE | Returns power state of a component. |
| PLAYER_RESTART | Event message stating that a player has just been restarted. |
| ENTER_STANDBY | Puts component into standby. |
| LEAVE_STANDBY | Takes component out of standby. |
| Idle Mode | |
| GET_SYSTEM_READINESS_STATE | Returns the idle mode of Encore Movie Player or Cinema One (2nd generation). Idle mode is not supported on Premiere line products. |
| LEAVE_IDLE_MODE | Takes Encore Movie Player or Cinema One (2nd generation) out of idle mode.<br>Idle mode is not supported on Premiere line products. |
| Verification | |
| GET_AVAILABLE_DEVICES | Returns a list of CPDIDs for all system components powered on. |
| GET_AVAILABLE_DEVICES_BY_SERIAL_NUMBER | Returns list containing the serial number device identifiers of all components in the system. |
| GET_DEVICE_TYPE_NAME | Returns component type. |
| GET_NUM_ZONES | Returns number of zones in a component. |
| GET_SYSTEM_VERSION | Returns protocol version number and the version of kOS. |
| Protocol | |
| GET_PROTOCOL | Returns protocol version number. |
| SET_PROTOCOL_SETTINGS | Changes protocol syntax. |
| SET_SUPPORTED_PROTOCOL | Changes Protocol version for the session. |

| Command | Description |
|---|---|
| GET_ACTIVE_PROTOCOL | Returns the current protocol for the session. |
| Event message registration | |
| ENABLE_EVENTS | Enables event messages from a specified movie or music zone. |
| DISABLE_EVENTS | Disables event messages from a specified movie or music zone. |
| GET_DEVICE_INFO | Returns component device type, serial number, device ID, and IP address. |
| Module registration | |
| SEND_TO_SYSLOG | Posts message to Kaleidescape System logs. |
| Friendly name | |
| GET_FRIENDLY_NAME | Returns name of component or music zone. |
| SET_FRIENDLY_NAME | Renames component or music zone. |
| GET_FRIENDLY_SYSTEM_NAME | Returns the system name. |

## Power commands

### GET_DEVICE_POWER_STATE

| | |
|---|---|
| Affects | All components |
| Command | `GET_DEVICE_POWER_STATE:` |
| Response/Event | *status*:DEVICE_POWER_STATE:*power_state*:<br>[*zone_1_state*:…*zone_n_state*:] |

A component responds with the component's current power state.

*power_state* has the following values.

| | |
|---|---|
| 0 | component is in standby |
| 1 | component is powered on |

*zone_1_state* … *zone_n_state*

describe the availability of each zone for the component.

| | |
|---|---|
| 0 | zone is disabled |
| 1 | zone is available |

An event message is generated when the power state of a component changes. Because the 1080p Player, 1080p Mini Player, Movie Player 2, and Music Player disconnect the TCP/IP connection when entering or leaving standby, the controller might not be connected to the system when the DEVICE_POWER_STATE event is generated on these components.

Example 1 Command/Response

Controller sends:

```
01/1/GET_DEVICE_POWER_STATE:
```

Kaleidescape System sends:

```
01/1/000:DEVICE_POWER_STATE:0:0:1:1:/77
```

Example 2 Event message

Controller sends:

```
01/1/LEAVE_STANDBY:
```

Kaleidescape System sends:

```
01/!/000:DEVICE_POWER_STATE:1:1:/50
```

In the first example, the power state of a Kaleidescape Cinema One (1st generation) is requested. The response says that the system is in standby, but that zones 2 and 3 are still available and active.

In the second example, a `LEAVE_STANDBY` command is sent to an M500 Player, causing it to leave standby. As the player exits standby, a `DEVICE_POWER_STATE` event message is generated stating that the component is powered on and its zone is available.

### PLAYER_RESTART

| | |
|---|---|
| Affects | Any component with zones |
| Event | `PLAYER_RESTART:` |

This event message is generated when a user presses the Power button or after the component receives a `LEAVE_STANDBY` command. The message is generated only after the component has finished powering up and is ready for user input.

This notification can be used to enable buttons on a touch panel, or inform the user that the component is ready to receive input. Receiving this event message can also be a cue to send commands that establish settings, such as `SET_STATUS_CUE_PERIOD`, and get the current state of the system through messages such as `GET_UI_STATE` and `GET_MOVIE_LOCATION`.

Note: This event message does not contain a status code.

### ENTER_STANDBY

| | |
|---|---|
| Affects | Any component with zones |
| Command | `ENTER_STANDBY:` |
| Response | *`status`*`:` |

This command causes the component to enter standby immediately. After entering standby mode, the component sends a `DEVICE_POWER_STATE` event message.

Use the LEAVE_STANDBY command to turn the component back on. If the component is already in standby when this command is received, only the normal status reply is sent.

When the 1080p Player, 1080p Mini Player, Music Player, or Movie Player 2 enters standby, the TCP/IP connection is temporarily dropped. This situation can cause the controller to miss the DEVICE_POWER_STATE message while the controller reconnects to the player. These players do not return a response to the ENTER_STANDBY command.

Example

Controller sends:

```
01/1/ENTER_STANDBY:
```

Kaleidescape System sends:

```
01/1/000:/89
01/!/000:DEVICE_POWER_STATE:0:0:/47
01/!/000:VIDEO_MODE:00:00:00:/56
01/!/000:MUSIC_NOW_PLAYING_STATUS:00000:00000:1:0:0000000046::/76
01/!/000:HIGHLIGHTED_SELECTION::/63
```

In this example, the ENTER_STANDBY command is sent to an M500 Player. The component enters standby and then sends several event messages indicating the change in state.

### LEAVE_STANDBY

| | |
|---|---|
| Affects | Any component with zones |
| Command | LEAVE_STANDBY: |
| Response | *status*: |

This command causes the component to exit standby mode. If the component is in standby when this command is received, the component sends out a DEVICE_POWER_STATE event message when startup is complete. If the component is on when this command is received, nothing more happens. The ENTER_STANDBY command can be used to put the component into standby.

For the Movie Player, this command can only be issued directly to the RS-232 port.

For the 1080p Player, 1080p Mini Player, Movie Player 2 and Music Player, this command works over both RS-232 and TCP/IP. After issuing this command over TCP/IP, the connection is disconnected. Wait 15 seconds and reconnect before sending further commands.

For the Cinema One (1st generation) and M-Class players, this command works on both the RS-232 port and the TCP/IP port. The connection is not disconnected after issuing the command over TCP/IP. To retrieve information about the current state of the Kaleidescape component, send GET_UI_STATE and GET_CHILD_MODE_STATE commands.

For the Encore products, this command works only on the TCP/IP port. The connection is not disconnected after issuing the command over TCP/IP. To retrieve information about the current state of the Kaleidescape component, send `GET_UI_STATE` and `GET_CHILD_MODE_STATE` commands.

> Note: Command routing can be used to leave standby on Encore players, M-Class players and Cinema Ones (1st generation), but not on the 1080p Player, 1080p Mini Player, Movie Player 2, Music Player, or Movie Player.

Example

Controller sends:

```
01/1/LEAVE_STANDBY:
```

Kaleidescape System sends:

```
01/1/000:/89
01/!/000:DEVICE_POWER_STATE:1:1:/50
01/!/000:VIDEO_MODE:02:02:13:/65
```

In this example, the `LEAVE_STANDBY` command is sent to an M500 Player, causing the player to leave standby. This causes `DEVICE_POWER_STATE` and `VIDEO_MODE` event messages to be generated.

## Idle Mode

GET_SYSTEM_READINESS_STATE

| | |
|---|---|
| Affects | Encore players and Cinema One (2nd generation) |
| Command | `GET_SYSTEM_READINESS_STATE:` |
| Responase/Event | *`status`*`: SYSTEM_READINESS_STATE:`*`state`*`:` |

A component responds with the systems current idle mode.

*`state`* has the following values.

| | |
|---|---|
| `0` | system is ready |
| `1` | system is becoming ready |
| `2` | system is idle |

An event message is generated when the idle mode of the Encore player or Cinema One (2nd generation) changes.

Example 1 Event message

Controller sends:

```
01/1/LEAVE_IDLE_MODE:
```

Kaleidescape System sends:

```
01/!/000:SYSTEM_READINESS_STATE:1:/68
01/!/000:SYSTEM_READINESS_STATE:0:/67
```

Example 2 Command/Response

Controller sends:

```
01/1/GET_SYSTEM_READINESS_STATE:
```

Kaleidescape System sends:

```
01/1/000:SYSTEM_READINESS_STATE:2:/85
```

In the first example, a LEAVE_IDLE_MODE command is sent to Alto causing it to exit idle mode. Alto responds by first indicating it is leaving idle mode, then that it has left idle mode and its zones are available.

In the second example the idle mode of Alto is requested. The response indicates the system is in idle mode.

### LEAVE_IDLE_MODE

| | |
|---|---|
| Affects | Encore players and Cinema One (2nd generation) |
| Command | LEAVE_IDLE_MODE: |

This command causes Encore player or Cinema One (2nd generation) to exit idle mode.

Example 1 Command/Response

Controller sends:

```
01/1/LEAVE_IDLE_MODE:
```

Kaleidescape System sends:

```
01/1/000:/89
```

In this example, the LEAVE_IDLE_MODE command is sent to Strato.

## Verification

### GET_AVAILABLE_DEVICES

| | |
|---|---|
| Affects | All components |
| Command | `GET_AVAILABLE_DEVICES:` |
| Response/Event | *status*:AVAILABLE_DEVICES:*dev1*:[*dev2*:…] |

A component responds to this command with a list of CPDID numbers of all available and CPDID-addressable components in the system, each separated by a colon. To be available a component must be powered on but the component may be in standby mode.

The directly connected component (CPDID 01) is always listed. If the directly connected component has an assigned CPDID (other than 01), the assigned CPDID will be listed too. Available components without CPDIDs are not listed; see `GET_AVAILABLE_DEVICES_BY_SERIAL_NUMBER` below.

After a list of CPDIDs is obtained, the `GET_NUM_ZONES` and `GET_DEVICE_INFO` commands can be used to get information about each component.

The `AVAILABLE_DEVICES` message is sent as an unsolicited event message when the list of available components changes.

Example 1 Event message

Kaleidescape System sends:

```
01/!/000:AVAILABLE_DEVICES:01:03:04:/68
01/!/000:AVAILABLE_DEVICES:01:03:04:05:/11
```

Example 2 Command/Response

Controller sends:

```
01/1/GET_AVAILABLE_DEVICES:
```

Kaleidescape System sends:

```
01/1/000:AVAILABLE_DEVICES:01:03:04:05:/27
```

In the first example, only the component directly connected (CPDID 01) and a component with CPDID 03 are active when a component with CPDID 04 is powered on, causing an `AVAILABLE_DEVICES` message to be generated with the new information. Shortly after that, a component with CPDID 05 is powered on, causing another `AVAILABLE_DEVICES` event message to be generated with the new list.

The second example shows the same information when requested by a `GET_AVAILABLE_DEVICES` command executed later.

### GET_AVAILABLE_DEVICES_BY_SERIAL_NUMBER

| | |
|---|---|
| Affects | All components |
| Command | `GET_AVAILABLE_DEVICES_BY_SERIAL_NUMBER:` |
| Response | *`status`*`:AVAILABLE_DEVICES_BY_SERIAL_NUMBER:`*`sn1`*`:[`*`sn2`*`:…]` |

A component responds to this command with a list of serial numbers of available components in the system, each separated by a colon. The serial numbers are zero-padded to 12 hexadecimals. To be available a component must be powered on but the component may be in standby mode.

After a list of serial number device identifiers is obtained, the `GET_NUM_ZONES` and `GET_DEVICE_INFO` commands can be used to get information about each component.

The `AVAILABLE_DEVICES_BY_SERIAL_NUMBER` message is sent as an unsolicited event message when the list of available components changes.

Example

Controller sends:

```
01/1/GET_AVAILABLE_DEVICES_BY_SERIAL_NUMBER:
```

Kaleidescape System sends:

```
01/1/000:AVAILABLE_DEVICES_BY_SERIAL_NUMBER:000000003638:
0000001CFF1B:/56
```

### GET_DEVICE_TYPE_NAME

| | |
|---|---|
| Affects | All components |
| Command | `GET_DEVICE_TYPE_NAME:` |
| Response | *`status`*`:DEVICE_TYPE_NAME:`*`device_name`*`:` |

A component responds to this command with its type name. This command can be used to generate messages displayed to the user.

| | |
|---|---|
| *`device_name`* | is the name of the Kaleidescape component, which is one of `Server`, `Cinema One`, `Strato`, `Alto`, `Terra Movie Server`, `Player`, `Music Player`, or `Disc Vault`. |

Example

Controller sends:

```
01/1/GET_DEVICE_TYPE_NAME:
```

Kaleidescape System sends:

```
01/1/000:DEVICE_TYPE_NAME:Music Player:/04
```

### GET_NUM_ZONES

| | |
|---|---|
| Affects | All components |
| Command | `GET_NUM_ZONES:` |
| Response | *status*:NUM_ZONES:*num_movie_zones*:<br>*num_music_zones*: |

This command is used to identify the number of zones for a component. A controller can then determine which commands a specific component supports.

| | |
|---|---|
| *num_movie_zones* | is `01` if there is an onscreen display associated with the component, and `00` if there is not. |
| *num_music_zones* | tells how many music zones are associated with the component. |

If the component has a movie zone, the first music zone corresponds to the audio outputs of that zone.

Example 1 How a Music Player responds

Controller sends:

```
01/1/GET_NUM_ZONES:
```

Kaleidescape System sends:

```
01/1/000:NUM_ZONES:00:04:/93
```

In this example, the Music Player does not have a movie zone, but has 4 music zones.

Example 2 How an M500 Player responds

Controller sends:

```
01/1/GET_NUM_ZONES:
```

Kaleidescape System sends:

```
01/1/000:NUM_ZONES:01:01:/91
```

In this example, the M500 Player has a single movie zone and a single music zone. The single music zone is the same zone as the movie zone.

### GET_SYSTEM_VERSION

| | |
|---|---|
| Affects | All components |
| Command | `GET_SYSTEM_VERSION:` |
| Response | *status*:SYSTEM_VERSION:*control_protocol_version*:<br>*kOS_version*: |

The response to this message shows the current version of the control protocol used by the Kaleidescape component and the version of kOS that is running.

| | |
|---|---|
| *control_protocol_version* | is a zero-padded, two-digit number representing the current protocol version. The version described in this document is 16. |
| *kOS_version* | is the string representation of the version number currently running on the device. |

Example

Controller sends:

```
01/1/GET_SYSTEM_VERSION:
```

Kaleidescape System sends:

```
01/1/000:SYSTEM_VERSION:16:10.4.2-19218:/91
```

This is an expected response from a component running the latest Kaleidescape software version (10.4.2).

## Protocol

### GET_PROTOCOL

| | |
|---|---|
| Affects | All components |
| Command | `GET_PROTOCOL:` |
| Response | *status*:PROTOCOL:*version*: |

The response to this message shows the current version of the control protocol used by the Kaleidescape component.

| | |
|---|---|
| *version* | is a zero-padded, two-digit number representing the current protocol version. The default is currently 16. |

Example

Controller sends:

```
01/1/GET_PROTOCOL:
```

Kaleidescape System sends:

```
01/1/000:PROTOCOL:16:/34
```

This is the expected response from a component running the latest Kaleidescape software.

> Note: This command replaces `GET_PROTOCOL_VERSION` which returned only the version number without the `PROTOCOL` message and was more difficult to use. The `GET_PROTOCOL_VERSION` command is still available, but `GET_PROTOCOL` is preferred.

### SET_PROTOCOL_SETTINGS

| | |
|---|---|
| Affects | All components |
| Command | `SET_PROTOCOL_SETTINGS:`*`delimiter_type`*`:`*`character_set`*`:` |
| Response | *`status`*`:` |

Normal control protocol messages transmit extended ASCII characters as escaped characters. For example, the character é is transmitted as `\d138`. Some controllers are unable to parse this format quickly. This command transmits extended ASCII characters in the clear (unescaped).

This command also replaces the default delimiters (colon, slash, and line feed) which are sometimes used in the message body of a control message. These characters are replaced with the binary delimiters SOH, STX, and EOT in all command responses and event messages. This change allows colons, slashes, and line feeds to appear unescaped in the message body for a simpler parsing algorithm.

> Note: Binary delimiters are not supported for control via RS-232. These settings are only valid for a given connection and are reset when that connection is terminated.

A controller receiving messages with binary delimiters should read messages until an EOT character (ASCII value 4) is found. Then the controller can parse the message by SOH characters (ASCII value 1) and STX characters (ASCII value 2) for message content. This command can only set the delimiters used for responses and events sent from the Kaleidescape component to the controller. Commands from the controller to the component must still be sent using printable delimiters.

| | |
|---|---|
| *`delimiter_type`* | `PRINTABLE_DELIMITERS`<br>Sends messages with the default printable characters, slash (`/`), colon (`:`) and carriage return (`\r\n`). |
| | `BINARY_DELIMITERS`<br>Sends messages with the binary delimiters SOH, STX, and EOT. |
| *`character_set`* | `LATIN-1`<br>Text fields can contain any printable character (ISO 8859-1). |

Table 7: Delimiter functions

| Printable delimiter | Binary delimiter | Function |
|---|---|---|
| / | SOH \x01 | Delimits the segments of a Kaleidescape control message. |
| : | STX \x02 | Delimits the fields within the message body of a control message. |
| \r\n | EOT \x04 | Delimits individual control messages. |

When using binary delimiters, / is replaced by SOH (ASCII value 1), the : is replaced by STX (ASCII value 2), and \r\n is replaced by EOT (ASCII value 4). The checksum is omitted to speed processing.

### Example

Normally, messages have the following format:

Controller sends:

```
01/1/GET_CONTENT_DETAILS:1.W_3675::
```

Kaleidescape System sends:

```
01/1/000:CONTENT_DETAILS:2:Album_title:Bart\d243k\:
Concerto for Orchestra; Music for Strings, Percussion and Celesta;
Hungarian Sketches:/81
```

However, after sending the `SET_PROTOCOL_SETTINGS` command, these messages have the following format.

Controller sends:

```
01/1/SET_PROTOCOL_SETTINGS:BINARY_DELIMITERS:LATIN-1:
01/1/GET_CONTENT_DETAILS:1.W_3675::
```

Kaleidescape System sends:

```
01(SOH)1(SOH)000(STX)CONTENT_DETAILS(STX)2(STX)Album_title
(STX)Bartók: Concerto for Orchestra; Music for Strings, Percussion
and Celesta; Hungarian Sketches(STX)(EOT)
```

## SET_SUPPORTED_PROTOCOL

| | |
|---|---|
| Affects | All components |
| Command | `SET_SUPPORTED_PROTOCOL:`*`version`*`:` |
| Response | *`status`*`:` |

This command sets a protocol version for the current session. This allows control protocol clients to enable newer protocol semantics for advanced kOS features.

If not used, the active session control protocol feature set is roughly limited to what is available in version 14 of the control protocol.

| | |
|---|---|
| *version* | is a zero-padded, two-digit number representing the current protocol version. |

Controller sends:

```
01/1/SET_SUPPORTED_PROTOCOL:15:
```

Kaleidescape System sends:

```
01/1/000:/89
```

### GET_ACTIVE_PROTOCOL

| | |
|---|---|
| Affects | All components |
| Command | `GET_ACTIVE_PROTOCOL:` |
| Response | *`status`*`:ACTIVE_PROTOCOL:`*`version`*`:` |

This command reports a subset of control protocol features for the established control protocol session.

If required, the `SET_SUPPORTED_PROTOCOL` command can be used to limit or expand the session feature set. For reference, use the `GET_PROTOCOL` command to identify the most capable protocol version supported by the component.

Controller sends:

```
01/1/GET_ACTIVE_PROTOCOL:
```

Kaleidescape System sends:

```
01/1/000:ACTIVE_PROTOCOL:14:/71
```

## Event message registration

### ENABLE_EVENTS

| | |
|---|---|
| Affects | All components |
| Command | `ENABLE_EVENTS:`*`target_device_id`*`:` |
| Response | *`status`*`:` |

Enables event messages for the zone identified by *`target_device_id`*. The command must be sent to the directly connected component (CPDID 01). Use the `DISABLE_EVENTS` command to turn off events from another component.

*target_device_id* is the device ID for the zone from which the controller wants to receive event messages.

This must include the actual CPDID or serial number of the component referenced. If enabling events from a zone on the component directly connected, CPDID 01 can only be used if the component does not have an assigned CPDID.

A controller normally receives events only from the component with a direct connection. This allows controllers that connect to a single component to use simple logic in handling event messages, because the controller does not see events from any other zones in the system. To see events from other zones, send an `ENABLE_EVENTS` command for each zone. The `GET_NUM_ZONES` and `GET_AVAILABLE_DEVICES` commands can be used to discover which routable IDs exist and which zones are associated with each ID.

Once enabled, the device ID for incoming events from the designated component matches the format of the *target_device_id* (i.e., if the *target_device_id* is in serial format, event messages are prefaced with the serial number of the component).

Example 1

A controller that handles events from all active movie zones might issue the following messages.

Controller sends:

```
01/1/GET_AVAILABLE_DEVICES:
```

Kaleidescape System sends:

```
01/1/000:AVAILABLE_DEVICES:01:09:/16
```

Controller sends:

```
01/2/GET_NUM_ZONES:
```

Kaleidescape System sends:

```
01/2/000:NUM_ZONES:00:00:/90
```

Controller sends:

```
09/3/GET_NUM_ZONES:
```

Kaleidescape System sends:

```
09/3/000:NUM_ZONES:01:01:/01
```

Controller sends:

```
01/4/ENABLE_EVENTS:09:
```

The first field of the `NUM_ZONES` response indicates how many movie zones are available. The first component does not have a movie zone, the second component does. The controller checks this value and issues an `ENABLE_EVENTS` command for each component that has a movie zone.

Example 2

The *target_device_id* can also be specified as the serial number of the device whose events are to be displayed, preceded by a pound sign. To receive all events for component with serial number 0000 0000144B, submit the following request:

```
01/1/ENABLE_EVENTS:#144B:
```

Events from the device directly connected are normally prefaced by the CPDID, or CPDID 01 if the device does not have an assigned CPDID. To receive these events in serial format, disable events for the device directly connected, and enable events using the serial number of the connected device.

For component with serial number 0000 0000144B with no assigned CPDID, the following commands would accomplish this.

```
01/1/ENABLE_EVENTS:#144B:
01/1/DISABLE_EVENTS:01:
```

All subsequent events will be in this format.

```
#00000000144B/!/000:TITLE_NAME:The English Patient:/92
```

Example 3

The music zone identifier can be appended to the target device ID to receive events for a particular music zone. For example, to receive all music-oriented events from an M500 Player with CPDID 35, to which the controller is directly connected, issue this command.

```
01/1/ENABLE_EVENTS:35.01:
```

The `35` in the field is the CPDID of the directly connected component (the M500 Player). The `01` is the music zone identifier. Note that this example does not use CPDID 01 even though the controller is directly connected to the M500 Player.

Example 4

To receive events from music zone 03 on a Music Player with serial number 0000 00001E88, issue this command.

```
01/1/ENABLE_EVENTS:#1E88.03:
```

The serial number format with the optional music zone can be mixed and matched.

### DISABLE_EVENTS

| | |
|---|---|
| Affects | All components |
| Command | `DISABLE_EVENTS:`*`target_device_id`*`:` |
| Response | *`status`*`:` |

Disables event messages from the zone referenced. The command must be sent to the directly connected component (CPDID 01).

*`target_device_id`* identifies the zone from which to disable event messages. The format is identical to the format for `ENABLE_EVENTS`.

Example

Kaleidescape System sends:

```
10/!/000:TITLE_NAME:24 (Season 1):/72
10/!/000:MOVIE_MEDIA_TYPE:01:/34
10/!/000:MOVIE_LOCATION:03:/68
```

Controller sends:

```
01/1/DISABLE_EVENTS:10:
```

Kaleidescape System sends:

```
01/1/000:/89
```

In this example, events are received for a component with CPDID 10, in this case stating that movie playback has begun.

When this information is no longer required, sending the `DISABLE_EVENTS` command to the local device tells the device that event messages from CPDID 10 are no longer required.

### GET_DEVICE_INFO

| | |
|---|---|
| Affects | All components |
| Command | `GET_DEVICE_INFO:` |
| Response | *`status`*`:DEVICE_INFO:`*`device_type`*`:`*`serial_num`*`:`<br>*`cpdid`*`:`*`ip_address`*`:` |

A component responds to this command with information about the component.

*`device_type`* was previously used to identify the capabilities of the component. This value no longer provides sufficient information. Instead, use `GET_NUM_ZONES` to identify the capabilities of the component and use `GET_DEVICE_TYPE_NAME` to get the type name of the component.

| | |
|---|---|
| *serial_num* | is the serial number of the component that received the request. *serial_num* is zero-padded to 16 hexadecimals. |
| *cpdid* | is the assigned device identifier of the component. |
| *ip_address* | is the network TCP/IP address of the component. |

Note: The returned `cpdid` is the assigned CPDID even if the command was sent to `01`. If no CPDID has been assigned to the component, `00` is returned instead.

Example

Controller sends:

```
01/1/GET_DEVICE_INFO:
```

Kaleidescape System sends:

```
01/1/000:DEVICE_INFO:11:000000000018E6D6:00:010.100.012.194:/63
```

In this example, the results of the `GET_DEVICE_INFO` command show that the device type is 11 (deprecated information), the serial number of the component is 0000 0018E6D6, there is no assigned CPDID, and the component has IP address 10.100.12.194.

## Module registration

SEND_TO_SYSLOG

| | |
|---|---|
| Affects | All components |
| Command | `SEND_TO_SYSLOG:INFORMATION:`*message*`:` |
| Response | *status* |

This message is used to post information to the Kaleidescape System log. This information is then used by Kaleidescape to track modules in the field and is included in all control system modules provided by Kaleidescape.

| | |
|---|---|
| *message* | is the string logged by the Kaleidescape System. |

Note: Any information sent to Kaleidescape logs is covered by the Kaleidescape privacy policy.

Example

Controller sends:

```
01/1/SEND_TO_SYSLOG:INFORMATION:OSD Control Module version 8.2:
```

Kaleidescape System sends:

```
01/1/000:/89
```

This example registers a module with the name OSD Control Module and version number 8.2. This information can be used by Kaleidescape Support for troubleshooting.

## Friendly name

### GET_FRIENDLY_NAME

| | |
|---|---|
| Affects | Any zone |
| Command | `GET_FRIENDLY_NAME:` |
| Response | *`status`*`:FRIENDLY_NAME:`*`name`*`:` |

The response to this command provides the friendly name of the zone or component.

| | |
|---|---|
| *name* | is the music zone friendly name when sent to a music zone. If sent to a movie zone, *name* is the friendly name of the component. |

On Premiere line systems, the friendly name is set on the Components tab in the browser interface or by using the `SET_FRIENDLY_NAME` command. On the Components tab, the friendly name is set in the Zone n Name (for music) text box for music zones, and in the component Device Name text box for movie zones. For Encore players and Cinema One (2nd generation), the friendly name is the player name which is set on the Player Name settings page on the onscreen display.

Example

Controller sends:

```
01/1/GET_FRIENDLY_NAME:
```

Kaleidescape System sends:

```
01/1/000:FRIENDLY_NAME:Dining Room Player:/93
```

Controller sends:

```
01.01/1/GET_FRIENDLY_NAME:
```

Kaleidescape System sends:

```
01.01/1/000:FRIENDLY_NAME:Dining Room Music:/28
```

In this example, two `GET_FRIENDLY_NAME` commands are sent to an M500 Player serving music in the dining room. In the first command, the controller is requesting the name of the directly connected component, which is Dining Room Player. In the second example, the controller is requesting the friendly name of the music zone for the same player, which is Dining Room Music.

### SET_FRIENDLY_NAME

| | |
|---|---|
| Affects | Any zone |
| Command | `SET_FRIENDLY_NAME:`*`name`*`:` |
| Response | *`status`*`:FRIENDLY_NAME:`*`name`*`:` |

Sets the friendly name of the zone or component to the string in the name field.

If sent to a music zone, the friendly name of the music zone is changed. If sent to a movie zone, the friendly name of the component is changed.

| | |
|---|---|
| *name* | is the friendly name to assign to the zone or component. |

The friendly name is remembered even if the component is turned off. On Premiere line systems, the friendly name can also be configured using the Components tab in the browser interface. For Strato players, the friendly name can be set on the Settings - Components tab of the browser interface. For Alto and Cinema One (2nd generation), the friendly name can be set on the Player Name settings page of the onscreen display.

The response to this command contains the new friendly name for the zone or component; see `GET_FRIENDLY_NAME` for a description of the response message.

Example

Controller sends:

```
01/1/SET_FRIENDLY_NAME:Dining Room Player:
```

Kaleidescape System sends:

```
01/1/000:FRIENDLY_NAME:Dining Room Player:/93
```

Controller sends:

```
01.01/1/SET_FRIENDLY_NAME:Dining Room Music:
```

Kaleidescape System sends:

```
01.01/1/000:FRIENDLY_NAME:Dining Room Music:/28
```

In this example, a controller is making changes to the friendly names associated with an M500 Player serving the dining room. The first command changes the name of the component itself, to Dining Room Player. The second command changes the name of the music zone for the same player to Dining Room Music.

### GET_FRIENDLY_SYSTEM_NAME

| | |
|---|---|
| Affects | Any zone |
| Command | `GET_FRIENDLY_SYSTEM_NAME:` |
| Response | *`status`*`:FRIENDLY_SYSTEM_NAME:`*`name`*`:` |

The response to this command provides the friendly system name of which the target component is a part.

| | |
|---|---|
| *name* | is the system name assigned during system registration. |

On Premiere line systems, the friendly system name is set initially on system registration but can be changed on the Settings tab in the browser interface or on the Kaleidescape Movie Store under the Manage - Account - My Systems page. For Encore systems the friendly system name is initially set at registration and can be changed at the above mentioned online movie store page.

Example

Controller sends:

```
01/1/GET_FRIENDLY_SYSTEM_NAME:
```

Kaleidescape System sends:

```
01/1/000:FRIENDLY_SYSTEM_NAME:My Encore System:/85
```

In this example, the `GET_FRIENDLY_SYSTEM_NAME` command is sent to a Strato Movie Player. The controller is requesting the name of the system this player is a part of, which is My Encore System.

## OSD Control

The onscreen display (OSD) control commands navigate the Kaleidescape onscreen display, control movie and music playback through the onscreen display, and respond to playback events. Commands are grouped with a detailed description of each command. Table 8 lists onscreen display control commands.

Table 8: OSD Control command summary

| Command | Description |
|---|---|
| Basic navigation | |
| Arrow commands | Used to navigate the onscreen display. |
| Page up/down commands | Used to navigate by pages on the onscreen display. |
| `SELECT` | Selects the highlighted item in the onscreen display. |
| `CANCEL` | Dismisses a page, dialog, or text entry. |
| `POSITION_SELECT` | Transmits touch screen interaction to the onscreen display. |
| `CHILD_SELECT` | Selects the highlighted item in the onscreen display. Also activates the child user interface. |
| Kaleidescape menu | |
| `KALEIDESCAPE_MENU_ON` | Displays Kaleidescape menu. |
| `KALEIDESCAPE_MENU_OFF` | Removes Kaleidescape menu. |
| `KALEIDESCAPE_MENU_TOGGLE` | Toggles Kaleidescape menu on and off. |

| Command | Description |
|---|---|
| Views | |
| GET_UI_STATE | Provides details about the current state of the user interface. |
| Movie views | |
| GO_MOVIES | Changes the interface from a music view to the corresponding movie view. |
| GO_MOVIE_LIST | Displays the Movie List view. |
| GO_MOVIE_COVERS | Displays the Movie Covers view. |
| GO_MOVIE_COLLECTIONS | Displays the Movie Collections view. |
| GO_MOVIE_COLLECTION | Displays a specific collection in the Movie Collections view. |
| Music views | |
| GO_MUSIC | Changes the interface from a movie view to the corresponding music view. |
| GO_MUSIC_LIST | Displays the Music List view. |
| GO_MUSIC_COVERS | Displays the Music Covers view. |
| GO_MUSIC_COLLECTIONS | Displays the Music Collections view. |
| GO_MUSIC_COLLECTION | Displays a specific collection in the Music Collections view. |
| GO_NOW_PLAYING | Displays the Now Playing view. |
| Other views | |
| GO_MOVIE_STORE | Displays the Kaleidescape Movie Store view. |
| GO_PARENTAL_CONTROL | Displays the Parental Control view. |
| GO_SYSTEM_STATUS | Displays the System Status view. |
| GO_VAULT_SUMMARY | Displays the Vault Summary view. |

| Command | Description |
|---|---|
| User input | |
| GET_USER_INPUT | Provides information about user input requested from the user interface. |
| GET_USER_INPUT_PROMPT | Returns user input prompt details currently shown on screen. |
| SET_USER_INPUT_ENTRY | Sends a complete string to the on-screen prompt. |
| KEYBOARD_CHARACTER | Sends a single character to the onscreen display. |
| KEYBOARD_CHARACTER | Sends a single character to the onscreen keyboard. |
| BACKSPACE | Erases the last character entered. |
| View-specific commands | |
| FILTER_LIST | Filters the list view to search criteria. |
| SHUFFLE_COVER_ART | Shuffles cover art on covers view. |
| CHILD_SHUFFLE_COVER_ART | Shuffles cover art on the child user interface if the child user interface is displayed. If not, displays child user interface. |
| ALPHABETIZE_COVER_ART | Arranges covers alphabetically. |
| DEFAULT_LEVEL | Changes the parental control level to the default level. |
| SAFE_LEVEL | Changes parental control to highest level without a passcode. |
| Page and content details | |
| DETAILS | Toggles between the details page and the current display. |
| DISC_IN_TRAY_TOGGLE | Toggles the disc in player details |
| GET_CONTENT_DETAILS | Provides information about a movie or album selected. |
| GET_HIGHLIGHTED_SELECTION | Returns the handle of the movie or album currently selected on the user interface. |

| Command | Description |
|---|---|
| Screen saver commands | |
| `GO_SCREEN_SAVER` | Displays the screen saver. |
| `STOP_SCREEN_SAVER` | Removes screen saver. |

## Basic navigation commands

### Arrow commands

| | | |
|---|---|---|
| `UP_PRESS` | `UP_RELEASE` | `UP` |
| `DOWN_PRESS` | `DOWN_RELEASE` | `DOWN` |
| `LEFT_PRESS` | `LEFT_RELEASE` | `LEFT` |
| `RIGHT_PRESS` | `RIGHT_RELEASE` | `RIGHT` |
| `CHILD_UP_PRESS` | `CHILD_UP_RELEASE` | `CHILD_UP` |
| `CHILD_DOWN_PRESS` | `CHILD_DOWN_RELEASE` | `CHILD_DOWN` |
| `CHILD_LEFT_PRESS` | `CHILD_LEFT_RELEASE` | `CHILD_LEFT` |
| `CHILD_RIGHT_PRESS` | `CHILD_RIGHT_RELEASE` | `CHILD_RIGHT` |

| | |
|---|---|
| Affects | Any movie zone |
| Command | `UP_PRESS:` (same for other arrow commands) |
| Response | *`status`*`:` |

These commands send directional movement to the onscreen display to navigate menus and lists. During movie playback, these commands navigate DVD or Blu-ray Disc menus and handle interactive content.

The `_PRESS` and `_RELEASE` versions of these commands allow the onscreen display to handle auto-repeat when the user holds down a button for continuous scrolling through lists. Use these commands, instead of the plain directional commands if the controller supports press and release handling.

Send a `_PRESS` command when the corresponding button is pressed and a `_RELEASE` command when the button is released. If a command is sent that affects the user interface between the `_PRESS` and `_RELEASE` commands, including a different direction command, the auto-repeat is canceled and the command that interrupted the repeat is handled normally. Informational commands such as `GET_PLAYING_TITLE_NAME` do not cancel auto-repeat.

Plain directional commands, `UP`, `DOWN`, `LEFT`, and `RIGHT` are available for controllers that do not support press and release handling. Each conveys a single movement in one direction. The controller can be programmed to send a plain directional command repeatedly at some fixed interval to simulate continuous scrolling.

`CHILD_` commands activate the child user interface if not already active, and navigate the child user interface once the child user interface is active.

Example 1

Controller sends:

```
01/1/DOWN:
```

Kaleidescape System sends:

```
01/1/000:/89
```

Controller sends:

```
01/2/DOWN:
```

Kaleidescape System sends:

```
01/2/000:/90
02/!/000:HIGHLIGHTED_SELECTION:1.0-S_1baaf:/73
```

Example 2

Controller sends:

```
01/1/DOWN_PRESS:
```

Kaleidescape System sends:

```
01/1/000:/89
```

Controller sends:

```
01/2/DOWN_RELEASE:
```

Kaleidescape System sends:

```
01/2/000:/90
```

In the first example, two `DOWN` commands are sent, resulting in the current selection moving down two items. In the second example, a `DOWN_PRESS` is followed by a `DOWN_RELEASE`, which results in the current selection moving down one or more items, depending on the time between sending the two commands.

Example 3

Controller sends:

```
01/1/CHILD_RIGHT:
```

Kaleidescape System sends:

```
01/1/000:/89
02/!/000:CHILD_MODE_STATE:1:/63
02/!/000:USER_DEFINED_EVENT:SELECT_KALEIDESCAPE_INPUT:/77
02/!/000:HIGHLIGHTED_SELECTION:1.0-S_40f4:/84
```

The player was not previously displaying the child user interface, but after receiving the `CHILD_RIGHT` command, switched to the child user interface and emitted the appropriate events.

Page up/down commands

| | | |
|---|---|---|
| `PAGE_UP_PRESS` | `PAGE_UP_RELEASE` | `PAGE_UP` |

| | | |
|---|---|---|
| PAGE_DOWN_PRESS | PAGE_DOWN_RELEASE | PAGE_DOWN |

| | |
|---|---|
| Affects | Any movie zone |
| Command | PAGE_UP_PRESS: (same for other page up/down commands) |
| Response | *status*: |

These commands cause the onscreen display to move up and down in lists by entire pages. Unlike directional arrows, page up/down commands are not passed to any DVD or Blu-ray content currently playing.

Context-Sensitive Commands are available for controllers that do not have dedicated Page Up/Page Down buttons. These commands scroll a page in the user interface or skip forward/backward through movie chapters or music tracks.

Example 1

Controller sends:

```
01/8/PAGE_UP:
```

Kaleidescape System sends:

```
01/8/000:/96
02/!/000:HIGHLIGHTED_SELECTION:1.0-S_1baaf:/73
```

Example 2

Controller sends:

```
01/9/PAGE_UP_PRESS:
```

Kaleidescape System sends:

```
01/9/000:/97
```

Controller sends:

```
01/0/PAGE_UP_RELEASE:
```

Kaleidescape System sends:

```
01/0/000:/88
```

The first example shows the PAGE_UP being sent to the component with a response indicating success and an event indicating a new item is selected.

The second example shows a PAGE_UP_PRESS command followed by a PAGE_UP_RELEASE.

### SELECT

| | |
|---|---|
| Affects | Any movie zone |
| Command | `SELECT:` |
| Response | *`status`*`:` |

Selects the highlighted item in the onscreen display. When a movie or album is already highlighted, the details page for that movie or album usually appears. In the Movie List and Music List views, if the highlight is on a column not currently sorted, this command sorts by that column. When highlighting a menu item, submitting `SELECT` performs the action for that item. In some contexts, such as passcode entry, this command indicates that the passcode has been entered. This command can be entered from an Enter button on a numeric keypad or keyboard.

During playback this command is passed to the DVD or Blu-ray Disc playing, which allows the user to interact with menus and interactive disc features.

Example

Controller sends:

```
01/0/SELECT:
```

Kaleidescape System sends:

```
01/0/000:/88
02/!/000:UI_STATE:01:01:00:0:/40
```

In this example, the `SELECT` command is sent while the onscreen display is in the Movie List view and on a specific movie. The `UI_STATE` event message is generated as the details page for that movie appears on the screen.

### CANCEL

| | |
|---|---|
| Affects | Any movie zone |
| Command | `CANCEL:` |
| Response | *`status`*`:` |

In the onscreen display, dismisses a page, dialog, or text entry. See also the `STOP_OR_CANCEL` context-sensitive command.

Example

Controller sends:

```
01/4/CANCEL:
```

Kaleidescape System sends:

```
01/4/000:/92
02/!/000:UI_STATE:01:00:00:0:/39
```

In this example, the onscreen display is in the Movie List view and the details page for a movie is displayed. Sending the `CANCEL` command dismisses the details page. A `UI_STATE` message is generated to indicate the change.

### POSITION_SELECT

| | |
|---|---|
| Affects | Any movie zone |
| Command | `POSITION_SELECT:`*`x_loc`*`:`*`y_loc`*`:` |
| Response | *`status`*`:` |

Sends the onscreen display a touch event at coordinates *x_loc* and *y_loc*.

*x_loc*, *y_loc* identify the location that was touched on the screen. (Can be any ASCII decimal integers from 0 to 2 billion.)

A controller can use this command to transmit user touches on a video feed to the onscreen display, allowing direct manipulation of screen elements such as cover images, tabs, and list view columns.

The values a controller sends for *x_loc* and *y_loc* can be relative to the origin and scale required (the full screen of the touch panel, the frame of the video feed, or something else). When the component is calibrated to work with a touch panel using the onscreen display, the component determines how to interpret the coordinates the controller sends.

The onscreen display can be calibrated for the touch panel through the System Status view. Select the System Setup tab, then select Calibrate Touch Panel.

If the onscreen display has not been calibrated to work with a touch panel, this command acts as a plain `SELECT` command.

Example

Controller sends:

```
01/2/POSITION_SELECT:220:500:
```

Kaleidescape System sends:

```
01/2/000:/90
```

This example sends a `POSITION_SELECT` command at the coordinates 220, 500 to the onscreen display. The onscreen display uses prior calibration to determine what was touched on the video and reacts appropriately.

### CHILD_SELECT

| | |
|---|---|
| Affects | Any movie zone |
| Command | `CHILD_SELECT:` |
| Response | *`status`*`:` |

This command functions like the `SELECT` command. If the onscreen display was not displaying the child user interface, the player switches to the child user interface. If the selected media is available in the Child collection, playback begins; otherwise, only the transition to the child user interface occurs.

Example

Controller sends:

```
01/7/CHILD_SELECT:
```

Kaleidescape System sends:

```
01/7/000:/95
02/!/000:CHILD_MODE_STATE:1:/63
02/!/000:USER_DEFINED_EVENT:SELECT_KALEIDESCAPE_INPUT:/77
02/!/000:MUSIC_NOW_PLAYING_STATUS:00000:00000:1:0:0000000010::/68
02/!/000:UI_STATE:07:00:00:0:/45
02/!/000:TITLE_NAME:Ice Age:/34
```

The player was not displaying the child user interface. The movie selected was also available in the Child collection. After CHILD_SELECT was sent, the player activated the child user interface, and began playing the selection.

## Kaleidescape menu commands

### KALEIDESCAPE_MENU_ON

| | |
|---|---|
| Affects | Any movie zone |
| Command | KALEIDESCAPE_MENU_ON: |
| Response | *status*: |

Displays the Kaleidescape menu if not already onscreen; otherwise this command has no effect. In either case, a status message is returned.

Any active movie playback continues behind the menu until the user chooses an item or a control command interrupting playback is received. If the menu is dismissed, playback simply continues. KALEIDESCAPE_MENU_TOGGLE can also be used to display the menu when the menu is not currently displayed.

The context-sensitive command DISC_OR_KALEIDESCAPE_MENU can be used to display the menu when movie playback is not active. See also KALEIDESCAPE_MENU_OFF.

Example

Controller sends:

```
01/4/KALEIDESCAPE_MENU_ON:
```

Kaleidescape System sends:

```
01/4/000:/92
01/!/000:UI_STATE:08:00:01:0:/46
```

In this example, the KALEIDESCAPE_MENU_ON command is sent while onscreen display is on the System Status menu. A UI_STATE event message is generated indicating that the Kaleidescape menu is now displayed.

### KALEIDESCAPE_MENU_OFF

| | |
|---|---|
| Affects | Any movie zone |
| Command | `KALEIDESCAPE_MENU_OFF:` |
| Response | *`status`*`:` |

Removes the Kaleidescape menu if onscreen; otherwise has no effect. In either case a status message is returned. If the menu is already displayed, `CANCEL`, `STOP_OR_CANCEL`, `KALEIDESCAPE_MENU_TOGGLE`, and `DISC_OR_KALEIDESCAPE_MENU` commands also dismiss the menu.

Example

Controller sends:

```
01/1/KALEIDESCAPE_MENU_OFF:
```

Kaleidescape System sends:

```
01/1/000:/89
01/!/000:UI_STATE:03:00:00:0:/40
```

Controller sends:

```
01/2/KALEIDESCAPE_MENU_OFF:
```

Kaleidescape System sends:

```
01/2/000:/90
```

In this example, `KALEIDESCAPE_MENU_OFF` is called twice. The first time, the Kaleidescape menu was previously displayed and when turned off, caused the `UI_STATE` event to be sent. The second time, the Kaleidescape menu was already off, so no event was sent.

### KALEIDESCAPE_MENU_TOGGLE

| | |
|---|---|
| Affects | Any movie zone |
| Command | `KALEIDESCAPE_MENU_TOGGLE:` |
| Response | *`status`*`:` |

Toggles the display of the Kaleidescape menu as described in `KALEIDESCAPE_MENU_ON` and `KALEIDESCAPE_MENU_OFF`. When the menu is not displayed, this command causes the menu to be displayed. When the menu is displayed on screen, this command dismisses the menu.

Example

Controller sends:

```
01/4/KALEIDESCAPE_MENU_TOGGLE:
```

Kaleidescape System sends:

```
01/4/000:/92
01/!/000:UI_STATE:03:00:00:0:/40
```

Controller sends:

```
01/5/KALEIDESCAPE_MENU_TOGGLE:
```

Kaleidescape System sends:

```
01/5/000:/93
01/!/000:UI_STATE:03:00:01:0:/41
```

In this example, the `KALEIDECAPE_MENU_TOGGLE` message is sent twice. The first time removes the Kaleidescape menu and returns to the Movie Covers view, confirmed by the `UI_STATE` event message generated. The second time causes the menu to be displayed again.

## Views

### GET_UI_STATE

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GET_UI_STATE:` |
| Response/Event | *status*:UI_STATE:*screen*:*popup*:*dialog*:*saver*: |

The `UI_STATE` message provides information about the current state of the user interface, including which view is active, what pages, menus and dialogs are displayed, and whether or not the screen saver is active.

The event message is generated when any of these conditions change.

The information in this message can be used to provide feedback to the user about a state.

*screen* identifies the view currently active.

| | |
|---|---|
| 00 | Unknown |
| 01 | Movie List |
| 02 | Movie Collections |
| 03 | Movie Covers |
| 04 | Parental Control |
| 05 | unused |
| 06 | unused |
| 07 | Playing a movie |
| 08 | System Status |
| 09 | Music List |
| 10 | Music Covers |
| 11 | Music Collections |
| 12 | Music Now Playing |
| 13 | unused |
| 14 | Vault Summary |
| 15 | System Settings |
| 16 | Movie Store |
| 17 | Paired unit lobby |

*popup* identifies any page or menu that appears to display information or settings.

| | |
|---|---|
| `00` | No page or menu |
| `01` | Details page |
| `02` | Movie overlay displaying the status page (appears only during playback) |
| `03` | Movie overlay, but not status page (appears only during playback) |

*dialog* identifies a floating message box.

| | |
|---|---|
| `00` | No dialog |
| `01` | Kaleidescape menu |
| `02` | Passcode entry (the controller should display a numeric keypad if necessary) |
| `03` | Simple question |
| `04` | Informational message (such as system upgrade) |
| `05` | Warning message |
| `06` | Error message (such as a network error) |
| `07` | Preplay |
| `08` | Import_Warranty |
| `09` | Keyboard |
| `10` | IP configuration |

*saver* indicates whether the screen saver is active.

| | |
|---|---|
| `0` | Screen saver inactive |
| `1` | Screen saver active |

Example

Controller sends:

```
01/5/GET_UI_STATE:
```

Kaleidescape System sends:

```
01/5/000:UI_STATE:07:01:00:0:/65
```

The response indicates that a movie was playing back and a movie details page was being displayed over the movie.

## Movie views

### GO_MOVIES

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GO_MOVIES:` |
| Response | *status*: |

Displays a movie view on the onscreen display.

If the interface is displaying a music view, changes to the equivalent movie view. That is, changes from the Music List view to the Movie List view, from the Music Collections view to the Movie Collections view, and from the Music Covers view to the Movie Covers view.

If there is no equivalent movie view (i.e., Vault Summary, Now Playing, System Status), the OSD displays the last movie view that appeared. If no movie view has been displayed since booting, the Movie List view is displayed.

If sent during movie playback, playback is halted and the current playback location is saved in the Paused list.

Example

Controller sends:

```
01/3/GO_MOVIES:
```

Kaleidescape System sends:

```
01/3/000:/91
02/!/000:UI_STATE:01:00:00:0:/39
02/!/000:HIGHLIGHTED_SELECTION:1.0-S_4c4de:/34
```

In this example, the onscreen display is showing the Music List view when the `GO_MOVIES` command is sent. This command causes the Movie List view to be displayed and a `UI_STATE` event message to be generated acknowledging the change. A `HIGHLIGHTED_SELECTION` message is also generated with the handle of the newly-highlighted movie.

### GO_MOVIE_LIST

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GO_MOVIE_LIST:` |
| Response | *`status`*`:` |

Displays the Movie List view. If sent during movie playback, playback is halted and the current playback location is saved in the Paused list.

While in the Movie List view, the `FILTER_LIST` command can be used to search the list, and the `KEYBOARD_CHARACTER` command can be used to jump to specific letters in the alphabet.

Example

Controller sends:

```
01/7/GO_MOVIE_LIST:
```

Kaleidescape System sends:

```
01/7/000:/95
02/!/000:UI_STATE:01:00:00:0:/39
```

In this example, the onscreen display is showing the Movie Covers view when the `GO_MOVIE_LIST` command is sent. This command causes the display to change to the Movie List view and generates a `UI_STATE` event message with that information. Because the same movie remains highlighted, a `HIGHLIGHTED_SELECTION` event message is not generated.

### GO_MOVIE_COVERS

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GO_MOVIE_COVERS:` |
| Response | *`status`*`:` |

Displays the Movie Covers view. If sent during movie playback, playback is halted and the current playback location is saved in the Paused list.

While in the Movie Covers view, the `SHUFFLE_COVER_ART` and `ALPHABETIZE_COVER_ART` commands can be used to reorganize the cover art.

Example

Controller sends:

```
01/8/GO_MOVIE_COVERS:
```

Kaleidescape System sends:

```
01/8/000:/96
02/!/000:UI_STATE:03:00:00:0:/41
```

In this example, the onscreen display is showing the Movie List view when the `GO_MOVIE_COVERS` command is sent. This command causes the display to switch to the Movie Covers view and generate a `UI_STATE` event message confirming the new view. A `HIGHLIGHTED_SELECTION` event message is not generated because the same movie remains highlighted.

> Note: This command replaces the `GO_COVER_ART` command, which is still supported.

### GO_MOVIE_COLLECTIONS

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GO_MOVIE_COLLECTIONS:` |
| Response | *`status`*`:` |

Displays the Movie Collections view. If sent during movie playback, playback is halted and the current playback location is saved in the Paused list. While in the Movie Collections view, the `FILTER_LIST` command can be used to search the selected collection, and the `KEYBOARD_CHARACTER` command can be used to jump to specific letters in the alphabet.

Example

Controller sends:

```
01/3/GO_MOVIE_COLLECTIONS:
```

Kaleidescape System sends:

```
01/3/000:/91
02/!/000:UI_STATE:02:00:00:0:/40
02/!/000:HIGHLIGHTED_SELECTION:1.0-u_3877:/81
```

In this example, the onscreen display is showing the Movie List view when the `GO_MOVIE_COLLECTIONS` command is sent. This command causes the display to switch to the Movie Collections view and generate a `UI_STATE` event message confirming the new view. A `HIGHLIGHTED_SELECTION` message is generated in this example because a new movie is highlighted.

> Note: This command replaces the `GO_COLLECTIONS` command and the `GO_FAVORITES` command; however, both are still supported.

GO_MOVIE_COLLECTION

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GO_MOVIE_COLLECTION:`*`collection_name`*`:` |
| Response | *`status`*`:` |

Displays the Movie Collections view and selects a collection for onscreen display.

*`collection_name`* is the name of the predefined or user-defined collection.

If the *`collection_name`* does not match one of the movie collections in the system exactly, this command acts like the `GO_MOVIE_COLLECTIONS` command.

If this command is sent during movie playback, playback is halted and the current playback location is saved in the Paused list.

Example

Controller sends:

```
01/9/GO_MOVIE_COLLECTION:Favorites:
```

Kaleidescape System sends:

```
01/9/000:/97
02/!/000:HIGHLIGHTED_SELECTION:1.0-u_3877:/81
```

In this example, the `GO_MOVIE_COLLECTIONS:Favorites:` command was sent while the onscreen display was on another collection. The command causes the collection currently active to change to the Favorites collection. Because this change causes a new movie to be highlighted, a `HIGHLIGHTED_SELECTION` event message is sent.

## Music views

> Note: Music view commands will re-direct to the movie collection “Songs” when sent to a player which does not support music.

### GO_MUSIC

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GO_MUSIC:` |
| Response | *`status`*`:` |

Displays a music view on the onscreen display.

If the interface is displaying a movie view, changes to the equivalent music view. That is, changes from the Movie List view to the Music List view, from the Movie Collections view to the Music Collections view, and from the Movie Covers view to the Music Covers view.

If there is no equivalent music view (i.e., Vault Summary, Parental Control, System Status), the OSD displays the last music view that appeared. If no music view has been displayed since booting, the Music List view is displayed.

If sent during movie playback, playback is halted and the current playback location is saved in the Paused list.

Example

Controller sends:

```
01/0/GO_MUSIC:
```

Kaleidescape System sends:

```
01/0/000:/88
02/!/000:UI_STATE:10:00:00:0:/39
02/!/000:HIGHLIGHTED_SELECTION:1.w_3675:/54
```

In this example, the onscreen display was on the Movie Covers view when the `GO_MUSIC` command was sent. This command causes the view to switch to the Music Covers view and a `UI_STATE` event message is generated confirming the change. Because the highlighted selection changed from a movie to an album, a `HIGHLIGHTED_SELECTION` event message is generated with the handle of the newly-selected album.

### GO_MUSIC_LIST

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GO_MUSIC_LIST:` |
| Response | *`status`*`:` |

Displays the Music List view. If sent during movie playback, playback is halted and the current playback location is saved in the Paused list. While in the Music List view, the `FILTER_LIST` command can be used to search the list, and the `KEYBOARD_CHARACTER` command can be used to jump to specific letters in the alphabet.

Example

Controller sends:

```
01/7/GO_MUSIC_LIST:
```

Kaleidescape System sends:

```
01/7/000:/95
01/!/000:UI_STATE:09:00:00:0:/46
```

In this example, the onscreen display is showing the Music Covers view when the GO_MUSIC_LIST command is sent. This command causes the display to change to the Music List view, and generates a UI_STATE event message. Because the same album remains highlighted, no HIGHLIGHTED_SELECTION event message is generated.

### GO_MUSIC_COVERS

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GO_MUSIC_COVERS:` |
| Response | *`status`*`:` |

Displays the Music Covers view. If sent during movie playback, playback is halted and the current playback location is saved in the Paused list.

While in the Movie Covers view, the SHUFFLE_COVER_ART and ALPHABETIZE_COVER_ART commands can be used to reorganize cover art.

Example

Controller sends:

```
01/8/GO_MUSIC_COVERS:
```

Kaleidescape System sends:

```
01/8/000:/96
01/!/000:UI_STATE:10:00:00:0:/38
```

In this example, the onscreen display is showing the Music List view when the GO_MUSIC_COVERS command is sent. This command causes the display to switch to the Music Covers view, and generates a UI_STATE event message. No HIGHLIGHTED_SELECTION event message is generated because the same album remains highlighted.

### GO_MUSIC_COLLECTIONS

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GO_MUSIC_COLLECTIONS:` |
| Response | *`status`*`:` |

Displays the Music Collections view. If sent during movie playback, playback is halted and the current playback location is saved in the Paused list. While in the Music Collections view, the FILTER_LIST command can be used to search the selected collection, and the KEYBOARD_CHARACTER command can be used to jump to specific letters in the alphabet.

Example

Controller sends:

```
01/9/GO_MUSIC_COLLECTIONS:
```

Kaleidescape System sends:

```
01/9/000:/97
01/!/000:UI_STATE:11:00:00:0:/39
01/!/000:HIGHLIGHTED_SELECTION::/63
```

In this example, the onscreen display is showing the Music List view when the `GO_MUSIC_COLLECTIONS` command is sent. This command causes the display to switch to the Music Collections view, and generates a `UI_STATE` event message. A `HIGHLIGHTED_SELECTION` message is generated because a new album is now highlighted.

### GO_MUSIC_COLLECTION

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GO_MUSIC_COLLECTION:`*`collection_name`*`:` |
| Response | *`status`*`:` |

Displays the Music Collections view and selects the collection indicated.

| | |
|---|---|
| *`collection_name`* | is the name of the predefined or user-defined collection. |

If *`collection_name`* does not exactly match one of the music collections in the system, this command acts like the `GO_MUSIC_COLLECTIONS` command.

If sent during movie playback, playback is halted and the current playback location is saved in the Paused list.

Example

Controller sends:

```
01/1/GO_MUSIC_COLLECTION:Artists:
```

Kaleidescape System sends:

```
01/1/000:/89
01/!/000:HIGHLIGHTED_SELECTION::/63
```

In this example, the `GO_MUSIC_COLLECTION:Artists:` command was sent while the onscreen display was on another collection. The command causes the currently active collection to change to the Artists collection. This change causes the highlighted selection to switch from an album to an artist, which has no content details, so a blank `HIGHLIGHTED_SELECTION` event message is generated.

### GO_NOW_PLAYING

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GO_NOW_PLAYING:` |
| Response | *`status`*`:` |

Displays the Now Playing view. If sent during movie playback, playback is halted and the current playback location is saved in the Paused list. The Now Playing view is automatically displayed when music starts playing.

Example

Controller sends:

```
01/2/GO_NOW_PLAYING:
```

Kaleidescape System sends:

```
01/2/000:/90
01/!/000:UI_STATE:12:00:00:0:/40
```

In this example, the `GO_NOW_PLAYING` command is sent while the onscreen display is on the System Status view. The command causes the display to switch to the Now Playing view and generates a `UI_STATE` event message indicating the change.

## Store views

### GO_MOVIE_STORE

| | |
|---|---|
| Affects | Any Encore movie zone |
| Command | `GO_MOVIE_STORE:` |
| Response | *`status`*`:` |

Displays the Movie Store view.

If this command is sent during movie playback, playback is halted and the current playback location is saved in the Paused list.

Example

Controller sends:

```
01/1/GO_MOVIE_STORE:
```

Kaleidescape System sends:

```
01/1/000:/89
01/!/000:UI_STATE:16:00:00:0:/44
```

In this example, the `GO_MOVIE_STORE` command was sent resulting in a change from the current view to the Store view. The change in `UI_STATE` is also returned showing the player returned with a screen state of 16 which is the Movie Store screen.

## Other views

### GO_SYSTEM_STATUS

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GO_SYSTEM_STATUS:` |
| Response | *`status`*`:` |

Displays the System Status view. If sent during movie playback, playback is halted and the current playback location is saved in the Paused list. The `STATUS_AND_SETTINGS` command also displays the System Status view when no movie is playing.

Example

Controller sends:

```
01/6/GO_SYSTEM_STATUS:
```

Kaleidescape System sends:

```
01/6/000:/94
01/!/000:UI_STATE:08:00:00:0:/45
```

In this example, the onscreen display is on the Movie Collections view when the `GO_SYSTEM_STATUS` command is sent. This command causes the System Status view to be displayed, and a `UI_STATE` event message to be generated confirming the change in view.

### GO_PARENTAL_CONTROL

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GO_PARENTAL_CONTROL:` |
| Response | *status*: |

Displays the Parental Control view. If sent during movie playback, playback is halted and the current playback location is saved in the Paused list.

The user can also change to a parental control level that has a passcode at any time from the Kaleidescape onscreen display by entering the passcode for that level. Sending a digit character with the `KEYBOARD_CHARACTER` command when the onscreen display is active causes the passcode dialog box to appear.

Example

Controller sends:

```
01/3/GO_PARENTAL_CONTROL:
```

Kaleidescape System sends:

```
01/3/000:/91
01/!/000:UI_STATE:04:00:00:0:/41
```

In this example, the `GO_PARENTAL_CONTROL` command is sent, and the view changes to the Parental Control view. This causes a `UI_STATE` event message to be generated confirming the change.

### GO_VAULT_SUMMARY

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GO_VAULT_SUMMARY:` |
| Response | *status*: |

Displays the Vault Summary view. If no vault is connected to a player in the system, the onscreen display displays text indicating there are no vaults available. If sent during playback, playback is halted and the current playback location is saved in the Paused list.

Example

Controller sends:

```
02/1/GO_VAULT_SUMMARY:
```

Kaleidescape System sends:

```
02/1/000:/90
02/!/000:UI_STATE:14:00:00:0:/43
02/!/000:HIGHLIGHTED_SELECTION::/64
```

## User input

### GET_USER_INPUT

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GET_USER_INPUT:` |
| Response/Event | *`status`*`:USER_INPUT:`*`type`*`:`*`prompt`*`:`*`entry`*`:` |

This message provides information about the user input requested by the user interface. The message includes the type of input request, as well as prompts that can be displayed on a controller.

The event message is generated when the need for input changes, or when the prompt or input changes.

*`type`* specifies the type of prompt.

| | |
|---|---|
| `00` | No prompt |
| `01` | Alphanumeric prompt (full keyboard required) |
| `02` | Numeric prompt (numeric keypad required). |

*`prompt`* specifies the query, such as `Search for title`.

*`entry`* contains the text entered by the user.

To add characters to the entry, the controller sends `KEYBOARD_CHARACTER` commands. To remove characters from the input prompt, use the `BACKSPACE` command. Every change to the input generates another `USER_INPUT` event message.

Example

Controller sends:

```
01/1/GET_USER_INPUT:
```

Kaleidescape System sends:

```
01/1/000:USER_INPUT:01:Search for title::/67
```

In this example, the controller sends the `GET_USER_INPUT` command shortly after connecting, to the resynchronize state. The response indicates that the user interface is requesting alphanumeric text in response to the prompt `Search for title`.

### KEYBOARD_CHARACTER

| | |
|---|---|
| Affects | Any movie zone |
| Request | `KEYBOARD_CHARACTER:`*`character`*`:` |
| Response | *`status`*`:` |

Sends a single character to the onscreen display.

*`character`* is the character to be sent to the onscreen display. The character can be a letter, digit, or any other symbol.

If a letter is sent to the onscreen display while the Movie List, Music List, Movie Collections, or Music Collections view is active with no prior `FILTER_LIST` command, this command causes the list to jump to the first entry that starts with the character sent. (This is known as quick search.)

Note that a colon character must be escaped with a backslash:

```
01/1/KEYBOARD_CHARACTER:\::          (correct format)
01/1/000:/89
01/2/KEYBOARD_CHARACTER:::           (incorrect format)
01/2/012:/93
```

Example 1

Controller sends:

```
01/4/KEYBOARD_CHARACTER:r:
```

Kaleidescape System sends:

```
01/4/000:/92
02/!/000:HIGHLIGHTED_SELECTION:1.W_22033:/91
```

Controller sends:

```
01/5/KEYBOARD_CHARACTER:o:
```

Kaleidescape System sends:

```
01/5/000:/93
02/!/000:HIGHLIGHTED_SELECTION:1.W_21892:/03
```

In this example, the OSD was displaying the music list. The user sent the character `r` and the OSD scrolled to the first selection that started with `r`, then sent the `HIGHLIGHTED_SELECTION` event. The user then sent the character `o` and the OSD scrolled to the first selection starting with `o` and sent the `HIGHLIGHTED_SELECTION` event for that command. Note that the OSD did not filter for a title that contains, `ro`, but instead performed two quick searches based on individual characters.

Example 2

See example for `FILTER_LIST` command.

> Note: This command replaces the `DIGIT` command. Both can be used to send letters and digits. `KEYBOARD_CHARACTER` is preferred because the name is more accurate.

### KEYBOARD_LITERAL

| | |
|---|---|
| Affects | Any movie zone |
| Request | `KEYBOARD_LITERAL:`*`character`*`:` |
| Response | *`status`*`:` |

Sends a single character to the onscreen keyboard.

*`character`* is the ASCII character >=32 to be sent to the onscreen display. The character can be a letter, digit, or any other symbol.

In most cases works this command similar to `KEYBOARD_CHARACTER`. If a letter is sent to the onscreen display while the Movie List, Music List, Movie Collections, or Music Collections view is active with no prior `FILTER_LIST` command, this command causes the list to jump to the first entry that starts with the character sent. (This is known as quick search.)

For onscreen keyboard entries, it is recommended to use KEYBOARD_LITERAL as it allows upper and lowercase letters to be sent.

Note that a colon or slash characters must be escaped with a backslash:

```
01/1/KEYBOARD_LITERAL:\::          (correct format)
01/1/000:/89
01/2/KEYBOARD_LITERAL:::           (incorrect format)
01/2/012:/93
```

Example 1

Controller sends:

```
01/4/KEYBOARD_LITERAL:r:
```

Kaleidescape System sends:

```
01/4/000:/92
02/!/000:HIGHLIGHTED_SELECTION:1.W_22033:/91
```

Controller sends:

```
01/5/KEYBOARD_LITERAL:o:
```

Kaleidescape System sends:

```
01/5/000:/93
02/!/000:HIGHLIGHTED_SELECTION:1.W_21892:/03
```

In this example, the OSD was displaying the movie list. The user sent the character `r` and the OSD scrolled to the first selection that started with `r`, then sent the `HIGHLIGHTED_SELECTION` event. The user then sent the character `o` and the OSD scrolled to the first selection starting with `o` and sent the `HIGHLIGHTED_SELECTION` event for that command. Note that the OSD did not filter for a title that contains, `ro`, but instead performed two quick searches based on individual characters.

Example 2

With an onscreen keyboard open for Player Name:

Controller sends:

```
01/1/KEYBOARD_LITERAL:r:
```

Kaleidescape System sends:

```
01/1/000:/89
01/!/000:USER_INPUT:01:Name:r:/15
```

Controller sends:

```
01/1/KEYBOARD_LITERAL:O:
```

Kaleidescape System sends:

```
01/1/000:/89
01/!/000:USER_INPUT:01:Name:rO:/94
```

In this example, the OSD was the Enter a Player Name from the settings menu. The user sent the character `r` and the `r` was entered into text box and the system responded with the first `USER_INPUT` status message.  Next a capital `O` was sent, resulting in a capital `O` appearing next to `r` in the onscreen text box and the `USER_INPUT` status was updated to `rO`.

### BACKSPACE

| | |
|---|---|
| Affects | Any movie zone |
| Command | `BACKSPACE:` |
| Response | *`status`*`:` |

During any kind of data entry, such as a `FILTER_LIST` or passcode entry, this command erases the last character entered.

Example

See example for `FILTER_LIST` command.

### GET_USER_INPUT_PROMPT

| | |
|---|---|
| Affects | Any Encore movie zone |
| Command | `GET_USER_INPUT_PROMPT:` |
| Response | *`status`*`:USER_INPUT_PROMPT:`*`type:icon:prompt:displayed:char_limit:valid`* |

Returns user input prompt info and properties currently shown on screen.

| | | |
|---|---|---|
| *type* | specifies the type of prompt. | |
| | `00` | No prompt |
| | `01` | Alphanumeric prompt (full keyboard required) |
| | `02` | Numeric prompt (numeric keypad required). |
| *icon* | `##` | Current icon displayed in the prompt |
| *prompt* | specifies the query, such as `Search for title`. | |
| *displayed* | `0` | Entry not displayed |
| | `1` | Entry is displayed |
| *char_limit* | Character limit for input | |
| *valid* | `0` | Current entry is not valid |
| | `1` | Current entry is valid |

Example

Controller sends:

```
01/1/GET_USER_INPUT_PROMPT:
```

Kaleidescape System sends:

```
01/1/000:USER_INPUT_PROMPT:00:00::00:0:1:/71
```

### SET_USER_INPUT_ENTRY

| | |
|---|---|
| Affects | Any Encore movie zone |
| Command | `SET_USER_INPUT_ENTRY:`*`string`*`:` |
| Response | *`status`*`:` |

Sends the entry in *`string`* to the current on screen user entry prompt.

Example

Controller sends:

```
01/1/SET_USER_INPUT_ENTY:mypassword:
```

Kaleidescape System sends:

```
01/1/000:/89
```

## View-specific commands

### FILTER_LIST

| | |
|---|---|
| Affects | Any movie zone |
| Command | `FILTER_LIST:` |
| Response | *`status`*`:` |

This command is used to search the list or collections views according to a user-entered string. This command causes a message to appear in the upper right corner, indicating that the column currently selected is being searched. The controller can then send `KEYBOARD_CHARACTER` and `BACKSPACE` commands to build a string used by the onscreen display to filter the list.

A `CANCEL` command halts filtering.

There is no need to implement a keyboard with extra characters, for example, accents and other similar characters. Filtering is accomplished using just the basic, unaccented characters.

Example

Controller sends:

```
01/7/GO_MOVIE_LIST:
```

Kaleidescape System sends:

```
01/7/000:/95
01/!/000:UI_STATE:01:00:00:0:/38
```

Controller sends:

```
01/8/FILTER_LIST:
```

Kaleidescape System sends:

```
01/8/000:/96
01/!/000:USER_INPUT:01:Search for title::/51
```

Controller sends:

```
01/9/KEYBOARD_CHARACTER:S:
```

Kaleidescape System sends:

```
01/9/000:/97
01/!/000:USER_INPUT:01:Search for title:S:/34
```

Controller sends:

```
01/0/KEYBOARD_CHARACTER:E:
```

Kaleidescape System sends:

```
01/0/000:/88
01/!/000:USER_INPUT:01:Search for title:Se:/35
01/!/000:HIGHLIGHTED_SELECTION:1.0-u_67a47:/76
```

Controller sends:

```
01/3/BACKSPACE:
```

Kaleidescape System sends:

```
01/3/000:/91
01/!/000:USER_INPUT:01:Search for title:S:/34
```

In this example, the controller selects the Movie List view, then sends the FILTER_LIST command. The user sends the letters S and E, and a selection starting with Se is highlighted. This is shown by the HIGHLIGHTED_SELECTION event message. Subsequently, the user decides to remove a character and sends the BACKSPACE command, causing the E to be removed.

### SHUFFLE_COVER_ART

| | |
|---|---|
| Affects | Any movie zone |
| Command | `SHUFFLE_COVER_ART:` |
| Response | *`status`*`:` |

This command shuffles the cover art on the Movie Covers and Music Covers views. Movies or albums similar to the item currently selected are rearranged to surround the selected cover.

If automatic cover shuffle is off, this command is the only way to activate the shuffling. Additionally, if the cover art has been alphabetized by the `ALPHABETIZE_COVER_ART` command, this command removes the alphabetization.

Example

Controller sends:

```
01/6/SHUFFLE_COVER_ART:
```

Kaleidescape System sends:

```
01/6/000:/94
```

In this example, the `SHUFFLE_COVER_ART` command is sent while on the Movie Covers view. Because the selected item does not change, the controller receives no further response.

### CHILD_SHUFFLE_COVER_ART

| | |
|---|---|
| Affects | Any movie zone |
| Command | `CHILD_SHUFFLE_COVER_ART:` |
| Response | *`status`*`:` |

If the player is displaying the child user interface, the cover art is shuffled. If the child user interface is not active, activates the child user interface and emits appropriate events to indicate the change. This command has no effect during movie playback.

Example

Controller sends:

```
01/1/CHILD_SHUFFLE_COVER_ART:
```

Kaleidescape System sends:

```
01/1/000:/89
02/!/000:CHILD_MODE_STATE:1:/63
02/!/000:USER_DEFINED_EVENT:SELECT_KALEIDESCAPE_INPUT:/77
```

The player was not displaying the child user interface, but after receiving the `CHILD_SHUFFLE_COVER_ART` command, switched to the child user interface and emitted the appropriate events.

### ALPHABETIZE_COVER_ART

| | |
|---|---|
| Affects | Any movie zone |
| Command | `ALPHABETIZE_COVER_ART:` |
| Response | *`status`*`:` |

Arranges covers alphabetically by title in the Movie Covers or Music Covers view around a highlighted cover. A subsequent `SHUFFLE_COVER_ART` command removes the alphabetization.

Example

Controller sends:

```
01/7/ALPHABETIZE_COVER_ART:
```

Kaleidescape System sends:

```
01/7/000:/95
```

In this example, the `ALPHABETIZE_COVER_ART` command is sent while on the Movie Covers view. No other messages occur because the selected item does not change.

### DEFAULT_LEVEL

| | |
|---|---|
| Affects | Any movie zone |
| Command | `DEFAULT_LEVEL:` |
| Response | *`status`*`:` |

Directly sets the parental control level to the default level set in the browser interface. This command provides the same function as accessing the Parental Control view and selecting the appropriate parental control level. See also `SAFE_LEVEL`.

Example

Controller sends:

```
01/4/DEFAULT_LEVEL:
```

Kaleidescape System sends:

```
01/4/000:/92
```

### SAFE_LEVEL

| | |
|---|---|
| Affects | Any movie zone |
| Command | `SAFE_LEVEL:` |
| Response | *`status`*`:` |

Directly sets the parental control level to the highest level that has no passcode as set in the browser interface. This command provides the same function as accessing the Parental Control view and selecting the appropriate parental control level. See also `DEFAULT_LEVEL`.

Example

Controller sends:

```
01/4/SAFE_LEVEL:
```

Kaleidescape System sends:

```
01/4/000:/92
```

## Page and content details

DETAILS

| | |
|---|---|
| Affects | Any movie zone |
| Command | `DETAILS:` |
| Response | *`status`*`:` |

Toggles the display of the details page over the current display. The details page shows detailed information for the highlighted selection (movie, album, etc.). A subsequent `CANCEL` command hides the details page, the same as if the user chooses the Return option on the details page menu.

Example

Controller sends:

```
01/2/DETAILS:
```

Kaleidescape System sends:

```
01/2/000:/90
01/!/000:UI_STATE:01:00:00:0:/38
```

Controller sends:

```
01/3/DETAILS:
```

Kaleidescape System sends:

```
01/3/000:/91
01/!/000:UI_STATE:01:01:00:0:/39
```

In this example, the onscreen display is on the Movie List view showing a details page for a particular movie. The `DETAILS` command is sent, causing the details page to disappear, and a `UI_STATE` event message to be generated.

The second `DETAILS` command subsequently restores the details page and produces another `UI_STATE` event message that indicates that the details page is displayed.

### DISC_IN_TRAY_TOGGLE

| | |
|---|---|
| Affects | Any movie player with an optical disc drive |
| Command | `DISC_IN_TRAY_TOGGLE:` |
| Response | *`status`*`:` |

Toggles the display of the disc in player page over the current display. The disc in player page shows detailed information about the disc currently in the optical drive of the player

Example

Controller sends:

```
01/1/DISC_IN_TRAY_TOGGLE:
```

Kaleidescape System sends:

```
01/1/000:/89

01/!/000:UI_STATE:03:01:00:0:/42
01/!/000:HIGHLIGHTED_SELECTION:1.283ec4aa94a9267d-L10005_263d:/01
```

In this example, the onscreen display is on the Movie Covers view. `DISC_IN_TRAY_TOGGLE` is sent causing the disk in player popup to appear, as indicated by the `UI_STATE` event. A `HIGHLIGHTED_SELECTION` message is also generated with the handle of the movie or album in the optical drive. The controller could now send `GET_CONTENT_DETAILS` to retrieve information about the movie or album.

### GET_CONTENT_DETAILS

| | |
|---|---|
| Affects | All components |
| Command | `GET_CONTENT_DETAILS:`*`handle`*`:`*`passcode`*`:` |
| Response | *`status`*`:CONTENT_DETAILS_OVERVIEW:`<br>*`num_lines`*`:`*`handle`*`:`*`table`*`:` |
| Response | *`status`*`:CONTENT_DETAILS:`*`line`*`:`*`name`*`:`*`value`*`:` |

This command provides concise information about a movie or album.

Command parameters

*`handle`* is the identifier for the movie or album for which content details are being requested. This handle can come from a `HIGHLIGHTED_SELECTION` message, a `BROWSE_RESPONSE` message, or a `MUSIC_TITLE` message.

*`passcode`* is used to access content not available on the current parental control level. When blank, only content for the current parental control level can be displayed. Most controllers do not have to use this field.

Response

The response to this command always includes a `CONTENT_DETAILS_OVERVIEW` message. If the *`num_lines`* field of the message is not `0`, several `CONTENT_DETAILS` messages follow containing actual information for the content.

**CONTENT_DETAILS_OVERVIEW** response

*num_lines* is the number of CONTENT_DETAILS messages following the overview.

*handle* is the identifier for the movie or album that was requested in the GET_CONTENT_DETAILS command.

*table* is the type of content details information being provided, either movies or albums.

**CONTENT_DETAILS** response

Each CONTENT_DETAILS response contains a single name/value pair providing information about the piece of content.

*line* identifies the index of this particular CONTENT_DETAILS response among the total number of CONTENT_DETAILS responses being sent. This will count from 1 up to *num_lines*.

*name* is the name of the value being provided by this CONTENT_DETAILS response. For example, Title.

*value* is the actual information associated with the name for this response. For example, Toy Story. If the response contains multiple values, values are separated by a carriage return (\r ASCII 13).

Depending on the table of content details, different name/value pairs are returned. Many common values for content are listed below. A controller must listen for all pertinent values when parsing CONTENT_DETAILS messages.

For content in the movies table

`Content_handle` reiterates the handle for this piece of content, as requested in the GET_CONTENT_DETAILS command.

`Title` is the title of the movie.

`Cover_URL` is a URL for the cover art image, for display on touch panels.

`HiRes_cover_URL` is a URL for a 1080p resolution cover art image, for display on touch panels.

`Rating` is the parental control rating for the movie, for example, G, PG, PG-13, R.

`Rating_reason` is a text string that is reason for the rating, for example, `sexual material and language`.

`Year` is the year that the movie was released.

`Running_time` is the running time of the movie in minutes.

`Actors` is a carriage return (\r) delimited list of actors and actresses that star in the movie.

`Director` is the name of the director of the movie.

`Directors` is a carriage return (\r) delimited list of directors that worked on the movie.

`Genre` is the genre for the movie.

`Genres` is a carriage return (`\r`) delimited list of genres associated with the movie.

`Synopsis` is the synopsis of the movie.

`Color_description` indicates whether the movie is in color or black and white.

`Country` is the country which produced the movie.

`Aspect_ratio` is the aspect ratio of the movie, for example, 2.40.

`Disc_location` provides the location of the disc, for example,

```
in the tray
in the disc vault "Disc Vault - 0600
00001234."
```

Example

Kaleidescape System sends:

```
01/!/000:HIGHLIGHTED_SELECTION:1.0-S_ca4fb:/77
```

Controller sends:

```
01/1/GET_CONTENT_DETAILS:1.0-S_ca4fb::
```

Kaleidescape System sends:

```
01/1/000:CONTENT_DETAILS_OVERVIEW:16:1.0-S_ca4fb:movies:/74
01/1/000:CONTENT_DETAILS:1:Content_handle:1.0-S_ca4fb:/82
01/1/000:CONTENT_DETAILS:2:Title:The Incredibles:/82
01/1/000:CONTENT_DETAILS:3:Cover_URL:http\:\/\/10.100.12.194\/panelcover
art\/b9bca9a6f224fb54\/3866055.jpg:/53
01/1/000:CONTENT_DETAILS:4:HiRes_cover_URL:http\:\/\/10.100.12.194\/pane
lcoverart\/b9bca9a6f224fb54\/4254312.jpg:/44
01/1/000:CONTENT_DETAILS:5:Rating:PG:/89
01/1/000:CONTENT_DETAILS:6:Year:2004:/25
01/1/000:CONTENT_DETAILS:7:Running_time:115:/41
01/1/000:CONTENT_DETAILS:8:Actors:Craig T. Nelson\rHolly Hunter\rJason
Lee\rSamuel L. Jackson\rBrad Bird\rSarah Vowell\rSpencer Fox\rWallace
Shawn\rElizabeth Pe\d241a:/34
01/1/000:CONTENT_DETAILS:9:Directors:Brad Bird:/66
01/1/000:CONTENT_DETAILS:10:Genres:Animated\rAction\rComedy\rFamily:/88
01/1/000:CONTENT_DETAILS:11:Rating_reason:action violence:/49
01/1/000:CONTENT_DETAILS:12:Synopsis:A middle-aged hero living in the
suburbs with his super-powered family dusts off his tights to confront a
mysterious threat.:/22
01/1/000:CONTENT_DETAILS:13:Color_description:Color:/77
01/1/000:CONTENT_DETAILS:14:Country:USA:/62
01/1/000:CONTENT_DETAILS:15:Aspect_ratio:2.40:/16
01/1/000:CONTENT_DETAILS:16:Disc_location::/14
```

For content in the albums table

`Album_content_handle` is the handle for the album, as requested by the `GET_CONTENT_DETAILS` command.

`Album_title` is the title of the album.

`Artist` is the artist listed for the album.

`Artists` is a carriage return (\r) delimited list of artists associated with the album.

`Composer` is the composer for the album.

`Composers` is a carriage return (\r) delimited list of composers that are associated with the album.

`Performer` is the performer or artist associated with the album.

`Performers` is a carriage return (\r) delimited list of performers or artists associated with the album.

`Cover_URL` is a URL for the cover art image, for display on touch panels.

`HiRes_cover_URL` is a URL for a 1080p resolution cover art image, for display on touch panels.

`Year` is the year the album was released.

`Running_time` is the length of the album in seconds.

`Last_played` future use

`Last_played_relative` future use

`Genre` is the genre of music associated with the album.

`Genres` is a carriage return (\r) delimited list containing the genres associated with the album.

`Review` is the first 255 characters of the album review provided by the onscreen display. (Note that this is generally not the full review.)

`Disc_location` is a string indicating where the disc is currently located, for example, `in the tray`.

Example

Kaleidescape System sends:

```
01/!/000:HIGHLIGHTED_SELECTION:1.R_18760:/97
```

Controller sends:

```
01/1/GET_CONTENT_DETAILS:1.R_18760::
```

Kaleidescape System sends:

```
01/1/000:CONTENT_DETAILS_OVERVIEW:12:1.R_18760:albums:/75
01/1/000:CONTENT_DETAILS:1:Album_content_handle:1.R_18760:/26
01/1/000:CONTENT_DETAILS:2:Album_title:Greetings from Asbury Park, N.J.:/77
01/1/000:CONTENT_DETAILS:3:Artist:Bruce Springsteen:/54
01/1/000:CONTENT_DETAILS:4:Cover_URL:http\:\/\/10.100.12.194\/panelcoverart\/b9bca9a6f224fb54\/_music_\/f35592zh2ls.jpg:/24
```

```
01/1/000:CONTENT_DETAILS:5:HiRes_cover_URL:http\:\/\/10.100.12.194\/pane
lcoverart\/b9bca9a6f224fb54\/_music_\/ff2237sr35q.jpg:/16
01/1/000:CONTENT_DETAILS:6:Year:1973:/39
01/1/000:CONTENT_DETAILS:7:Running_time:2236:/95
01/1/000:CONTENT_DETAILS:8:Last_played:0000-00-00 00\:00\:00:/60
01/1/000:CONTENT_DETAILS:9:Last_played_relative::/22
01/1/000:CONTENT_DETAILS:10:Genres:Pop\/Rock\rRock & Roll\rContemporary
Pop\/Rock\rAlbum Rock:/95
01/1/000:CONTENT_DETAILS:11:Review:Bruce Springsteen's debut album found
him squarely in the tradition of Bob Dylan\: folk-based tunes arranged
for an electric band featuring piano and organ (plus, in Springsteen's
case, 1950s-style:/32
01/1/000:CONTENT_DETAILS:12:Disc_location::/10
```

## GET_HIGHLIGHTED_SELECTION

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GET_HIGHLIGHTED_SELECTION:` |
| Response/Event | *status*:HIGHLIGHTED_SELECTION:*handle*: |

This message provides the handle for the currently highlighted item in the onscreen display. This message can be used with the `GET_CONTENT_DETAILS` command to get content details for the item. If nothing is highlighted, or if the item does not have content details, `HIGHLIGHTED_SELECTION` returns a blank handle.

The event message is generated when the item currently highlighted changes, for example, when the user highlights various movies and albums.

| | |
|---|---|
| *handle* | is an identifier for the currently highlighted item that can be used in other commands, for example, `GET_CONTENT_DETAILS`. |

Example

Kaleidescape System sends:

```
01/!/000:HIGHLIGHTED_SELECTION:1.R_18760:/97
```

Controller sends:

```
01/1/GET_CONTENT_DETAILS:1.R_18760::
```

In this example, the user is browsing through the music list and rests on a particular album. The controller then uses the `HIGHLIGHTED_SELECTION` to request content details for that album.

## Screen saver commands

### GO_SCREEN_SAVER

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GO_SCREEN_SAVER:` |
| Response | *`status`*`:` |

Displays the screen saver (has no effect if the screen saver is already active). Any active playback continues without interruption, the video is merely being obscured by the screen saver.

For the user’s convenience, the screen saver is removed with the next user input, ignoring the command itself, and returning a success response. (Commands unrelated to user interface or playback, such as `GET_TIME`, are generally interpreted without removing the screen saver.)

An explicit `STOP_SCREEN_SAVER` command also removes the screen saver.

A `UI_STATE` event message is sent any time the screen saver is activated, whether automatically or by a command.

Example

Controller sends:

```
01/8/GO_SCREEN_SAVER:
```

Kaleidescape System sends:

```
01/8/000:/96
01/!/000:UI_STATE:03:00:00:1:/41
```

This example shows the `GO_SCREEN_SAVER` command sent to the Kaleidescape System. The system generates a `UI_STATE` event message indicating that the screen saver is now active.

### STOP_SCREEN_SAVER

| | |
|---|---|
| Affects | Any movie zone |
| Command | `STOP_SCREEN_SAVER:` |
| Response | *`status`*`:` |

Removes the screen saver, returning the display to whatever the screen saver was covering. This command has no effect if the screen saver is not active.

For the user’s convenience, the screen saver is removed with the next user input, ignoring the command itself and returning a success response. (Commands unrelated to user interface or playback, such as `GET_TIME`, are generally interpreted without removing the screen saver.)

A `UI_STATE` event message is sent out any time the screen saver is removed, whether by a command or a user action.

The onscreen display activates the screen saver with a `GO_SCREEN_SAVER` command, or automatically after a certain time passes with no user input (the time period can be set in the browser interface).

Example

Controller sends:

```
01/9/STOP_SCREEN_SAVER:
```

Kaleidescape System sends:

```
01/9/000:/97
01/!/000:UI_STATE:03:00:00:0:/40
```

This example shows the `STOP_SCREEN_SAVER` command sent to remove the screen saver. The onscreen display returns to the Movie Covers view and generates a `UI_STATE` message indicating the change.

# OSD Playback Control Commands

> Note: The response code for music related commands will return “Command is not available” for products which do not support music.

These commands control movie and music playback, and the user interface. Commands are grouped and a detailed description of each command follows. Table 9 lists playback control commands.

Table 9: Playback control command summary

| Command | Description |
|---|---|
| Playback control | |
| `PLAY` | Begins playback of movies and music. |
| `PAUSE` | Toggles pause. |
| `STOP` | Stops playback. |
| `NEXT` and `PREVIOUS` | Skips forward or backward through chapters or songs. |
| `SCAN_FORWARD` and `SCAN_REVERSE` | Cycles through fast-forward or fast-reverse. |
| `REPLAY` | Skips back five seconds during movie playback. |

| Command | Description |
|---|---|
| Child Remote playback control | |
| CHILD_PLAY | If child user interface is active, or if selected movie is in Child collection, plays the selected movie. Otherwise, activates child user interface. |
| CHILD_PAUSE | Toggles pause and displays child user interface. |
| CHILD_STOP | Stops playback. |
| Playback information | |
| SET_STATUS_CUE_PERIOD | Sets the frequency of PLAY_STATUS and MUSIC_PLAY_STATUS event messages. |
| GET_PLAY_STATUS | Identifies movie play mode, speed, location and chapter. |
| GET_PLAYING_TITLE_NAME | Provides the title of the movie currently playing. |
| GET_MUSIC_NOW_PLAYING_STATUS | Sends the state of the Now Playing list. |
| GET_MUSIC_PLAY_STATUS | Provides playback information for the currently playing music. |
| GET_MUSIC_TITLE | Provides information about the song currently playing. |
| Music playback controls | |
| MUSIC_RANDOM_ON | Turns on random playback for music. |
| MUSIC_RANDOM_OFF | Turns off random playback for music. |
| MUSIC_RANDOM_TOGGLE | Toggles random music playback on and off. |
| MUSIC_REPEAT_ON | Turns on repeat playback for music. |
| MUSIC_REPEAT_OFF | Turns off repeat playback for music. |
| MUSIC_RANDOM_TOGGLE | Toggles repeat music playback on and off. |
| GET_CONTROLLED_ZONE | Returns the music zone currently under control. |
| SET_CONTROLLED_ZONE | Changes the music zone controlled by the onscreen display. |

| Command | Description |
|---|---|
| DVD/Blu-ray Disc navigation | |
| DISC_MENU | Displays DVD or Blu-ray Disc menu for the current playback. |
| DISC_TOP_MENU | Displays the top menu for the DVD or Blu-ray Disc. |
| DISC_RESUME | Resumes playback from the point of interruption. |
| START_CHAPTER_ENTRY | Displays a tab to enter chapter number to skip directly to. |
| START_DISC_TITLE_ENTRY | Displays a tab to enter title number to skip directly to a title. |
| Movie playback options | |
| SHOW_NAVIGATION_OVERLAY | During playback, opens the movie overlay to the chapter/title navigation option. |
| STATUS_AND_SETTINGS | During playback, toggles the display of the movie overlay. Otherwise, brings up the System Status page. |
| INTERMISSION_ON | Pauses playback and displays intermission screen. |
| INTERMISSION_OFF | Removes the intermission screen and resumes playback. |
| INTERMISSION_TOGGLE | Toggles intermission screen on and off. |
| SET_FAVORITE_SCENE_START | Records a bookmark for the start of a scene. |
| SET_FAVORITE_SCENE_END | Records a bookmark for the end of a scene. |
| START_SEND_NUMBER_TO_DISC_ENTRY | Displays a tab to enter a number key to send to a DVD or Blu-ray Disc. |
| ANGLE_NEXT | Changes to the next camera angle defined for playback. |
| ANGLE_PREVIOUS | Changes to the previous camera angle defined for playback. |
| AUDIO_NEXT | Changes to the next audio stream during movie playback. |
| SUBTITLES_NEXT | Changes to the next subtitle track during playback. |

| Command | Description |
|---|---|
| GET_CAMERA_ANGLE | Provides information about the current camera angle. |
| Blu-ray Disc playback options | |
| Blu-ray color buttons | Performs actions associated with color buttons. |
| GET_MOVIE_MEDIA_TYPE | Identifies the type of media being played. |
| BLURAY_SPECIAL_STOP | Stops Blu-ray Disc playback. Use with caution. |
| BLURAY_POPUP_MENU_TOGGLE | Toggles display of Blu-ray Disc pop-up menu. |

## Playback control

### PLAY

| | |
|---|---|
| Affects | Any zone |
| Command | `PLAY:` |
| Response | *`status`*`:` |

Begins playback of movie or music. When sent to a music zone, it resumes (if paused) or restarts music playback if stopped or finished playing.

If sent to a movie zone, begins playing the highlighted movie, album, track, or other playable item (scene, script, etc.). Movies previously stopped via a `STOP` command resume playback from the stopping point. Other movies begin playback at the start of the feature if bookmarked in the Kaleidescape Movie Guide. If a movie has no feature bookmark, playback begins like a normal DVD or Blu-ray Disc.

During movie playback, this command begins playing the selected item in a DVD or Blu-ray Disc menu, returns playback to regular forward speed, resumes paused playback or turns intermission off. Compare with `PAUSE_OFF` and `INTERMISSION_OFF`.

During music playback, if the music is paused, `PLAY` always resumes. In the Now Playing view, `PLAY` also restarts the music if music has been stopped or has finished playing.

Example 1

Controller sends:

```
01/5/PLAY:
```

Kaleidescape System sends:

```
01/5/000:/93
03/!/000:UI_STATE:07:00:00:0:/46
03/!/000:TITLE_NAME:Serenity:/12
```

```
03/!/000:MOVIE_MEDIA_TYPE:01:/36
03/!/000:VIDEO_MODE:02:04:04:/68
```

In this example, the onscreen display has the movie Serenity highlighted. The PLAY command is sent, and playback begins. The start of playback generates several event messages about the playback.

Example 2

Controller sends:

```
01.01/1/PLAY:
```

Kaleidescape System sends:

```
01.01/1/000:/32
01.01/!/000:MUSIC_NOW_PLAYING_STATUS:00011:00000:1:0:0000000010::/12
01.01/!/000:PLAYING_MUSIC_INFORMATION:R_4026:Eric Clapton - Time
Pieces\: Best of Eric Clapton:/94
01.01/!/000:MUSIC_TITLE:I Shot the Sheriff:Eric Clapton:Time
Pieces\:Best of Eric Clapton:1.b9bca9a6f224fb54-
t301_21:1.R_4026:2.20000:/00
01.01/!/000:MUSIC_PLAY_STATUS:2:0:00263:+00000:000.00:/51
```

In this example, the user has previously stopped playback of the Best of Eric Clapton album. The PLAY command is sent to the music zone, and playback of the album restarts. Playback restart generates several event messages about the music playback.

PAUSE

| | |
|---|---|
| Affects | Any zone |
| Command | PAUSE:<br>PAUSE_ON:<br>PAUSE_OFF: |
| Response | *status*: |

During movie or music playback, PAUSE_ON pauses playback, PAUSE_OFF resumes playback, and PAUSE toggles between pausing and resuming playback of the movie or music.

Example 1

Controller sends:

```
01/8/PAUSE:
```

Kaleidescape System sends:

```
01/8/000:/96
03/!/000:PLAY_STATUS:1:0:01:07136:00027:001:00300:00027:/23
```

In this example, the PAUSE command is sent during movie playback, causing movie playback to pause. This command generated a PLAY_STATUS event message that indicates the new paused state.

Example 2

Controller sends:

```
01/2/PAUSE:
```

Kaleidescape System sends:

```
01/2/000:/90
03/!/000:MUSIC_PLAY_STATUS:1:0:00144:+00081:056.25:/34
03/!/000:MUSIC_NOW_PLAYING_STATUS:00012:00000:1:0:0000000033:2.20000:/15
```

In this example, the PAUSE command is sent during music playback. This command caused music playback to pause, generating a MUSIC_PLAY_STATUS event message and a MUSIC_NOW_PLAYING_STATUS event message, with the modified information about the playback.

STOP

| | |
|---|---|
| Affects | Any zone |
| Command | STOP: |
| Response | *status*: |

During movie playback, stops the movie and displays the last screen of the Kaleidescape user interface that was visible when playback last began. The movie is placed in the Paused collection to be resumed later.

During music playback, stops playback.

> Note: Some Blu-ray Discs support a special stop command that does not always display the Kaleidescape user interface. See BLURAY_SPECIAL_STOP for information about these special stop functions.

Example

Controller sends:

```
01/4/STOP:
```

Kaleidescape System sends:

```
01/4/000:/92
03/!/000:MUSIC_NOW_PLAYING_STATUS:00012:00000:1:0:0000000035:2.20000:/17
03/!/000:PLAYING_MUSIC_INFORMATION:::/94
03/!/000:MUSIC_TITLE::::::/47
03/!/000:MUSIC_PLAY_STATUS:0:0:00000:+00000:000.00:/97
03/!/000:MUSIC_NOW_PLAYING_STATUS:00012:00000:1:0:0000000036::/80
```

This example shows the way the system reacts when the STOP command is sent during music playback. The event messages state that no music is now playing.

### NEXT and PREVIOUS

| | |
|---|---|
| Affects | Any zone |
| Command | `NEXT:`<br>`PREVIOUS:` |
| Response | *`status`*`:` |

During movie playback, skips forward or backward to the next chapter boundary. A reverse returns to the beginning of the current chapter, not the beginning of the previous chapter. To return to the beginning of the previous chapter, press a Previous button twice.

During music playback, skips forward or backward to the next track boundary. A reverse returns to the beginning of the current track, not the beginning of the previous track. To return to the beginning of the previous track, press a Previous button twice.

See also the context-sensitive commands that page through user interface lists (`PAGE_UP_OR_PREVIOUS`, etc.).

Example

Controller sends:

```
01/2/PREVIOUS:
```

Kaleidescape System sends:

```
01/2/000:/90
```

Controller sends:

```
01/3/PREVIOUS:
```

Kaleidescape System sends:

```
01/3/000:/91
03/!/000:PLAY_STATUS:2:0:01:07570:00151:002:00270:00000:/22
```

Controller sends:

```
01/4/NEXT:
```

Kaleidescape System sends:

```
01/4/000:/92
03/!/000:PLAY_STATUS:2:0:01:07570:00422:003:00142:00000:/22
```

In this example, the `PREVIOUS` command was sent twice to return to the previous chapter, then a `NEXT` command was sent to skip forward one chapter.

### SCAN_FORWARD and SCAN_REVERSE

| | |
|---|---|
| Affects | Any zone |
| Command | `SCAN_FORWARD:`<br>`SCAN_REVERSE:` |
| Response | *`status`*`:` |

During movie playback, cycles through fast-forward or fast-reverse at 2×, 4×, and 8× the speed. To return to normal speed, issue a PLAY command.

During music playback, cycles through fast-forward or fast-reverse at 4× and 16× the speed. To return to normal speed, issue a PLAY command.

Example

Controller sends:

```
01/5/SCAN_FORWARD:
```

Kaleidescape System sends:

```
01/5/000:/93
03/!/000:PLAY_STATUS:4:1:01:07570:00587:004:00314:00022:/43
```

Controller sends:

```
01/6/SCAN_FORWARD:
```

Kaleidescape System sends:

```
01/6/000:/94
03/!/000:PLAY_STATUS:4:2:01:07570:00594:004:00314:00029:/49
```

Controller sends:

```
01/7/PLAY:
```

Kaleidescape System sends:

```
01/7/000:/95
03/!/000:PLAY_STATUS:2:0:01:07570:00608:004:00314:00042:/36
```

In this example, the SCAN_FORWARD command is sent to cause video to fast forward at 4× speed. The updated PLAY_STATUS messages indicate the change in playback mode. After the appropriate location is reached, the PLAY command is sent, causing playback to return to normal speed, and a PLAY_STATUS message to be generated.

### REPLAY

| | |
|---|---|
| Affects | Any movie zone |
| Command | `REPLAY:` |
| Response | *`status`*`:` |

Skips back five seconds during movie playback. At other times, this command has no effect.

Example

Controller sends:

```
01/0/REPLAY:
```

Kaleidescape System sends:

```
01/0/000:/88
03/!/000:PLAY_STATUS:2:0:01:07195:00061:001:00070:00061:/29
```

In this example, the REPLAY command is sent, causing playback status to go back five seconds. A PLAY_STATUS event message is generated to indicate the change in position.

### CHILD_PLAY

| | |
|---|---|
| Affects | Any zone |
| Command | CHILD_PLAY: |
| Response | *status*: |

If the child user interface is already active, begins playback of movie. If the child user interface is not active, activates the child user interface.

Example

Controller sends:

```
02/8/CHILD_PLAY:
```

Kaleidescape System sends:

```
02/8/000:/97
02/!/000:CHILD_MODE_STATE:1:/63
02/!/000:USER_DEFINED_EVENT:SELECT_KALEIDESCAPE_INPUT:/77
02/!/000:MUSIC_NOW_PLAYING_STATUS:00000:00000:1:0:0000000007::/74
02/!/000:UI_STATE:07:00:00:0:/45
02/!/000:TITLE_NAME:Ice Age:/34
```

The player was not displaying the child user interface, but after receiving the CHILD_PLAY command, switched to the child user interface, and began playback. This selection was available in the Child collection; otherwise, playback would not begin and a movie from the Child collection would be selected.

### CHILD_PAUSE

| | |
|---|---|
| Affects | Any zone |
| Command | CHILD_PAUSE:<br>CHILD_PAUSE_ON:<br>CHILD_PAUSE_OFF: |
| Response | *status*: |

If the child user interface is active, toggles pause on and off. If not, activates the child user interface.

### CHILD_STOP

| | |
|---|---|
| Affects | Any zone |
| Command | CHILD_STOP: |
| Response | *status*: |

Stops the movie and displays the child user interface.

## Playback information

Note: The response code for music related commands will return “Command is not available” for products which do not support music.

### SET_STATUS_CUE_PERIOD

| | |
|---|---|
| Affects | Any zone |
| Command | `SET_STATUS_CUE_PERIOD:`*`period`*`:` |
| Response | *`status`*`:STATUS_CUE_PERIOD:`*`period`*`:` |

Sets the time in seconds between `PLAY_STATUS` event messages to period.

*period* is the time in seconds between `PLAY_STATUS` and `MUSIC_PLAY_STATUS` messages.

`0` No updates for title and chapter locations are sent.

`1` A `PLAY_STATUS/MUSIC_PLAY_STATUS` event message is sent every second as title and chapter locations change.

Note: Setting the period to a value greater than 1 is not yet implemented.

Even when the period is set to `0`, the `PLAY_STATUS/MUSIC_PLAY_STATUS` event message is generated when fields, other than the playback time change (for example, playback mode, playback scanning speed, title number, chapter number). This command only affects whether the event is sent when only the time changes.

This command is useful for getting automatic updates of the play status, for example, to display the time remaining on a touch panel.

Example

Controller sends:

```
03:40:28.144  > 01/4/SET_STATUS_CUE_PERIOD:1:
```

Kaleidescape System sends:

```
03:40:28.160    01/4/000:STATUS_CUE_PERIOD:0001:/47
03:40:28.473    01/!/000:PLAY_STATUS:2:0:01:07136:00135:
                001:00300:00135:/22
03:40:29.473    01/!/000:PLAY_STATUS:2:0:01:07136:00136:001:
                00300:00136:/24
                    …
```

Before this command was sent, movie playback was started. When the `SET_STATUS_CUE_PERIOD` command is sent, the system responds with the updated `STATUS_CUE_PERIOD`. Subsequently, `PLAY_STATUS` messages start being generated once per second.

Note: This command replaces the `ENABLE_STATUS_CUES` command (and associated response fields `STATUS_CUES` and the period), which is still supported for backward compatibility.

### GET_PLAY_STATUS

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GET_PLAY_STATUS:` |
| Response/Event | *status*`:PLAY_STATUS:`*mode*`:`*speed*`:`<br>*title_num*`:`*title_length*`:`*title_loc*`:`<br>*chap_num*`:`*chap_len*`:`*chap_loc*`:` |

This message indicates the current movie play mode and speed, as well as the location in the movie title and chapter structure. The controller can use this information to change, disable/enable, or dim/highlight buttons, or provide other feedback to the user.

This event message is generated when any of the included values changes, typically once per second during playback. This behavior can be changed by `SET_STATUS_CUE_PERIOD`.

*mode* is the current playback mode of the movie.

| | |
|---|---|
| `0` | Nothing playing |
| `1` | Paused (*speed* does not apply)<br>Use `GET_MOVIE_LOCATION` to distinguish between freeze-frame and intermission. |
| `2` | Playing (*speed* does not apply) |
| `3` | unused |
| `4` | Forward scan (*speed* applies) |
| `5` | unused |
| `6` | Reverse scan (*speed* applies) |

*speed* is the speed of the playback scanning and applies only to mode `4` or `6`, indicating scan. Value is an integer between `1` (closest to normal playback speed) and `3` (farthest from normal playback speed).

*title_num* is a zero-padded, two-digit number identifying the current movie title playing. If there is no title for the movie playing, this value is `00`.

*title_length* is a zero-padded, five-digit number providing the total length (in seconds) of the title. If there is no current title, or the value cannot be determined, the value is `00000`.

*title_loc* is a zero-padded, five-digit number providing the current location into the title (in seconds). If there is no current title, or the value cannot be determined, the value is `00000`.

*chap_num* is a zero-padded, three-digit number identifying the current chapter playing. If no chapter is playing, this value is `000`.

*chap_length* is a zero-padded, five-digit number providing the total length in seconds of the chapter. If there is no current chapter, or the value cannot be determined, the value is `00000`.

*chap_loc* is a zero-padded, five-digit number providing the current location into the chapter. If there is no current chapter, or the value cannot be determined, the value is `00000`.

Example

Kaleidescape System sends:

```
01/!/000:PLAY_STATUS:2:0:01:05343:01175:004:00600:00034:/25
```

This event message indicates that the movie is playing (`2`), in the first title (`01`), and the title is 5343 seconds long. Playback is 1175 seconds into the title. The current chapter is the fourth chapter (`004`), which is 600 seconds long. Playback is 34 seconds into the current chapter.

### GET_PLAYING_TITLE_NAME

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GET_PLAYING_TITLE_NAME:` |
| Response/Event | *`status`*`:TITLE_NAME:`*`title`*`:` |

This message provides the title of the movie currently playing. The event message is sent any time playback begins or ends for a new movie.

*title* is the title of the movie currently playing. This is the same title displayed in the Kaleidescape user interface. If there is no active playback the title field is empty.

The controller can display the title of the movie currently playing on a touch screen, for example.

Received messages for escaped characters must always be processed, and it is especially important for `TITLE_NAME` commands because these commands often contain embedded colons. See Message character set on page 24 for details.

Example

Controller sends:

```
01/0/GET_PLAYING_TITLE_NAME:
```

Kaleidescape System sends:

```
01/0/000:TITLE_NAME:Serenity:/25
```

In this example, the title name for the film currently playing is explicitly requested by the `GET_PLAYING_TITLE_NAME` command. The result indicates that the title is Serenity.

Note: This command replaces the `GET_TITLE_NAME` command which returns a similar response, but without the `TITLE_NAME` field, which makes the `GET_TITLE_NAME` command harder to parse.

### GET_MUSIC_NOW_PLAYING_STATUS

| | |
|---|---|
| Affects | Any zone |
| Command | `GET_MUSIC_NOW_PLAYING_STATUS:` |
| Response/Event | *status*`:MUSIC_NOW_PLAYING_STATUS:`*total*`:`<br>*location*`:`*repeat*`:`*random*`:`*generation*`:`<br>*now_playing_handle*`:` |

This event represents the state of the Now Playing list.

*total* is the number of tracks in the list, as a zero-padded, five-digit number.

*location* is the index of the song currently playing, with `0` representing the first position. Provided as a zero-padded, five-digit number.

*repeat* indicates if repeat is on or off. `0` is off; `1` is on.

*random* indicates if random is on or off. `0` is off; `1` is on.

*generation* is a number that changes when the text-based music list accessed by `BROWSE` protocol changes. The number will change for changes in both the static library and the now playing list. The number is a zero-padded, 10 digit number.

The actual value is not as useful as the change alert which can be used to determine whether to refresh the list of music playing.

*now_playing_handle* is a handle unique to the track currently playing.

Example

Controller sends:

```
01.01/4/GET_MUSIC_NOW_PLAYING_STATUS:
```

Kaleidescape System sends:

```
01.01/4/000:MUSIC_NOW_PLAYING_STATUS:00002:00000:1:0:0000000123:2.205cd:
/82
```

The response to `GET_MUSIC_NOW_PLAYING_STATUS` indicates that two songs are listed to play, and the first song is currently playing. Repeat is on, while random playback is off. The generation number is `123`, and the song currently playing is identified by the handle `2.205cd`.

### GET_MUSIC_PLAY_STATUS

| | |
|---|---|
| Affects | Any zone |
| Command | `GET_MUSIC_PLAY_STATUS:` |
| Response/Event | *`status`*`:MUSIC_PLAY_STATUS:`*`mode`*`:`*`speed`*`:`<br>*`length`*`:`*`position`*`:`*`progress`*`:` |

This message identifies the current playback mode, scan speed, track length, position in the current track, and the percentage of progress through the track.

The event message is generated when any of the included values changes, typically once per second. This behavior can be changed by the `SET_STATUS_CUE_PERIOD` message.

When music is not playing, all fields are zero.

*`mode`* is the current playback mode for the music and has the following values when music is playing.

| | |
|---|---|
| 1 | Paused |
| 2 | Normal play |
| 4 | Fast forward |
| 6 | Fast reverse |

*`speed`* Indicates the speed with which music playback is fast forwarding or reversing. Normally has a value of `0`.

In fast forward and fast reverse speed can be `2` or `3` to represent 4x or 16x speed, respectively.

*`length`* is the length of the current track, in seconds, as a zero-padded, five-digit number.

*`position`* is the position within the track, also in seconds as a zero-padded, five-digit number. The position is prepended by a + or – sign. Position can be negative if the song is about to start.

*`progress`* is the percentage of the way through the track, and is between `000.00` and `100.00`.

Example

Controller sends:

```
01.01/6/GET_MUSIC_PLAY_STATUS:
```

Kaleidescape System sends:

```
01.01/6/000:MUSIC_PLAY_STATUS:1:0:00298:+00036:012.08:/99
```

Controller sends:

```
01/7/PLAY:
```

Kaleidescape System sends:

```
01/7/000:/95
03/!/000:MUSIC_PLAY_STATUS:2:0:00298:+00036:012.08:/38
```

In this example, the first `MUSIC_PLAY_STATUS` response indicates that music is paused. Then the `PLAY` command is sent, and in response, a `MUSIC_PLAY_STATUS` event is sent indicating that music is playing normally.

Additionally, the second `MUSIC_PLAY_STATUS` message indicates that the track is 298 seconds long, and that playback is 36 seconds into the track, or 12.08 percent of the way through.

### GET_MUSIC_TITLE

| | |
|---|---|
| Affects | Any zone |
| Command | `GET_MUSIC_TITLE:` |
| Response/Event | *status*:MUSIC_TITLE:*track*:*artist*:*album*:<br>*track_handle*:*album_handle*:<br>*now_playing_handle*: |

This message provides useful information about the currently playing music track, such as the album it is associated with, the artist that wrote it, etc.

The event message is generated when the song currently playing changes. If no song is playing, all of the fields are empty.

*track* is the name of the song playing.

*artist* is the name of the track artist.

*album* is the name of the album that contains the song.

*track_handle* and *album_handle*
are handles for the track and the album with that track that can be passed to `GET_CONTENT_DETAILS` to get more information about the track or album.

*now_playing_handle* represents the track as it is positioned in the Now Playing list. The *now_playing_handle* can be passed to the `PERFORM_ACTION` command to jump to that track in the list.

The *track_handle*, *album_handle*, and *now_playing_handle* values are unique. For example, if the same track is in the Now Playing list twice, the track has the same *track_handle* but different *now_playing* handles.

Example

Controller sends:

```
01.01/0/GET_MUSIC_TITLE:
```

Kaleidescape System sends:

```
01.01/0/000:MUSIC_TITLE:Rubber Soul:The Beatles:Nowhere Man:
1.96de0c01d6fd4a9e-t30c_1951:1.R_955059:2.205cd:/97
```

The response to this `GET_MUSIC_TITLE` command indicates that the current track is called Nowhere Man by the Beatles from their Rubber Soul album. The handles provided can be used to get more information to display on a controller, for example, cover art.

## Music playback controls

> Note: The response code for music related commands will return "Command is not available" for products which do not support music.

### MUSIC_RANDOM_ON

| | |
|---|---|
| Affects | Any zone |
| Command | `MUSIC_RANDOM_ON:` |
| Response | *`status`*`:` |

Turns on random music playback in the zone. When random music playback is on, songs play in a random order. The state of random persists across component reboots. Random music playback can also be set in the Now Playing view on the user interface.

See also `MUSIC_RANDOM_TOGGLE` and `MUSIC_RANDOM_OFF`.

Example

Controller sends:

```
01.01/5/MUSIC_RANDOM_ON:
```

Kaleidescape System sends:

```
01.01/5/000:/36
03/!/000:MUSIC_NOW_PLAYING_STATUS:00000:00000:1:1:0000000007::/76
```

This example shows random music playback being turned on. The subsequent `MUSIC_NOW_PLAYING_STATUS` event message confirms this status in the appropriate field.

### MUSIC_RANDOM_OFF

| | |
|---|---|
| Affects | Any zone |
| Command | `MUSIC_RANDOM_OFF:` |
| Response | *`status`*`:` |

Turns off random music playback in the zone. The state of random persists across component reboots. Random music playback can also be set in the Now Playing view of the user interface.

See also `MUSIC_RANDOM_TOGGLE` and `MUSIC_RANDOM_ON`.

Example

Controller sends:

```
01.01/6/MUSIC_RANDOM_OFF:
```

Kaleidescape System sends:

```
01.01/6/000:/37
03/!/000:MUSIC_NOW_PLAYING_STATUS:00000:00000:1:0:0000000008::/76
```

This example shows random music playback being turned off. The subsequent `MUSIC_NOW_PLAYING_STATUS` event message confirms this status in the appropriate field.

### MUSIC_RANDOM_TOGGLE

| | |
|---|---|
| Affects | Any zone |
| Command | `MUSIC_RANDOM_TOGGLE:` |
| Response | *`status`*`:` |

Toggles random music playback of the zone on and off. When random music playback is on, songs play in a random order. The state of random persists across component reboots. Random music playback can also be set in the Now Playing view of the user interface.

See also `MUSIC_RANDOM_ON` and `MUSIC_RANDOM_OFF`.

Example

Controller sends:

```
01.01/7/MUSIC_RANDOM_TOGGLE:
```

Kaleidescape System sends:

```
01.01/7/000:/38
03/!/000:MUSIC_NOW_PLAYING_STATUS:00000:00000:1:1:0000000009::/78
```

Controller sends:

```
01.01/8/MUSIC_RANDOM_TOGGLE:
```

Kaleidescape System sends:

```
01.01/8/000:/39
03/!/000:MUSIC_NOW_PLAYING_STATUS:00000:00000:1:0:0000000010::/69
```

In this example, the first `MUSIC_RANDOM_TOGGLE` command turns the random play feature on, confirmed by the subsequent `MUSIC_NOW_PLAYING_STATUS` message. The next `MUSIC_RANDOM_TOGGLE` command turns the random play feature back off and generates another `MUSIC_NOW_PLAYING_STATUS` message.

### MUSIC_REPEAT_ON

| | |
|---|---|
| Affects | Any zone |
| Command | `MUSIC_REPEAT_ON:` |
| Response | *`status`*`:` |

Turns on repeat music playback in the zone. If repeat music playback is on when music playback ends, music playback starts again at the beginning. The state of repeat persists across component reboots. Repeat music playback can also be set in the Now Playing view of the user interface.

See also `MUSIC_REPEAT_TOGGLE` and `MUSIC_REPEAT_OFF`.

Example

Controller sends:

```
01.01/3/MUSIC_REPEAT_ON:
```

Kaleidescape System sends:

```
01.01/3/000:/34
03/!/000:MUSIC_NOW_PLAYING_STATUS:00000:00000:1:0:0000000012::/71
```

This example shows repeat music playback being turned on, and the resulting MUSIC_NOW_PLAYING_STATUS event message with the results.

### MUSIC_REPEAT_OFF

| | |
|---|---|
| Affects | Any zone |
| Command | MUSIC_REPEAT_OFF: |
| Response | *status*: |

Turns off repeat music playback in the zone. The state of repeat persists across component reboots. Repeat music playback can also be set in the Now Playing view of the user interface.

See also MUSIC_REPEAT_TOGGLE and MUSIC_REPEAT_ON.

Example

Controller sends:

```
01.01/8/MUSIC_REPEAT_OFF:
```

Kaleidescape System sends:

```
01.01/8/000:/39
03/!/000:MUSIC_NOW_PLAYING_STATUS:00002:00000:0:0:0000000133::/76
```

This example shows repeat music playback being turned off. Subsequently a MUSIC_NOW_PLAYING_STATUS event message is generated to confirm that repeat is off.

### MUSIC_REPEAT_TOGGLE

| | |
|---|---|
| Affects | Any zone |
| Command | MUSIC_REPEAT_TOGGLE: |
| Response | *status*: |

Toggles repeat music playback on and off for the zone. If repeat music playback is on when music playback ends, playback starts again from the beginning. The state of repeat persists across component reboots. Repeat music playback can also be set in the Now Playing view of the user interface.

See also MUSIC_REPEAT_ON and MUSIC_REPEAT_OFF.

Example

Controller sends:

```
01.01/4/MUSIC_REPEAT_TOGGLE:
```

Kaleidescape System sends:

```
01.01/4/000:/35
03/!/000:MUSIC_NOW_PLAYING_STATUS:00000:00000:0:0:0000000013::/71
```

Controller sends:

```
01.01/5/MUSIC_REPEAT_TOGGLE:
```

Kaleidescape System sends:

```
01.01/5/000:/36
03/!/000:MUSIC_NOW_PLAYING_STATUS:00000:00000:1:0:0000000014::/73
```

In this example, the first `MUSIC_REPEAT_TOGGLE` command turns music repeat off, which generates a `MUSIC_NOW_PLAYING_STATUS` event message, indicating the change. The next command turns repeat back on and generates a new `MUSIC_NOW_PLAYING_STATUS` event message with the change.

### GET_CONTROLLED_ZONE

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GET_CONTROLLED_ZONE:` |
| Response | *`status`*`:CONTROLLED_ZONE:#`*`sn.zn`*`:` |

A movie zone can control any music zone in the system. This command can be used by a controller to provide feedback on which music zone is currently being controlled.

The currently controlled music zone can be changed by sending a `SET_CONTROLLED_ZONE` command or through the onscreen display on the Now Playing view.

| | |
|---|---|
| *sn* | is the serial number of the component with the music zone being controlled. |
| *zn* | is the music zone (01–04) being controlled. |

Example

Controller sends:

```
01/6/GET_CONTROLLED_ZONE:
```

Kaleidescape System sends:

```
01/6/000:CONTROLLED_ZONE:#00000000019A.01:/60
```

The response to the `GET_CONTROLLED_ZONE` command indicates that the movie zone on the attached component is currently controlling the first music zone on the component with serial number 0000 0000019A.

### SET_CONTROLLED_ZONE

| | |
|---|---|
| Affects | Any movie zone |
| Command | `SET_CONTROLLED_ZONE:#`*`sn.zn`*`:` |
| Response | *`status`*`:` |

A movie zone can control any music-only zone in the system. This command can be used to change which music zone is currently being controlled by the onscreen display.

The currently controlled music zone can also be changed through the onscreen display on the Now Playing view.

Remote music zone control must be enabled for this command to function, otherwise an error is generated. To enable remote music zone control, go to the browser interface, select the General tab, and select the appropriate option next to Music Zone Control.

| | |
|---|---|
| *sn* | is the serial number of the component to be controlled. |
| *zn* | is the music zone (01–04) to be controlled. |

Example

Controller sends:

```
01/1/SET_CONTROLLED_ZONE:#000000120B91.02:
```

Kaleidescape System sends:

```
01/1/000:CONTROLLED_ZONE:#000000120B91.02:/60
```

In this example, the controlled music zone for the connected component is set to the second music zone of the component with serial number 0000 00120B91. The `CONTROLLED_ZONE` response confirms this change.

## DVD/Blu-ray Disc navigation

### DISC_MENU

| | |
|---|---|
| Affects | Any movie zone |
| Command | `DISC_MENU:` |
| Response | *`status`*`:` |

During movie playback, displays the DVD or Blu-ray Disc menu for the current playback context (which might not be the top level menu). At other times, this command has no effect, i.e., this command does not begin playing the disc. To display the top level menu for the disc, send a `DISC_TOP_MENU` command.

To resume playback while in the disc menu, use the `DISC_RESUME` or `PLAY` command.

If the controller has room for separate menu buttons for the Kaleidescape System and DVD/Blu-ray Disc, use this command. Otherwise, use the context-sensitive `DISC_OR_KALEIDESCAPE_MENU` command.

Example 1

Controller sends:

```
01/3/DISC_MENU:
```

Kaleidescape System sends:

```
01/3/000:/91
03/!/000:PLAY_STATUS:2:0:00:00000:00000:000:00000:00000:/84
```

In this example, DISC_MENU stopped playback of a DVD movie and displayed the DVD menu. The PLAY_STATUS event message confirms that playback has halted.

Example 2

Controller sends:

```
01/0/DISC_MENU:
```

Kaleidescape System sends:

```
01/0/000:/88
03/!/000:PLAY_STATUS:2:0:01:07195:00070:002:00077:00000:/30
03/!/000:PLAY_STATUS:2:0:01:07195:00070:002:00076:00000:/29
```

In the second example, DISC_MENU was sent while a Blu-ray movie was playing and playback continued while the menu was overlaid. This behavior can be the same that BLURAY_POPUP_MENU_TOGGLE causes as shown by the PLAY_STATUS event messages.

> Note: This command replaces the DVD_MENU command which is still supported.

DISC_TOP_MENU

| | |
|---|---|
| Affects | Any movie zone |
| Command | DISC_TOP_MENU: |
| Response | *status*: |

During playback, displays the menu for the DVD or Blu-ray Disc as a whole, rather than any menus defined for specific playback. At other times, this command has no effect, i.e., this command does not begin playing the disc. To display the disc menu for the current playback (which might not be the top level menu), send a DISC_MENU command. To resume playback from the menu, use the DISC_RESUME or PLAY commands.

Example

Controller sends:

```
01/5/DISC_TOP_MENU:
```

Kaleidescape System sends:

```
01/5/000:/93
03/!/000:PLAY_STATUS:2:0:00:00000:00000:000:00000:00000:/84)
```

In this example, the DISC_TOP_MENU command stops playback of a DVD movie and causes the DVD menu to be displayed. The PLAY_STATUS event message confirms the change.

> Note: This command replaces the DVD_TOP_MENU command which is still supported.

### DISC_RESUME

| | |
|---|---|
| Affects | Any movie zone |
| Command | `DISC_RESUME:` |
| Response | *status*: |

If a DVD or Blu-ray Disc menu is active, resumes playback from the point of interruption. At other times, this command has no effect. For most discs a PLAY command has the same effect.

Example

Controller sends:

```
01/1/DISC_RESUME:
```

Kaleidescape System sends:

```
01/1/000:/89
01/!/000:PLAY_STATUS:2:0:01:07152:00692:003:00163:00002:/30
```

In this example, DVD playback is currently halted, and the DVD menu is displayed. The DISC_RESUME command causes the menu to be dismissed and playback to resume, as evidenced by the PLAY_STATUS event message.

> Note: This command replaces the DVD_RESUME command, which is still supported.

### START_CHAPTER_ENTRY

| | |
|---|---|
| Affects | Any movie zone |
| Command | `START_CHAPTER_ENTRY:` |
| Response | *status*: |

During movie playback, displays a tab indicating that the user can enter a chapter number to skip directly to that chapter.

Follow this command with KEYBOARD_CHARACTER commands to supply the number to skip to, and a SELECT command to end entry and make the jump.

If no subsequent digits, or the select command, are received within a few seconds, this command times out and automatically jumps to the specified chapter number as though SELECT had been pressed. See also START_DISC_TITLE_ENTRY.

Example

Controller sends:

```
01/1/START_CHAPTER_ENTRY:
```

Kaleidescape System sends:

```
01/1/000:/89
01/!/000:USER_INPUT:02:Jump to Chapter::/31
```

Controller sends:

```
01/1/KEYBOARD_CHARACTER:5:
```

Kaleidescape System sends:

```
01/1/000:/89
01/!/000:USER_INPUT:02:Jump to Chapter:5:/84
```

Controller sends:

```
01/1/SELECT:
```

Kaleidescape System sends:

```
01/1/000:/89
01/!/000:USER_INPUT:00:::/15
01/!/000:PLAY_STATUS:2:0:01:07152:01226:005:00271:00000:/24
```

In this example, the START_CHAPTER_ENTRY command is sent during movie playback, causing the prompt to appear. Because the system is requesting user input, a USER_INPUT event message is sent with the appropriate prompt. The controller sends the number 5 to the system to indicate that chapter 5 is required. The system responds with an updated USER_INPUT event message. The controller finally sends the SELECT command to confirm the chapter number, causing playback to jump to that chapter, as shown by the PLAY_STATUS event message containing the new chapter update. The USER_INPUT event message is then generated to indicate that user input is no longer required.

### START_DISC_TITLE_ENTRY

| | |
|---|---|
| Affects | Any movie zone |
| Command | START_DISC_TITLE_ENTRY: |
| Response | *status*: |

Some movies have various titles that can be selected during playback. Titles can be episodes, menus, trailers, etc. The main movie is usually title number 1. This command displays a tab so the user can enter a title number to skip directly to that title.

Follow this command with KEYBOARD_CHARACTER commands to supply the number to skip to, and a SELECT command to end entry and make the jump.

If no subsequent digits, or the SELECT command, are received within a few seconds, this command times out and automatically jumps to the specified title as though SELECT had been pressed.

Users are unlikely to need to jump to a numbered title, but this feature can be useful for calibration discs that have several titles. Users often prefer to access different chapters, which can be enabled with the START_CHAPTER_ENTRY command.

Example

Controller sends:

```
01/1/START_DISC_TITLE_ENTRY:
```

Kaleidescape System sends:

```
01/1/000:/89
01/!/000:USER_INPUT:02:Jump to Title::/34
```

Controller sends:

```
01/1/KEYBOARD_CHARACTER:1:
```

Kaleidescape System sends:

```
01/1/000:/89
01/!/000:USER_INPUT:02:Jump to Title:1:/83
```

Controller sends:

```
01/1/SELECT:
```

Kaleidescape System sends:

```
01/1/000:/89
01/!/000:USER_INPUT:00:::/15
01/!/000:PLAY_STATUS:2:0:01:07152:00000:001:00394:00000:/15
```

In this example, the START_DISC_TITLE_ENTRY command is sent during movie playback, causing the prompt to appear. Because the system is requesting user input, a USER_INPUT event message is sent with the appropriate prompt. The controller sends the number 1 to the system to indicate that title 1 is required. The system responds with an updated USER_INPUT event message. The controller finally sends the SELECT command to confirm the chapter number, causing playback to jump to that title, as shown by the PLAY_STATUS event message containing the new chapter update. The USER_INPUT event message is then generated to indicate that user input is no longer required.

Note: This command replaces the START_DVD_TITLE_ENTRY command.

## Movie playback options

### SHOW_NAVIGATION_OVERLAY

| | |
|---|---|
| Affects | Any movie zone |
| Command | SHOW_NAVIGATION_OVERLAY: |
| Response | *status*: |

During movie playback, this command brings up the movie overlay showing the chapter/title navigation option. This command brings up the same overlay as the STATUS_AND_SETTINGS command, just on a different page. At other times, this command has no effect.

Example

Controller sends:

```
01/4/SHOW_NAVIGATION_OVERLAY:
```

Kaleidescape System sends:

```
01/4/000:/92
03/!/000:UI_STATE:07:03:00:0:/49
```

This example shows the results of the `SHOW_NAVIGATION_OVERLAY` command. The command was sent during playback, causing the `UI_STATE` event message to be generated, indicating that navigation option of the movie overlay is now active.

### STATUS_AND_SETTINGS

| | |
|---|---|
| Affects | Any movie zone |
| Command | `STATUS_AND_SETTINGS:` |
| Response | *`status`*`:` |

During movie playback, toggles display of the movie overlay. The movie overlay has a status page showing the title of the movie currently playing, remaining playing time, and other information. Pressing the left and right arrow buttons displays other options of the movie overlay, for example, audio, subtitles, navigation, scenes.

When not playing back a movie, this command displays the System Status view (see `GO_SYSTEM_STATUS`).

Example

Controller sends:

```
01/2/STATUS_AND_SETTINGS:
```

Kaleidescape System sends:

```
01/2/000:/90
03/!/000:UI_STATE:07:02:00:0:/48
```

Controller sends:

```
01/3/STATUS_AND_SETTINGS:
```

Kaleidescape System sends:

```
01/3/000:/91
03/!/000:UI_STATE:07:00:00:0:/46
```

In this example, the `STATUS_AND_SETTINGS` command is sent during playback, causing the movie overlay to appear. A `UI_STATE` message is generated to show that the status page is displayed. The second `STATUS_AND_SETTINGS` command hides the movie overlay and causes another `UI_STATE` event message to be generated showing the change.

### INTERMISSION_ON

| | |
|---|---|
| Affects | Any movie zone |
| Command | `INTERMISSION_ON:` |
| Response | *`status`*`:` |

Pauses any active playback and displays the Kaleidescape intermission screen, which shows the cover of the currently playing movie. When playing back a script, the intermission feature can be customized to show other cover art, scenes, or content. At other times, this command has no effect.

See also `INTERMISSION_OFF` and `INTERMISSION_TOGGLE`.

Example

Controller sends:

```
01/4/INTERMISSION_ON:
```

Kaleidescape System sends:

```
01/4/000:/92
03/!/000:MOVIE_LOCATION:04:/71
03/!/000:PLAY_STATUS:1:0:05:05700:00638:001:00887:00638:/58
```

This example shows the results of sending the `INTERMISSION_ON` command. Movie playback stops, generating a `PLAY_STATUS` event message. Additionally, a `MOVIE_LOCATION` event message is generated to indicate that the intermission screen is displayed, which can be used to trigger lighting and other events to coincide with the intermission.

### INTERMISSION_OFF

| | |
|---|---|
| Affects | Any movie zone |
| Command | `INTERMISSION_OFF:` |
| Response | *`status`*`:` |

During movie playback when the intermission screen is displayed, removes the intermission screen and resumes playback. At other times, this command has no effect.

See also `INTERMISSION_ON` and `INTERMISSION_TOGGLE`.

Example

Controller sends:

```
01/5/INTERMISSION_OFF:
```

Kaleidescape System sends:

```
01/5/000:/93
03/!/000:MOVIE_LOCATION:03:/70
03/!/000:PLAY_STATUS:2:0:05:05700:00638:001:00887:00638:/59
```

This example shows the results after sending the `INTERMISSION_OFF` command when intermission is active. The intermission screen vanishes, returning to movie playback, generating a `MOVIE_LOCATION` event message. Additionally, the `PLAY_STATUS` event message is generated to show that playback has been resumed.

### INTERMISSION_TOGGLE

| | |
|---|---|
| Affects | Any movie zone |
| Command | `INTERMISSION_TOGGLE:` |
| Response | *`status`*`:` |

During movie playback, toggles display of the Kaleidescape intermission screen. At other times, this command has no effect.

See also `INTERMISSION_OFF` and `INTERMISSION_ON`.

Example

Controller sends:

```
01/0/INTERMISSION_TOGGLE:
```

Kaleidescape System sends:

```
01/0/000:/88
03/!/000:MOVIE_LOCATION:04:/71
03/!/000:PLAY_STATUS:1:0:05:05700:00684:001:00887:00684:/60
```

Controller sends:

```
01/1/INTERMISSION_TOGGLE:
```

Kaleidescape System sends:

```
01/1/000:/89
03/!/000:MOVIE_LOCATION:03:/70
03/!/000:PLAY_STATUS:2:0:05:05700:00684:001:00887:00684:/61
```

In this example, the first `INTERMISSION_TOGGLE` command activates intermission. This generates a `MOVIE_LOCATION` message and a `PLAY_STATUS` message showing that playback has paused. The second `INTERMISSION_TOGGLE` command removes the intermission screen and returns to playback, generating new `MOVIE_LOCATION` and `PLAY_STATUS` event messages.

### SET_FAVORITE_SCENE_START

| | |
|---|---|
| Affects | Any movie zone |
| Command | `SET_FAVORITE_SCENE_START:` |
| Response | *`status`*`:` |

During movie playback, records a bookmark marking the start of a scene at the time the movie zone receives the command. If a start of a scene bookmark already exists, the original bookmark is replaced with the new time. At other times, this command has no effect.

If the current movie has an end of scene bookmark (see SET_FAVORITE_SCENE_END), the end bookmark is erased. A scene with no end bookmark continues playing to the end of the movie.

A movie can have one unnamed scene. Scenes are named using the movie details page or the overlay during playback, to allow several scenes per movie. Scenes can be played in sequences with the scripts feature. See the PLAY_SCRIPT command.

Start and end bookmarks for scenes can also be set from the movie overlay during playback, accessed by sending the STATUS_AND_SETTINGS command.

Example

Controller sends:

```
01/2/SET_FAVORITE_SCENE_START:
```

Kaleidescape System sends:

```
01/2/000:/90
```

Note: This command replaces the SET_FAVORITE_SCENE command.

### SET_FAVORITE_SCENE_END

| | |
|---|---|
| Affects | Any movie zone |
| Command | SET_FAVORITE_SCENE_END: |
| Response | *status*: |

During movie playback, records a bookmark marking the end of a scene at the time the movie zone receives the command. If a bookmark indicating the end of a scene already exists, the original is replaced with the new time.

The start of a scene must be defined before the end of scene can be defined. (See SET_FAVORITE_SCENE_START.) A scene with no ending bookmark continues playing to the end of the movie.

At other times, or if a bookmark marking the start of a scene has not been saved, this command has no effect. Scenes can be played in sequences with the scripts feature. See the PLAY_SCRIPT command.

Start and end bookmarks for scenes can also be set from the movie overlay during playback, accessed by sending the STATUS_AND_SETTINGS command.

Example

Controller sends:

```
01/3/SET_FAVORITE_SCENE_END:
```

Kaleidescape System sends:

```
01/3/000:/91
```

## START_SEND_NUMBER_TO_DISC_ENTRY

| | |
|---|---|
| Affects | Any movie zone |
| Command | `START_SEND_NUMBER_TO_DISC_ENTRY:` |
| Response | *`status`*`:` |

Some DVD and Blu-ray Discs require number keys to access part of the content. These keys are usually reserved for interactive DVDs and Blu-ray Discs that allow entering numbers to select from lists, as well as hidden Easter eggs on these discs.

To send a number key to the disc, use this command during movie playback. A tab appears onscreen to prompt the user to enter a number. Subsequent KEYBOARD_CHARACTER commands show in this tab, after which a SELECT command removes the tab and sends the number to the disc. A CANCEL command removes the tab without sending the number.

Example

Controller sends:

```
01/2/START_SEND_NUMBER_TO_DISC_ENTRY:
```

Kaleidescape System sends:

```
01/2/000:/90
03/!/000:USER_INPUT:02:Send Number to Disc::/40
```

Controller sends:

```
01/3/KEYBOARD_CHARACTER:2:
```

Kaleidescape System sends:

```
01/3/000:/91
03/!/000:USER_INPUT:02:Send Number to Disc:2:/90
```

Controller sends:

```
01/4/SELECT:
```

Kaleidescape System sends:

```
01/4/000:/92
```

In this example, the START_SEND_NUMBER_TO_DISC_ENTRY command is sent during movie playback. The system displays a tab prompting the user to enter a character and a USER_INPUT event message is generated indicating the need for the keypad. The KEYBOARD_CHARACTER command sends the number 2, generating a new USER_INPUT event message with the new data. The final SELECT command then sends the number to the disc.

> Note: This command replaces the START_SEND_NUMBER_TO_DVD_ENTRY command.

### ANGLE_NEXT

| | |
|---|---|
| Affects | Any movie zone |
| Command | `ANGLE_NEXT:` |
| Response | `*status*:` |

Changes to the next camera angle defined for video playback, looping back to the first camera angle if at the last camera angle. See also `ANGLE_PREVIOUS` and `GET_CAMERA_ANGLE`.

Example

Controller sends:

```
01/5/ANGLE_NEXT:
```

Kaleidescape System sends:

```
01/5/000:/93
```

### ANGLE_PREVIOUS

| | |
|---|---|
| Affects | Any movie zone |
| Command | `ANGLE_PREVIOUS:` |
| Response | `*status*:` |

Changes to the previous camera angle defined for video playback, looping to the last camera angle if currently at the first camera angle. See also `ANGLE_NEXT` and `GET_CAMERA_ANGLE`.

Example

Controller sends:

```
01/6/ANGLE_PREVIOUS:
```

Kaleidescape System sends:

```
01/6/000:/94
```

### AUDIO_NEXT

| | |
|---|---|
| Affects | Any movie zone |
| Command | `AUDIO_NEXT:` |
| Response | `*status*:` |

During movie playback, changes to the next audio stream. At other times, this command has no effect. The user can also change audio streams through the overlay displayed by the `STATUS_AND_SETTINGS` command, or through the current DVD or Blu-ray Disc menu.

Example

Controller sends:

```
01/7/AUDIO_NEXT:
```

Kaleidescape System sends:

```
01/7/000:/95
```

### SUBTITLES_NEXT

| | |
|---|---|
| Affects | Any movie zone |
| Command | `SUBTITLES_NEXT:` |
| Response | *`status`*`:` |

During playback, changes to the next subtitle stream. At other times, this command has no effect. The user can also change subtitle streams through the overlay displayed by the `STATUS_AND_SETTINGS` command or through current DVD or Blu-ray Disc menu.

Example

Controller sends:

```
01/9/SUBTITLES_NEXT:
```

Kaleidescape System sends:

```
01/9/000:/97
```

### GET_CAMERA_ANGLE

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GET_CAMERA_ANGLE:` |
| Response/Event | *`status`*`:CAMERA_ANGLE:`*`cur_angle`*`:`<br>*`num_angles`*`:`*`in_angle_block`*`:` |

This message provides information about the current camera angle and what camera angles are available. Information is generated as an event message when the current camera angle changes, or if the number of available camera angles changes.

*`cur_angle`* is a single digit from 1–9 that indicates which angle is currently active.

*`num_angles`* is a single digit from 1–9 that indicates how many angles are available at the current playback location. A controller can use these fields to provide information to the user about available camera angles.

*`in_angle_block`* indicates whether multiple camera angles are available. If set to `0`, there are no extra camera angles and the other fields are meaningless. If set to `1`, there are multiple camera angles available and the other fields indicate the current and available camera angles.
A controller can use this field to enable or disable buttons to switch between camera angles. (See the `ANGLE_NEXT` and `ANGLE_PREVIOUS` commands.)

Example

Controller sends:

```
01/1/GET_CAMERA_ANGLE:
```

Kaleidescape System sends:

```
01/1/000:CAMERA_ANGLE:1:1:0:/46
```

In this example, the response to the `GET_CAMERA_ANGLE` command indicates that there is only one angle available, and that it is currently active.

## Blu-ray Disc playback options

### Blu-ray color buttons

| | | |
|---|---|---|
| `RED_PRESS` | `RED_RELEASE` | `RED` |
| `GREEN_PRESS` | `GREEN_RELEASE` | `GREEN` |
| `BLUE_PRESS` | `BLUE_RELEASE` | `BLUE` |
| `YELLOW_PRESS` | `YELLOW_RELEASE` | `YELLOW` |

| | |
|---|---|
| Affects | Any movie zone |
| Command | `RED_PRESS:` (same for other color button commands) |
| Response | *`status`*`:` |

During Blu-ray Disc playback, performs actions associated with the color button.

These actions are defined by the disc itself and can vary from disc to disc. It is recommended, but not necessary, that these buttons are displayed to the user in a row, in the following order: Red, Green, Blue, Yellow.

The `_PRESS` and `_RELEASE` command pairs allow compatible Blu-ray Discs to detect when the user is holding down the associated color button. Use the `_PRESS` and `_RELEASE` commands instead of the plain color button commands (the right most column above) if the controller supports press and release handling.

Send a `_PRESS` command when the corresponding button is first pressed and a `_RELEASE` when that button is released.

The plain color button commands, `RED`, `GREEN`, `BLUE`, and `YELLOW`, are available for controllers that do not support press and release handling. Each plain color command represents a single button press of the colored button.

Example

Controller sends:

```
01/0/RED:
```

Kaleidescape System sends:

```
01/0/000:/88
```

### GET_MOVIE_MEDIA_TYPE

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GET_MOVIE_MEDIA_TYPE:` |
| Response/Event | *`status`*`:MOVIE_MEDIA_TYPE:`*`media_type`*`:` |

This message indicates the type of media currently being played. The event message is generated when the media type changes.

| | | |
|---|---|---|
| *media_type* | shows the media type currently being played. | |
| | 00 | No media playing |
| | 01 | DVD |
| | 02 | Video stream |
| | 03 | Blu-ray Disc |

This information can be used to change the controls made available to the user. For example, during Blu-ray Disc playback, the Blu-ray color buttons can be displayed to the user. During video stream playback, disc menu keys can be hidden because these streams do not have menus.

Example

Controller sends:

```
01/9/GET_MOVIE_MEDIA_TYPE:
```

Kaleidescape System sends:

```
01/9/000:MOVIE_MEDIA_TYPE:01:/58
```

This example response to the `GET_MOVIE_MEDIA_TYPE` command indicates that a DVD is currently being played.

### BLURAY_SPECIAL_STOP

| | |
|---|---|
| Affects | Any movie zone |
| Command | `BLURAY_SPECIAL_STOP:` |
| Response | *status*: |

During Blu-ray playback, sends the `BLURAY_SPECIAL_STOP` command to the disc currently playing. For all other content, this command behaves like the `STOP` command.

⚠ CAUTION

USING THIS COMMAND CAN TRAP THE USER.
Depending on how the disc was authored, this command does not always return the user to a Kaleidescape movie view. Some discs use this command to access special features. If used as the only stop command, the controller must provide another mechanism to return to the Kaleidescape menu or a Kaleidescape movie view.

Example

Controller sends:

```
01/8/BLURAY_SPECIAL_STOP:
```

Kaleidescape System sends:

```
01/8/000:/96
```

### BLURAY_POPUP_MENU_TOGGLE

| | |
|---|---|
| Affects | Any movie zone |
| Command | `BLURAY_POPUP_MENU_TOGGLE:` |
| Response | *`status`*`:` |

During Blu-ray Disc playback, toggles the display of the Blu-ray Disc pop-up menu. The specific function of this command is defined by the disc itself and can vary from disc to disc. For many discs, this command functions the same as the `DISC_MENU` command.

Example

Controller sends:

```
01/4/BLURAY_POPUP_MENU_TOGGLE:
```

Kaleidescape System sends:

```
01/4/000:/92
```

## Context-Sensitive Commands

Context-sensitive commands behave differently when used playing a movie or when in the user interface. These commands are especially useful when space is limited on a touch panel or when using an IR remote. Table 10 lists context-sensitive commands.

Table 10: Context-sensitive command summary

| Command | Description |
|---|---|
| `STOP_OR_CANCEL` | Behaves like `STOP` during movie playback or when sent directly to a music zone.<br>Behaves like `CANCEL` in the user interface. |
| `DISC_OR_KALEIDESCAPE_MENU` | Behaves like `DISC_MENU` during movie or music playback.<br>Behaves like `KALEIDESCAPE_MENU_TOGGLE` in user interface. |
| Paging and skipping | Behave like `NEXT` or `PREVIOUS` during movie playback.<br>Behave like `PAGE_UP` or `PAGE_DOWN` in the user interface. |

### STOP_OR_CANCEL

| | |
|---|---|
| Affects | Any zone |
| Command | `STOP_OR_CANCEL:` |
| Response | *`status`*`:` |

- Behaves like `STOP` during movie playback, in the Now Playing view, or when sent directly to a music zone.

- Behaves like `CANCEL` in the user interface.

The Stop button on remote controls is usually mapped to this command.

Example 1 During movie playback

Controller sends:

```
01/9/STOP_OR_CANCEL:
```

Kaleidescape System sends:

```
01/9/000:/97
03/!/000:UI_STATE:03:00:00:0:/42
…
03/!/000:PLAY_STATUS:0:0:00:00000:00000:000:00000:00000:/82
```

In this example, the `STOP_OR_CANCEL` command acts as a `STOP` command, causing the onscreen display to return to the Movie Covers view, which is confirmed by the `UI_STATE` message.

Example 2 In the user interface

Controller sends:

```
01/2/STOP_OR_CANCEL:
```

Kaleidescape System sends:

```
01/2/000:/90
03/!/000:UI_STATE:09:00:00:0:/48
```

In this second example, the onscreen display is in the Music List view showing the details page for an album. When the `STOP_OR_CANCEL` command is sent, the details page is dismissed and a `UI_STATE` event message is generated with the updated state.

### DISC_OR_KALEIDESCAPE_MENU

| | |
|---|---|
| Affects | Any movie zone |
| Command | `DISC_OR_KALEIDESCAPE_MENU:` |
| Response | *`status`*`:` |

- Behaves like `DISC_MENU` during movie playback.
- Behaves like `KALEIDESCAPE_MENU_TOGGLE` in the user interface.

The Menu button on a remote control is usually mapped to this command.

Example 1 During movie playback

Controller sends:

```
01/5/DISC_OR_KALEIDESCAPE_MENU:
```

Kaleidescape System sends:

```
01/5/000:/93
03/!/000:PLAY_STATUS:2:0:00:00000:00000:000:00000:00000:/84
```

In the first example, the DISC_OR_KALEIDESCAPE_MENU command is sent during movie playback, causing the movie menu to be displayed and a PLAY_STATUS event message is generated.

Example 2 In the user interface

Controller sends:

```
01/1/DISC_OR_KALEIDESCAPE_MENU:
```

Kaleidescape System sends:

```
01/1/000:/89
03/!/000:UI_STATE:09:00:01:0:/49
```

In the second example, no movie is playing when the DISC_OR_KALEIDESCAPE_MENU command is sent. The Kaleidescape menu is displayed and a UI_STATE event message is generated with updated information.

> Note: This command replaces the DVD_OR_KALEIDESCAPE_MENU command, which is still supported

### Paging and skipping

| | |
|---|---|
| Affects | Any movie zone |
| Command | PAGE_DOWN_OR_NEXT:<br>PAGE_DOWN_OR_NEXT_PRESS:<br>PAGE_DOWN_OR_NEXT_RELEASE:<br>PAGE_DOWN_OR_PREVIOUS:<br>PAGE_DOWN_OR_PREVIOUS_PRESS:<br>PAGE_DOWN_OR_PREVIOUS_RELEASE:<br>PAGE_UP_OR_NEXT:<br>PAGE_UP_OR_NEXT_PRESS:<br>PAGE_UP_OR_NEXT_RELEASE:<br>PAGE_UP_OR_PREVIOUS:<br>PAGE_UP_OR_PREVIOUS_PRESS:<br>PAGE_UP_OR_PREVIOUS_RELEASE: |
| Response | *status*: |

These commands are used when mapping Next/Previous buttons on a remote or small touch panel.

- Behaves like the NEXT and PREVIOUS commands during movie playback or in the Now Playing view.
- Behaves like the PAGE_UP and PAGE_DOWN commands in the user interface.

Each command has a single-shot version, which should be avoided, and _PRESS and _RELEASE versions, which are preferred because these commands allow the user to page through long lists.

Use either the UP/PREVIOUS and DOWN/NEXT pair, or use the UP/NEXT and DOWN/PREVIOUS pair depending on the physical configuration of the remote control or touch panel buttons. For example, it is a common for a remote control to have these buttons placed vertically with the top button labeled with a plus sign and the bottom button with a minus sign. The top (plus) button must skip to the next track or chapter, but because the button is located on top, this button must also page up. In that case, map the top button to UP/NEXT and the bottom button to DOWN/PREVIOUS.

In a horizontal layout, it is more practical to map the left button to UP/PREVIOUS and the right button to DOWN/NEXT. Layout has to be what feels more natural including the labeling involved.

> Note: The UP/PREVIOUS commands replace the SKIP_REVERSE command; and the DOWN/NEXT commands replace the SKIP_FORWARD command.

## Standalone Music Control (SATP and Keypad)

These commands provide the ability to control music playback without the Kaleidescape onscreen display. These commands are used for two control paradigms: text-based music browsing interface

(SATP) and keypads. SATP (Standalone Touch Panel Control) is used for controlling music with a graphical interface, for example, a touch panel or personal computer. Keypad control is used where there is no graphical display available. All the messages in this section can be sent to and from a music zone. Commands are grouped with a detailed description of each command. Table 11 lists control commands for SATP and keypads.

> Note: The response code for music related commands will return “Command is not available” for products which do not support music.

Table 11: Standalone music control command summary

| Command | Description |
|---|---|
| Text-based music browsing interface (SATP) | |
| BROWSE | Used to navigate the text-based music browsing interface (SATP). |
| PERFORM_ACTION | Performs a specified action on a music handle. |

| Keypad collections and presets | |
|---|---|
| PLAY_FIRST_IN_MUSIC_COLLECTION | Plays the first item in a music collection. |
| PLAY_NEXT_IN_MUSIC_COLLECTION | Plays the next item in a music collection. |
| PLAY_PREVIOUS_IN_MUSIC_COLLECTION | Plays the previous item in a music collection. |
| ASSIGN_PLAYING_MUSIC_TO_PRESET | Assigns a preset tag to the music item currently playing. |
| PLAY_MUSIC_PRESET | Plays the music item associated with a preset tag. |
| GET_MUSIC_PRESET_INFORMATION | Provides detailed information about a music preset. |
| GET_PLAYING_MUSIC_INFORMATION | Provides information about music currently playing. |

## Text-based music browsing interface (SATP)

### Overview

Kaleidescape music zones support a text-based interface for browsing the music library and controlling music playback. This interface allows a controller with a text display to control the music zone without requiring access to the Kaleidescape onscreen display.

The interface is organized as a hierarchical tree of nodes that contain lines of text to display on the controller. Figure 8 illustrates the node hierarchy.

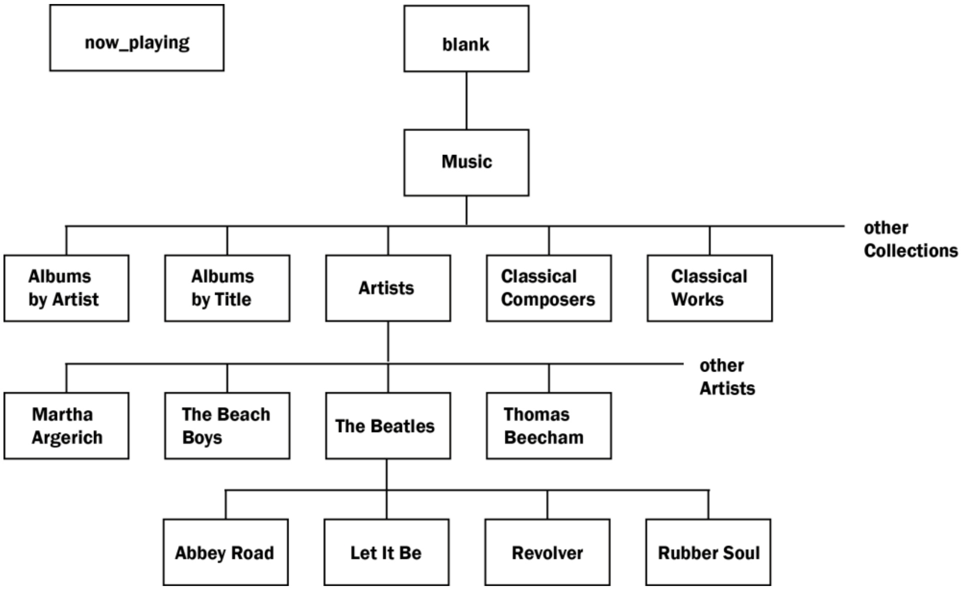
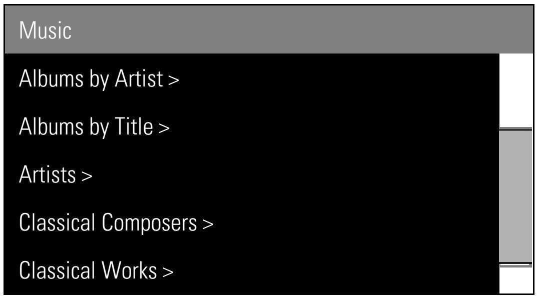


Figure 8: Node hierarchy

For example, the user can start with the top level music view, which can have the following selections.



Music

Albums by Artist >

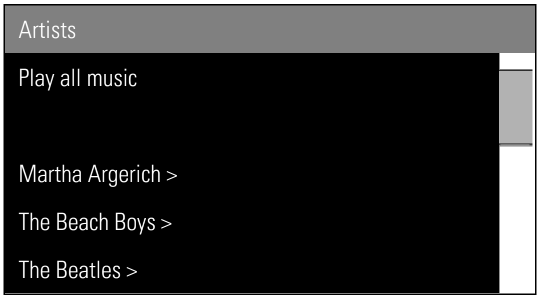
Albums by Title >

Artists >

Classical Composers >

Classical Works >

The user selects Artists, going to the next level in the hierarchy.



Artists

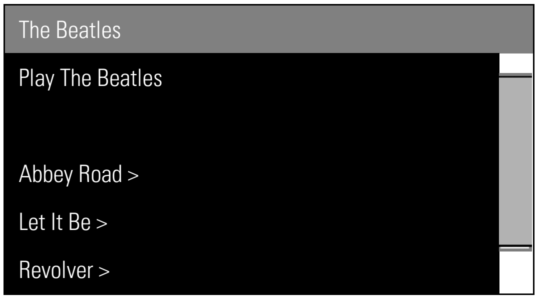
Play all music

Martha Argerich >

The Beach Boys >

The Beatles >

Then the user selects The Beatles, going down another level to display a list of albums by The Beatles.



The Beatles

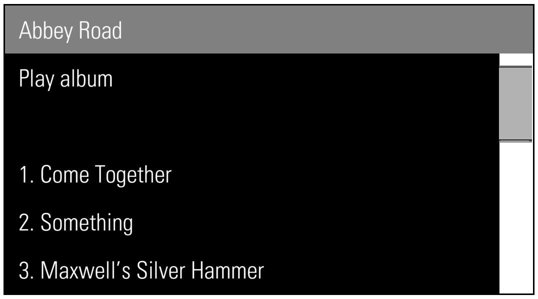
Play The Beatles

Abbey Road >

Let It Be >

Revolver >

Selecting Abbey Road displays a list of songs from the album.



Abbey Road

Play album

1. Come Together

2. Something

3. Maxwell's Silver Hammer

The user then selects Come Together to begin playback of that song. Selecting any line on this node starts playback for that song.

> Note: If an individual song is selected, only that song is added to the now playing queue. No other songs from the album are added. For example, if the user selected Come Together from the list, after playback of that song is finished, the system simply stops playback. The system does not move on to Something or any other song in the album.
> To play several songs from an album, the user must either select Play album or add songs individually to the playback queue.

Each node in the hierarchy has a unique handle that identifies it to the system. The `BROWSE` command is used to retrieve the information for a given node. Because a given node can have more lines than can be displayed on a controller, the `BROWSE` command is usually used to request a subset of the total lines for the node.

Figure 9 shows the Artists node for a particular system. Although the node has more than eleven lines, starting with Play all music and continuing past Michael Bublé, the controller can only display five lines. The controller has requested a window of five lines starting with the fifth line, The Beatles. As the user scrolls up and down, the controller will request different windows starting at different locations for the user display.

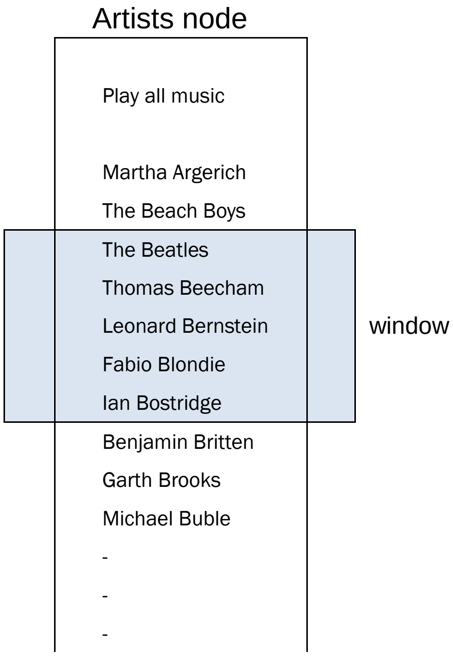


Figure 9: Example Artists node

Each line for a given node contains more information than simply the text for display. Lines also include technical information about how a controller should handle the line. For example, lines usually contain actions that can be performed on the line, information about whether or not the line is playing, etc. This interface does not support browsing the movie library.

BROWSE

| | |
|---|---|
| Affects | Any zone |
| Command | `BROWSE:browse_handle:passcode:lines:flags:` |
| Response | `status:BROWSE_RESULTS_OVERVIEW:`<br>`browse_handle:title:response_lines:total_lines:`<br>`[first_line_index:playing_line_index:]` |
| Response | `status:BROWSE_RESULT:`<br>`relative_line:absolute_line:text:play_status:`<br>`default_label:default_behavior:default_handle:default_pop:`<br>`action1_label:action1_behavior:action1_handle:action1_pop:`<br>`action2_label:action2_behavior:action2_handle:action2_pop:`<br>`action3_label:action3_behavior:action3_handle:action3_pop:`<br>`action4_label:action4_behavior:action4_handle:action4_pop:` |

The `BROWSE` command is used to load a window of information from a given node in the text-based music browsing hierarchy. The command includes the handle for the given node, along with the window of lines to be retrieved from the node. Flags can be set to filter the results for a given node, or allow the Kaleidescape System to suggest a location.

### BROWSE parameters

*`browse_handle`* is the handle of the node being requested.

Generally, this handle comes from a given line in a `BROWSE_RESULT` message. However, there are special nodes that can be reached only by explicit names.

`[blank]` Refers to the topmost node in the browsing tree. This currently has only the top level music node as its child.

`music` Refers to the top level music node. Contains all the predefined and user-defined collections in the system, such as Artists, Albums by Title, Albums by Artist, etc.

`now_playing` Refers to the top level node for the now playing queue.

The Now_Playing node is separate from the rest of the browsing tree nodes. This special node is used to view the list of what is currently queued for playback.

Note: The `music` and `now_playing` handles are text literals that must be entered exactly; `blank` should not be literal.

Note: Browse handles are usually different from the selection handles used for `GET_CONTENT_DETAILS`.

*`passcode`* is unused, leave blank.

*`lines`* specifies the window of lines to be retrieved from the indicated node. For example, `1-10` returns lines 1 through 10, inclusive; `5-10` returns lines 5 through 10, inclusive. Note that this parameter is partially ignored when using the `suggest` flag.

Note: The number of lines returned is limited to a maximum of 100.

*`flags`* are used to modify the `BROWSE` request in several ways. Multiple flags must be separated by a semi-colon. Two flags are currently supported: `filter` and `suggest`.

`filter` is used to apply a filter to the lines being returned so that only those lines that match the filter are displayed. This flag should be entered as `filter="searchstring"` where `searchstring` is the string to be filtered.

For example, a user searching for The Beatles in the Artists node might use `filter="Bea"`.

The search string supports any alphanumeric character, therefore a full keyboard is recommended to provide the best experience. For situations where a full keyboard is not practical, the `filter` tag supports a keyboard where individual keys stand for more than one letter. The user types normally, but key presses are added to the search string as multiple characters enclosed in square brackets, instead of raw characters.

For example, a user is searching for The Beatles on the Artists node using a limited alphanumeric keypad with the standard telephone letter configuration, (i.e., 2 = abc, 3=def, etc.) To enter Bea for the search, the user types 232. The controller then sends the corresponding filter tag of `filter="[abc][def][abc]"`. This tag returns a list of artists that match both the `Bea` string, as well as all other possible permutations of the three characters.

`suggest` When this flag is used, the Kaleidescape System ignores the specific window provided in the *`lines`* parameter, and returns a page of results centered on the item currently playing. The page of results has the same number of lines as the window defined in the *`lines`* parameter. This flag is useful when displaying the now playing queue, because the controller can jump directly to the currently playing item.

When the `suggest` flag is used, the `BROWSE_RESULTS_OVERVIEW` returns two additional fields indicating the *`absolute_line`* of the first item in the list, as well as the *`absolute_line`* index of the line that is currently playing.

Every `BROWSE` command is followed by a `BROWSE_RESULTS_OVERVIEW` response and several `BROWSE_RESULT` messages, if the node is not empty.

Example BROWSE bcommand

This is an example of a typical `BROWSE` command.

```
01.01/1/BROWSE:1.7.1.2.0::1-5:filter="Bo":
```

The following table describes each field in the example command.

| device_id | sequence | command name | browse handle | passcode | lines | flags |
|---|---|---|---|---|---|---|
| 01.01 | 1 | BROWSE | 1.7.1.2.0 | | 1-5 | filter="Bo" |

This command browses to the node with the handle 1.7.1.2.0. The command is requesting lines 1 through 5 of the node. The flags field of `filter="Bo"` states that the content of the node should be filtered using the string “Bo”.

### BROWSE_RESULTS_OVERVIEW

The `BROWSE_RESULTS_OVERVIEW` response provides general information about the node, including title, size, and the number of `BROWSE_RESULT` messages that follow.

*browse_handle* is the same *browse_handle* specified in the `BROWSE` command. Used to confirm that this is the correct result.

*title* is a plain text description of the node. For example, the Artists node has Artists as its title.

*response_lines* indicates the total number of lines being returned. This number is always equal to or less than the number of lines requested in the `BROWSE` command. `BROWSE_RESULT` messages equaling this number are sent subsequent to this response.

*total_lines* indicates the total number of lines that are present in this node.

The following two fields appear when the `suggest` flag is used in the `BROWSE` command. These fields do not appear otherwise.

*first_line_index* is the index of the first line, relative to the entire node. This index matches the *absolute_line* value of the first `BROWSE_RESULT` response.

*playing_line_index* is the index of the line that is being played. This index matches the *absolute_line* value of the `BROWSE_RESULT` that corresponds to the line.

#### BROWSE_RESULTS_OVERVIEW example

This is an example of a typical `BROWSE_RESULTS_OVERVIEW` message.

```
01.01/1/000:BROWSE_RESULTS_OVERVIEW:now_playing:Now
Playing:5:15:9:11:/52
```

The following tables describe each field in the example command.

| device_id | sequence | status code | message name | browse handle |
|---|---|---|---|---|
| 01.01 | 1 | 000 | BROWSE_RESULTS_OVERVIEW | now_playing |

| title | response lines | total lines | first line index | playing line index |
|---|---|---|---|---|
| Now Playing | 5 | 15 | 9 | 11 |

This message shows the overview information for the node with handle now_playing. The title that should be displayed on top of the controller interface is Now Playing. There are 15 lines in the node, and 5 lines will be returned. The first line in the window is line 9 in the node, and the index of the currently playing line is 11.

## BROWSE_RESULT

Each `BROWSE_RESULT` message represents a single line to display on the controller. The message contains all of the information required to display the message, as well as actions that can be performed.

*relative_line* is the index of the line relative to the window being requested, ranging from 1 to the *response_lines* field of the `BROWSE_RESULTS_OVERVIEW`.

Example: A controller requests lines 6-11 from a given node. The *relative_line* value of the `BROWSE_RESULT` responses goes from 1 through to 5.

*absolute_line* is the index of the line relative to the entire node. This ranges from 1 to the *total_lines* field in the `BROWSE_RESULTS_OVERVIEW` message.

Example: A controller requests lines 6-11 from a given node. The *absolute_line* value of the `BROWSE_RESULT` responses starts at 6 and counts up to 11.

*text* is the plain text displayed to the user for this line.

*play_status* indicates whether this line is playing or paused. This field is primarily used when displaying the now playing queue, to show which line is currently active.

| | |
|---|---|
| `0` | Not playing |
| `1` | Playing |
| `2` | Playing but currently paused |

### Action Tuples

Five action tuples follow the first four fields of a `BROWSE_RESULT` message. These action tuples represent different actions that can be taken on the line.

- The first action represents the default action for the line, what should be done when the line is selected or activated.
- Some lines can have other actions available, with different functions. The only action that is currently used is the play action, which can be used to tell the Kaleidescape System to play the contents of the associated line.

Each action tuple consists of 4 values.

*action_label* specifies the type of action. Can be used as a label for the available actions for a line.

| | |
|---|---|
| 0 | No specific action |
| 1 | Browse |
| 2 | Details |
| 3 | Play |
| 4 | OK |

*action_behavior* indicates how the controller executes this action.

0 No action, do nothing

1 Call BROWSE with the designated handle

2 Call GET_CONTENT_DETAILS with the designated handle

3 Call PERFORM_ACTION with the designated handle

*action_handle* is the handle associated with the action. What to do with the handle is specified by the *action_behavior* field.

*action_pop* is the number of nodes to move up the browse hierarchy after executing the action, i.e., how many times the controller should activate the back button after executing this action.

A pop value of 1 indicates that the controller should load the parent node upon completing the action. A pop value of 2 indicates that the controller should load the parent's parent, and so on.

BROWSE_RESULT example

This is an example of a typical BROWSE_RESULT message.

```
01.01/1/000:BROWSE_RESULT:3:3:Martha Argerich:0:1:1:1.7.1.2.2.Q_9457.0:
0:3:3:1.Q_9457;1=1:0:::::::::::::/86
```

The following tables describe each field in this example command.

| device_id | sequence | status code | message name | relative line | absolute line | text | play status |
|---|---|---|---|---|---|---|---|
| 01.01 | 1 | 000 | BROWSE_RESULT | 3 | 3 | Martha Argerich | 0 |

| default action | action 1 | action 2 | action 3 | action 4 |
|---|---|---|---|---|
| 1:1:1.7.1.2.2.Q_9457.0:0 | 3:3:1.Q_9457;1=1:0 | | | |

This example BROWSE_RESULT message states that this is the third line sent in the requested window, and the third line in the node overall. The text to display to the user is Martha Argerich, and the line is not playing.

The default action is 1:1:1.7.1.2.2.Q_9457.0:0, which can be expanded to the following sections.

| label | behavior | handle | pop |
|---|---|---|---|
| 1 | 1 | 1.7.1.2.2.Q_9457.0 | 0 |

Which means that the label for this action is Browse and that to execute this action, the controller should send a BROWSE command with the handle 1.7.1.2.2.Q_9457.0. Once the command is executed, no popping is required.

The first action is 3:3:1.Q_9457;1=1:0, which can be expanded to the following sections.

| label | behavior | handle | pop |
|---|---|---|---|
| 3 | 3 | 1.Q_9457;1=1 | 0 |

The label for this action is Play and that to execute this action, the controller should send a PERFORM_ACTION command with the handle 1.Q_9457;1=1.

Once the command is executed, no popping is required.

There are no other actions associated with this line.

### PERFORM_ACTION

| | |
|---|---|
| Affects | Any zone |
| Command | PERFORM_ACTION:*handle*:*passcode*:*action*: |
| Response | *status*:ACTION_PERFORMED:*text*: |

This command performs a specified action on the handle. Only music handles (albums and tracks), not movies, are supported by this command.

For the PERFORM_ACTION command:

*handle* is a unique identifier specifying the content to receive action. Handles can be content handles such as those provided by HIGHLIGHTED_SELECTION, or handles that combine action with a link to content, for example, handles provided by a BROWSE_RESULT message.

*passcode* must be blank.

*action* is the action to be performed, and can be blank if the handle is passed from a BROWSE_RESULT message.

For the ACTION_PERFORMED response:

*text* is a plain text description describing the action that has just occurred, suitable for display to a user. This text usually matches the text displayed in the upper right corner of the onscreen display, for example, Playing album or Song will play later.

When using the text-based music browsing protocol, handles containing an action are sometimes returned by BROWSE_RESULT messages. If the behavior field of one of these messages is 3, then the handle provided contains a combined action and content handle. In this situation, the action field of the PERFORM_ACTION command is ignored and can be left blank.

Example

Controller sends:

```
03.01/1/PERFORM_ACTION:1.96de0c01d6fd4a9e-t30c_1951;1=1:::
```

Kaleidescape System sends:

```
03.01/1/000:ACTION_PERFORMED:Playing Home:
```

In this example, an action is performed on the handle 1.96DE0C01D6FD4A9E-T30C_1951;1=1. This is an action handle returned by the BROWSE_RESULT message, so no action has to be specified. This handle causes the album Home to begin playback. The response to the PERFORM_ACTION command is the ACTION_PERFORMED message, describing the action that occurred as Playing Home.

## Implementation and examples

When using the text-based music browsing interface, the controller is expected to manage the behavior and navigation of the interface, unlike the OSD paradigm. A typical interface implements a method to switch between the music browsing hierarchy and the now playing queue, a back button, a way to apply a filter (i.e., a search) to a given node, the ability to scroll up and down in a given node, and an indicator showing where the controller is in a given node.

The control panel in the browser interface has an excellent example of this interface. See Figure 10.

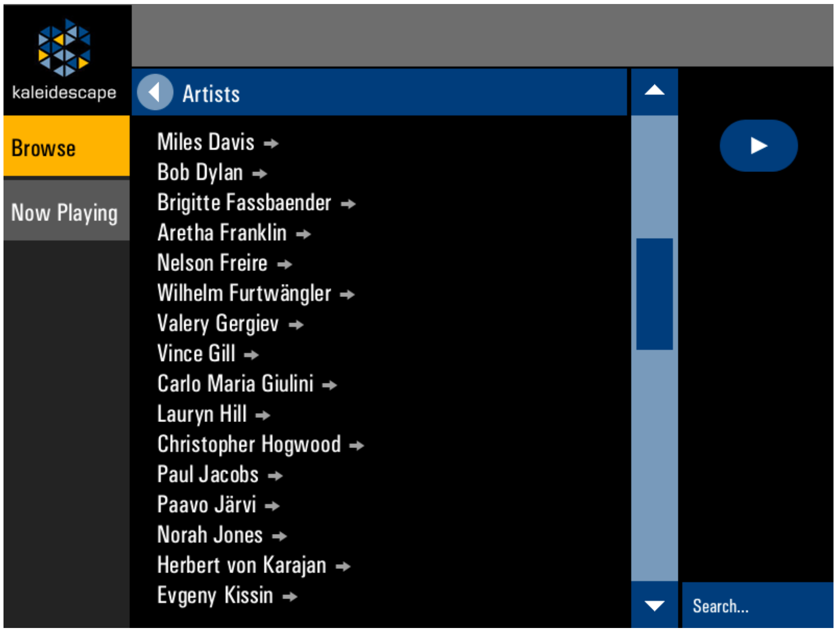


Figure 10: Browser interface control panel

The following examples assume that a controller has been designed to support these features, and describe the actual text sent back and forth. In these examples, the controller only supports 5 lines.

### Music browsing fundamentals

Browsing the music hierarchy usually starts with the top level music node with handle `music`.

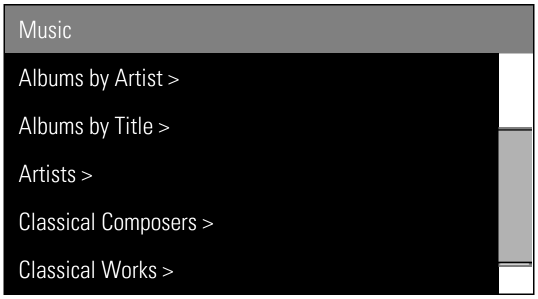
Controller sends:

```
01.01/1/BROWSE:music::1-5::
```

Kaleidescape System sends:

```
01.01/1/000:BROWSE_RESULTS_OVERVIEW:music:Music:5:8:/38
01.01/1/000:BROWSE_RESULT:1:1:Albums by Artist:
0:1:1:1.7.1.1.0:0:3:3:1.0-a_690008;1=1:0:::::::::::::/33
01.01/1/000:BROWSE_RESULT:2:2:Albums by Title:
0:1:1:1.7.1.d.0:0:3:3:1.0-a_690008;1=1:0:::::::::::::/69
01.01/1/000:BROWSE_RESULT:3:3:Artists:
0:1:1:1.7.1.2.0:0:3:3:1.0-a_690008;1=1:0:::::::::::::/58
01.01/1/000:BROWSE_RESULT:4:4:Classical Composers:
0:1:1:1.7.1.c.0:0:3:3:1.D_12117;1=1:0:::::::::::::/80
01.01/1/000:BROWSE_RESULT:5:5:Classical Works:
0:1:1:1.7.1.b.0:0:3:3:1.D_12117;1=1:0:::::::::::::/60
```

After the controller has processed the response, the following display might be shown to the user.



The `BROWSE_RESULTS_OVERVIEW` response indicates that these items are the results for the music node, as requested, and that the title that should be displayed to the user is Music. The response also shows that there are 8 lines in this node, and that there are five `BROWSE_RESULT` responses following this message.

The controller uses each `BROWSE_RESULT` response to determine what to put on the controller's display. For example, this is the `BROWSE_RESULT` message that describes the Artists line.

```
01.01/1/000:BROWSE_RESULT:3:3:Artists:0:1:1:1.7.1.2.0:0:3:3:
1.0-a_690008;1=1:0:::::::::::::/58
```

The text field of the line indicates that `Artists` be placed on the screen, and that the line is not currently playing. When the user selects this line, the default action tuple describes what the controller should do.

| label | behavior | handle | pop |
|---|---|---|---|
| 1 | 1 | 1.7.1.1.0 | 0 |

A behavior `1` indicates that the controller should call the `BROWSE` command with the handle `1.7.1.2.0` to execute this action.

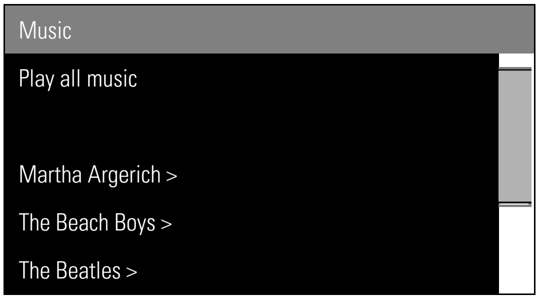
Controller sends:

```
01.01/1/BROWSE:1.7.1.2.0::1-5::
```

Kaleidescape System sends:

```
01.01/1/000:BROWSE_RESULTS_OVERVIEW:1.7.1.2.0:Artists:5:65:/12
01.01/1/000:BROWSE_RESULT:1:1:Play all music:0:3:3:
1.0-a_690008;1=1:0:::::::::::::::::/55
01.01/1/000:BROWSE_RESULT:2:2::0:::::::::::::::::::::/70
01.01/1/000:BROWSE_RESULT:3:3:Martha Argerich:0:1:1:1.7.1.2.2.Q_9457.0:
0:3:3:1.Q_9457;1=1:0:::::::::::::/86
01.01/1/000:BROWSE_RESULT:4:4:The Beach Boys:0:1:1:1.7.1.2.2.P_3640.0:
0:3:3:1.P_3640;1=1:0:::::::::::::/53
01.01/1/000:BROWSE_RESULT:5:5:The Beatles:0:1:1:1.7.1.2.2.P_3644.0:
0:3:3:1.P_3644;1=1:0:::::::::::::/55
```

After processing, the display is.



Note that this display is slightly different from the previous display, not all the lines have angle brackets. This change is based on a decision made by the programmer to display angle brackets next to lines that lead to further nodes.

The Play all music line does not contain an angle bracket. This is because this line default action does not lead to a new node. The tuple is described in this following table.

| label | behavior | handle | pop |
|---|---|---|---|
| 3 | 3 | 1.0-a_690008;1=1 | 0 |

A behavior `3` means that the controller should send the handle to the music zone using the `PERFORM_ACTION` command to activate this action. This does not lead to any new nodes, so this line is not displayed with a terminating angle bracket.

Meanwhile, the line with the text The Beatles does have an angle bracket. The following table shows its default action tuple.

| label | behavior | handle | pop |
|---|---|---|---|
| 1 | 1 | 1.7.1.2.2.P_3640.0 | 0 |

The behavior `1` means that to execute this action, the controller should call `BROWSE` with the indicated handle and display the results on the screen. This leads to a new node, so the controller places an angle bracket at the end of the line.

The user selects this line, and the controller sends the appropriate command to the music zone.

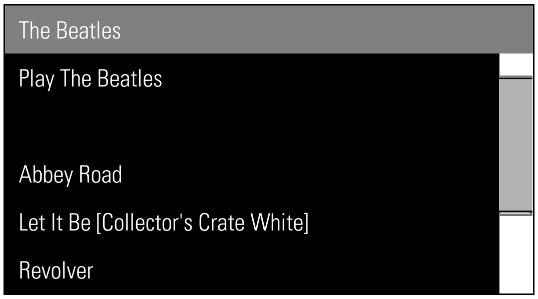
Controller sends:

```
01.01/1/BROWSE:1.7.1.2.2.P_3644.0::1-5::
```

Kaleidescape System sends:

```
01.01/1/000:BROWSE_RESULTS_OVERVIEW:1.7.1.2.2.P_3644.0:The Beatles:
5:6:/64
```

```
01.01/1/000:BROWSE_RESULT:1:1:Play The Beatles:
0:3:3:1.P_3644;1=1:0::::::::::::::::/78
01.01/1/000:BROWSE_RESULT:2:2::0::::::::::::::::::::/70
01.01/1/000:BROWSE_RESULT:3:3:Abbey Road:0:1:1:
1.7.1.2.2.P_3644.2.R_1525.0:0:3:3:1.R_1525;1=1:0::::::::::::/53
01.01/1/000:BROWSE_RESULT:4:4:Let It Be [Collector's Crate White]:
0:1:1:1.7.1.2.2.P_3644.2.R_1663460.0:0:3:3:1.R_1663460;1=1:0:::::::::::
:/54
01.01/1/000:BROWSE_RESULT:5:5:Revolver:0:1:1:1.7.1.2.2.P_3644.2.R_1518.0
:
0:3:3:1.R_1518;1=1:0::::::::::::/09
```



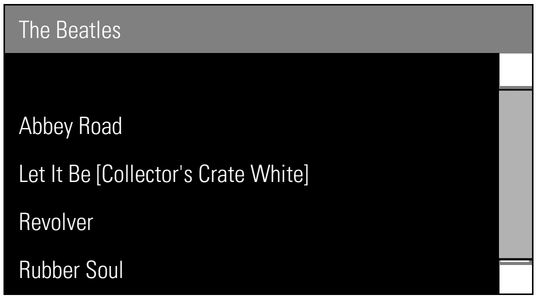
The user wants to access the album Rubber Soul, which is not visible on the screen. In this example, the user can tell that there are more items below this window by looking at the scrollbar. The user scrolls down one line.

Controller sends:

```
01.01/1/BROWSE:1.7.1.2.2.P_3644.0::2-6::
```

Kaleidescape System sends:

```
01.01/1/000:BROWSE_RESULTS_OVERVIEW:1.7.1.2.2.P_3644.0:The Beatles:
5:6:/64
01.01/1/000:BROWSE_RESULT:1:2::0::::::::::::::::::::/69
01.01/1/000:BROWSE_RESULT:2:3:Abbey Road:
0:1:1:1.7.1.2.2.P_3644.2.R_1525.0:0:3:3:1.R_1525;1=1:0::::::::::::/52
01.01/1/000:BROWSE_RESULT:3:4:Let It Be [Collector's Crate White]:
0:1:1:1.7.1.2.2.P_3644.2.R_1663460.0:0:3:3:1.R_1663460;1=1:
0::::::::::::/53
01.01/1/000:BROWSE_RESULT:4:5:Revolver:0:1:1:
1.7.1.2.2.P_3644.2.R_1518.0:0:3:3:1.R_1518;1=1:0::::::::::::/08
01.01/1/000:BROWSE_RESULT:5:6:Rubber Soul:
0:1:1:1.7.1.2.2.P_3644.2.R_1515.0:0:3:3:1.R_1515;1=1:0::::::::::::/12
```



Note that the BROWSE command in this example uses the same handle as the previous request for this node, but has updated the lines field to specify a different window to view for the node, in this case, lines 2-6.

Note that the scrollbar has moved. The scrollbar is rendered using three pieces of information.

- The number of lines returned, found in the BROWSE_RESULTS_OVERVIEW (5 for this example)
- The total number of lines in the node, also found in the BROWSE_RESULTS_OVERVIEW (6 for this example)
- The starting point for the window, found in the first BROWSE_RESULT, *absolute_line* (2 in this example)

This information is used to determine what percentage of the total view is displayed, and create an appropriate scrollbar for the results.

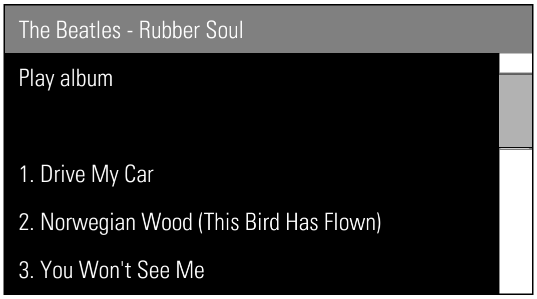
The user selects Rubber Soul.

Controller sends:

```
01.01/1/BROWSE:1.7.1.2.2.P_3644.2.R_1515.0::1-5::
```

Kaleidescape System sends:

```
01.01/1/000:BROWSE_RESULTS_OVERVIEW:1.7.1.2.2.P_3644.2.R_1515.0:
The Beatles - Rubber Soul:5:16:/06

01.01/1/000:BROWSE_RESULT:1:1:Play album:
0:3:3:1.R_1515;1=1:0:::::::::::::::::/79

01.01/1/000:BROWSE_RESULT:2:2::0:::::::::::::::::::::/70

01.01/1/000:BROWSE_RESULT:3:3:1. Drive My Car:
0:3:3:1.b9bca9a6f224fb54-t301_22;1=1:0:::::::::::::::::/28

01.01/1/000:BROWSE_RESULT:4:4:2. Norwegian Wood (This Bird Has Flown):
0:3:3:1.b9bca9a6f224fb54-t302_22;1=1:0:::::::::::::::::/69

01.01/1/000:BROWSE_RESULT:5:5:3. You Won't See Me:
0:3:3:1.b9bca9a6f224fb54-t303_22;1=1:0:::::::::::::::::/29
```



The Beatles - Rubber Soul

Play album

1. Drive My Car

2. Norwegian Wood (This Bird Has Flown)

3. You Won't See Me

Finally, the user selects Drive My Car to begin playback. This selection executes the default action for the line, so the controller examines the default action information to decide what to do.

| label | behavior | handle | pop |
|---|---|---|---|
| 3 | 3 | 1.b9bca9a6f224fb54-t301_22;1=1 | 0 |

The behavior is 3, indicating that to execute this action, the controller must send the PERFORM_ACTION command with the designated handle.

Controller sends:

```
01.01/1/PERFORM_ACTION:1.b9bca9a6f224fb54-t301_22;1=1:::
```

Kaleidescape System sends:

```
01.01/1/000:ACTION_PERFORMED:Playing Drive My Car:/67
01.01/!/000:MUSIC_TITLE:Drive My Car:The Beatles:Rubber Soul:
1.b9bca9a6f224fb54-t301_22:1.R_1515:2.202e9:/13
01.01/!/000:MUSIC_PLAY_STATUS:2:0:00150:+00000:000.00:/46
01.01/!/000:MUSIC_NOW_PLAYING_STATUS:00001:00000:1:0:0000002278:
2.202e9:/31
01.01/!/000:PLAYING_MUSIC_INFORMATION:b9bca9a6f224fb54-t301_22:
Drive My Car - The Beatles:/35
```

The music zone responds by beginning playback of the track Drive My Car which generates several event messages describing the start of playback.

### Adding music to the queue

Continuing the previous example, the user decides to play the entire Rubber Soul album instead of playing an individual song.

The user looks at the screen, which had just reloaded the Rubber Soul node after the PERFORM_ACTION command was sent.

Controller sends:

```
01.01/1/BROWSE:1.7.1.2.2.P_3644.2.R_1515.0::1-5::
```

Kaleidescape System sends:

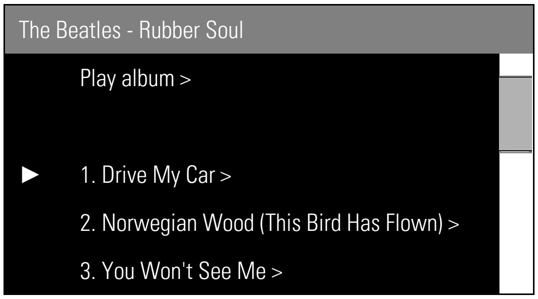
```
01.01/1/000:BROWSE_RESULTS_OVERVIEW:1.7.1.2.2.P_3644.2.R_1515.0:
The Beatles - Rubber Soul:5:16:/06
01.01/1/000:BROWSE_RESULT:1:1:Play album:
0:1:1:1.7.1.2.2.P_3644.2.R_1515.1.a.0:0::::::::::::::::/03
```

```
01.01/1/000:BROWSE_RESULT:2:2::0:::::::::::::::::::::/70
```

```
01.01/1/000:BROWSE_RESULT:3:3:1. Drive My Car:1:1:1:
1.7.1.2.2.P_3644.2.R_1515.2.b9bca9a6f224fb54-t301_22.0:
0:::::::::::::::::/38
```

```
01.01/1/000:BROWSE_RESULT:4:4:2. Norwegian Wood (This Bird Has Flown):
0:1:1:1.7.1.2.2.P_3644.2.R_1515.2.b9bca9a6f224fb54-t302_22.0:
0:::::::::::::::::/78
```

```
01.01/1/000:BROWSE_RESULT:5:5:3. You Won't See Me:0:1:1:
1.7.1.2.2.P_3644.2.R_1515.2.b9bca9a6f224fb54-t303_22.0:
0:::::::::::::::::/38
```



The Beatles - Rubber Soul

Play album >

▶ 1. Drive My Car >

2. Norwegian Wood (This Bird Has Flown) >

3. You Won't See Me >

Note that the `play_status` field for the line labeled 1. Drive My Car has been updated to show that the line is playing back. In this example, the controller puts a play icon next to the line to indicate that track is playing.

Note that all the lines have angle brackets now. This was not the case before beginning playback because the default action behavior was `3` for all the lines. Now, the default action behavior is `1`, indicating that the default action for the line is to `BROWSE` to a new node.

The reason for this change is that the music zone must now present the user with a choice when selecting music for playback. The user can decide either to add the selected item to the now playing list, or replace the now playing list with the selected item. This choice is accomplished by loading a new node with the relevant choices.

The user still wants to put the entire album in the now playing list, so the user selects Play album. The controller sends out the `BROWSE` command.

Controller sends:

```
01.01/1/BROWSE:1.7.1.2.2.P_3644.2.R_1515.1.a.0::1-5::
```

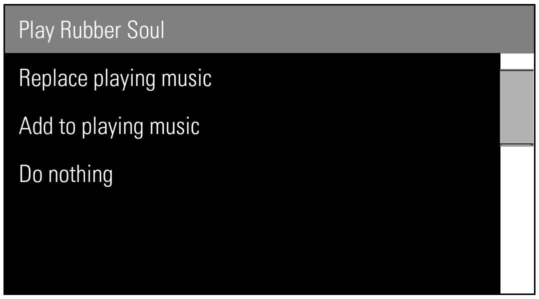
Kaleidescape System sends:

```
01.01/1/000:BROWSE_RESULTS_OVERVIEW:1.7.1.2.2.P_3644.2.R_1515.1.a.0:
Play Rubber Soul:3:3:/94
```

```
01.01/1/000:BROWSE_RESULT:1:1:Replace playing music:
0:3:3:1.R_1515;1=1:1:::::::::::::::::/78
```

```
01.01/1/000:BROWSE_RESULT:2:2:Add to playing music:
0:3:3:1.R_1515;1=3:1:::::::::::::::::/06
```

```
01.01/1/000:BROWSE_RESULT:3:3:Do nothing:0:3:3:
1.0-a_690009;1=1:1:::::::::::::::::/03
```



Play Rubber Soul

Replace playing music

Add to playing music

Do nothing

The user wants to replace the playing music and selects Replace playing music. The controller inspects the default action for this line.

| label | behavior | handle | pop |
|---|---|---|---|
| 3 | 3 | 1.R_1515;1=1 | 1 |

Another `PERFORM_ACTION` command is required. The controller sends the command.

Controller sends:

```
01.01/1/PERFORM_ACTION:1.R_1515;1=1:::
```

Kaleidescape System sends:

```
01.01/1/000:ACTION_PERFORMED:Playing Rubber Soul:/82
01.01/!/000:MUSIC_PLAY_STATUS:2:0:00150:+00000:000.00:/46
01.01/!/000:MUSIC_NOW_PLAYING_STATUS:00014:00000:1:0:0000002281:
2.202ea:/69
01.01/!/000:PLAYING_MUSIC_INFORMATION:R_1515:
The Beatles - Rubber Soul:/11
```

The controller remains on the browsing page and must reload the page. However, this action had a pop value of `1`, different than other actions that have been executed so far in this example.

A pop value of `1` means that the controller should not reload the current node, but rather, should load the parent of the current node. This is like pressing a back button once. To facilitate this action and to support a back button, Kaleidescape recommends that controllers implement a history stack to store the handles for nodes that have previously been visited. When the back button is pressed, or when a pop value is greater than `0`, simply pop the handle off the stack and re-request that handle using the `BROWSE` command. In this case, the controller re-requests the node for the Rubber Soul album.

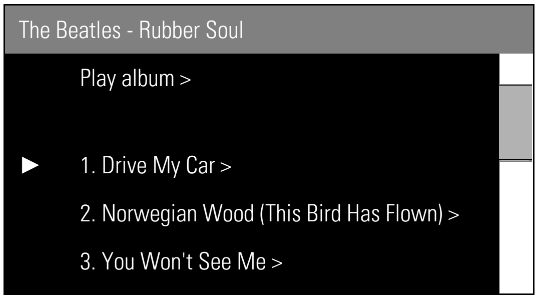
Controller sends:

```
01.01/1/BROWSE:1.7.1.2.2.P_3644.2.R_1515.0::1-5::
```

Kaleidescape System sends:

```
01.01/1/000:BROWSE_RESULTS_OVERVIEW:1.7.1.2.2.P_3644.2.R_1515.0:
The Beatles - Rubber Soul:5:16:/06
01.01/1/000:BROWSE_RESULT:1:1:Play album:
0:1:1:1.7.1.2.2.P_3644.2.R_1515.1.a.0:0:::::::::::::::/03
```

```
01.01/1/000:BROWSE_RESULT:2:2::0:::::::::::::::::::::/70
01.01/1/000:BROWSE_RESULT:3:3:1. Drive My Car:1:1:1:
1.7.1.2.2.P_3644.2.R_1515.2.b9bca9a6f224fb54-t301_22.0:
0:::::::::::::::::/38
01.01/1/000:BROWSE_RESULT:4:4:2. Norwegian Wood (This Bird Has Flown):
0:1:1:1.7.1.2.2.P_3644.2.R_1515.2.b9bca9a6f224fb54-t302_22.0:
0:::::::::::::::::/78
01.01/1/000:BROWSE_RESULT:5:5:3. You Won't See Me:0:1:1:
1.7.1.2.2.P_3644.2.R_1515.2.b9bca9a6f224fb54t303_22.0:
0:::::::::::::::::/38
```


The Beatles - Rubber Soul
Play album >
▶
1. Drive My Car >
2. Norwegian Wood (This Bird Has Flown) >
3. You Won't See Me >

Playback for the entire album has now begun.

### The Now Playing queue

Continuing the previous example, the user listens to several tracks and decides to change the currently active track. The user instructs the controller to go to the now playing place, probably by pressing a button marked Now Playing.

To fulfill this request, the controller must BROWSE to the special Now Playing node. This node uses the handle now_playing.

The BROWSE command for this would be the following command.

```
01.01/1/BROWSE:now_playing::1-5::
```

But the controller in this example is programmed to display the item currently playing to the user immediately when the controller switches to the now playing place. The controller adds a suggest flag to the BROWSE command. So the actual BROWSE command sent follows.

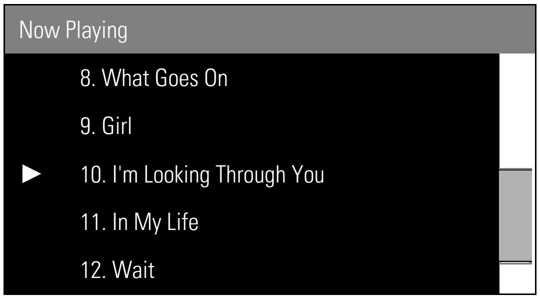
Controller sends:

```
01.01/1/BROWSE:now_playing::1-5:suggest:
```

Kaleidescape System sends:

```
01.01/1/000:BROWSE_RESULTS_OVERVIEW:now_playing:Now
Playing:5:15:9:11:/52
01.01/1/000:BROWSE_RESULT:1:9: 8. What Goes On:
0:3:3:2.202f1;1=1:0:::::::::::::::::/56
01.01/1/000:BROWSE_RESULT:2:10: 9. Girl:
0:3:3:2.202f2;1=1:0:::::::::::::::::/42
```

```
01.01/1/000:BROWSE_RESULT:3:11: 10. I'm Looking Through You:
1:3:3:2.202f3;1=1:0::::::::::::::::/82
01.01/1/000:BROWSE_RESULT:4:12: 11. In My Life:
0:3:3:2.202f4;1=1:0::::::::::::::::/20
01.01/1/000:BROWSE_RESULT:5:13: 12. Wait:
0:3:3:2.202f5;1=1:0::::::::::::::::/00
```



Now Playing

8. What Goes On<br>
9. Girl<br>
▶ 10. I'm Looking Through You<br>
11. In My Life<br>
12. Wait

Although the controller specified a window of 1-5 in the original BROWSE command, the music zone responded with lines 9 through 12. When the suggest flag is used, the Kaleidescape System ignores the requested window and replaces the window with a window centered on the currently playing item, which in this case is 10. I'm Looking Through You.

The user can now select other tracks to skip to by touching the tracks, causing an appropriate PERFORM_ACTION command to be generated. This action is identical to the examples above.

### The play action

Continuing the example, the user now decides to listen to the Beach Boys instead of the Beatles. The user switches back to the music browsing hierarchy and presses the back button several times to load previous nodes until reaching the Artists node.

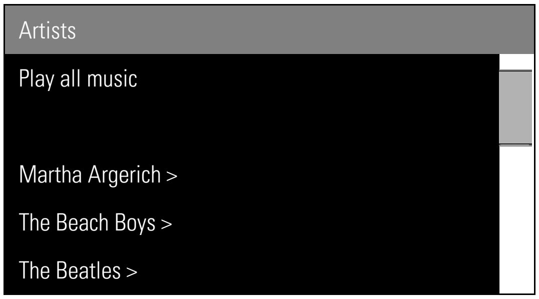
The controller switches to the appropriate history stack and begins popping handles off the stack until the user is satisfied. The following example skips that process and jumps right to the Artists node.

Controller sends:

```
01.01/1/BROWSE:1.7.1.2.0::1-5::
```

Kaleidescape System sends:

```
01.01/1/000:BROWSE_RESULTS_OVERVIEW:1.7.1.2.0:Artists:5:65:/12
01.01/1/000:BROWSE_RESULT:1:1:Play all music:
0:3:3:1.0-a_690008;1=1:0::::::::::::::::/55
01.01/1/000:BROWSE_RESULT:2:2::0::::::::::::::::::::/70
01.01/1/000:BROWSE_RESULT:3:3:Martha Argerich:
0:1:1:1.7.1.2.2.Q_9457.0:0:3:3:1.Q_9457;1=1:0::::::::::::/86
01.01/1/000:BROWSE_RESULT:4:4:The Beach Boys:
0:1:1:1.7.1.2.2.P_3640.0:0:3:3:1.P_3640;1=1:0::::::::::::/53
01.01/1/000:BROWSE_RESULT:5:5:The Beatles:
0:1:1:1.7.1.2.2.P_3644.0:0:3:3:1.P_3644;1=1:0::::::::::::/55
```



Play all music

Martha Argerich >

The Beach Boys >

The Beatles >

In this example, the controller has been programmed to support a highlight that can be moved around the screen. The controller also has a Play button programmed to activate the play action for the highlighted line when pressed.

The user wants to play The Beach Boys, but does not want to bother drilling down into the Beach Boys node to start playback. Instead, the user can highlight the Beach Boys line and press Play on the controller.

To determine the correct action, the controller looks for the line action with a play label.

| label | behavior | handle | pop |
|---|---|---|---|
| 3 | 3 | 1.P_3640;1=1 | 0 |

The label 3 indicates that this is the play action for the line. The behavior indicates that the controller should send a PERFORM_ACTION command with the handle 1.P_3640;1=1.

Controller sends:

```
01.01/1/PERFORM_ACTION:1.P_3640;1=1:::
```

Kaleidescape System sends:

```
01.01/1/000:ACTION_PERFORMED:Playing The Beach Boys:/54
01.01/!/000:MUSIC_NOW_PLAYING_STATUS:00000:00000:1:0:0000002297::/29
01.01/!/000:MUSIC_NOW_PLAYING_STATUS:00051:00000:1:1:0000002298::/37
01.01/!/000:PLAYING_MUSIC_INFORMATION:P_3640:The Beach Boys:/48
01.01/!/000:MUSIC_TITLE:Let's Go Away for While[Stereo Mix]:The Beach
Boys:Pet Sounds:1.b9bca9a6f224fb54-t314_36:1.R_485904:2.202f8:/66
01.01/!/000:MUSIC_PLAY_STATUS:2:0:00144:+00000:000.00:/49
```

If there were no play action for the line, the controller would simply have executed the default action.

| label | behavior | handle | pop |
|---|---|---|---|
| 1 | 1 | 1.7.1.2.2.P_3640.0 | 0 |

### Searching a node

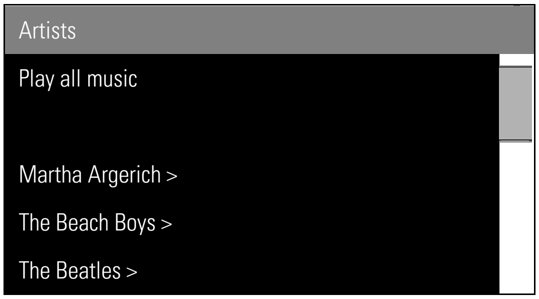
Continuing the example, after listening to The Beatles for a while, the user decides to listen to some Bob Dylan. The user returns to the Artists page.

Controller sends:

```
01.01/1/BROWSE:1.7.1.2.0::1-5::
```

Kaleidescape System sends:

```
01.01/1/000:BROWSE_RESULTS_OVERVIEW:1.7.1.2.0:Artists:5:65:/12
01.01/1/000:BROWSE_RESULT:1:1:Play all music:
0:3:3:1.0-a_690008;1=1:0:::::::::::::::::/55
01.01/1/000:BROWSE_RESULT:2:2::0:::::::::::::::::::::/70
01.01/1/000:BROWSE_RESULT:3:3:Martha Argerich:
0:1:1:1.7.1.2.2.Q_9457.0:0:3:3:1.Q_9457;1=1:0:::::::::::::/86
01.01/1/000:BROWSE_RESULT:4:4:The Beach Boys:
0:1:1:1.7.1.2.2.P_3640.0:0:3:3:1.P_3640;1=1:0:::::::::::::/53
01.01/1/000:BROWSE_RESULT:5:5:The Beatles:
0:1:1:1.7.1.2.2.P_3644.0:0:3:3:1.P_3644;1=1:0:::::::::::::/55
```



Observe that Bob Dylan does not appear on the screen. To avoid the need to scroll through the entire list looking for Bob Dylan, the user presses the search button. This action causes the controller to display a full keyboard, on which the user begins typing the world Bob, starting with the letter B.

The controller modifies the `BROWSE` command to include a filter on the Artists node. The filter just has the character `B`, so the filter is formatted as `filter="B"`.

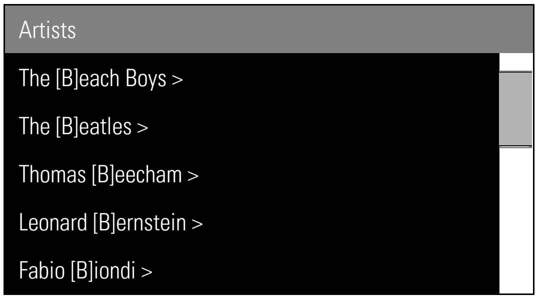
Controller sends:

```
01.01/1/BROWSE:1.7.1.2.0::1-5:filter="B":
```

Kaleidescape System sends:

```
01.01/1/000:BROWSE_RESULTS_OVERVIEW:1.7.1.2.0:Artists:5:13:/05
01.01/1/000:BROWSE_RESULT:1:1:The [B]each Boys:
0:1:1:1.7.1.2.2.P_3640.0:0:3:3:1.P_3640;1=1:0:::::::::::::/31
01.01/1/000:BROWSE_RESULT:2:2:The [B]eatles:
0:1:1:1.7.1.2.2.P_3644.0:0:3:3:1.P_3644;1=1:0:::::::::::::/33
01.01/1/000:BROWSE_RESULT:3:3:Thomas [B]eecham:
0:1:1:1.7.1.2.2.Q_8214.0:0:3:3:1.Q_8214;1=1:0:::::::::::::/37
01.01/1/000:BROWSE_RESULT:4:4:Leonard [B]ernstein:
0:1:1:1.7.1.2.2.Q_7057.0:0:3:3:1.Q_7057;1=1:0:::::::::::::/97
```

```
01.01/1/000:BROWSE_RESULT:5:5:Fabio [B]iondi:
0:1:1:1.7.1.2.2.Q_13362.0:0:3:3:1.Q_13362;1=1:0:::::::::::::/18
```



Artists

The [B]each Boys >

The [B]eatles >

Thomas [B]eecham >

Leonard [B]ernstein >

Fabio [B]iondi >

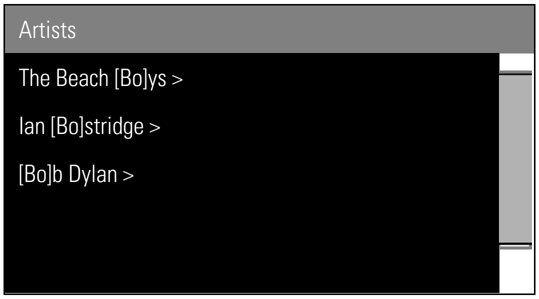
The `BROWSE_RESULTS_OVERVIEW` message identifies 13 responses, and a visual inspection shows that Bob Dylan is not visible in the window. The user types another character, o. Note that the handle remains the same.

Controller sends:

```
01.01/1/BROWSE:1.7.1.2.0::1-5:filter="Bo":
```

Kaleidescape System sends:

```
01.01/1/000:BROWSE_RESULTS_OVERVIEW:1.7.1.2.0:Artists:3:3:/54
01.01/1/000:BROWSE_RESULT:1:1:The Beach [Bo]ys:
0:1:1:1.7.1.2.2.P_3640.0:0:3:3:1.P_3640;1=1:0:::::::::::::/31
01.01/1/000:BROWSE_RESULT:2:2:Ian [Bo]stridge:
0:1:1:1.7.1.2.2.Q_14299.0:0:3:3:1.Q_14299;1=1:0:::::::::::::/65
01.01/1/000:BROWSE_RESULT:3:3:[Bo]b Dylan:
0:1:1:1.7.1.2.2.P_4147.0:0:3:3:1.P_4147;1=1:0:::::::::::::/19
```



Artists

The Beach [Bo]ys >

Ian [Bo]stridge >

[Bo]b Dylan >

Bob Dylan is now visible on the screen, and the user can select Bob Dylan to begin playing music.

Note: The letters searched for in this example appear bracketed and can appear at the beginning of any word in the artist's name.

### Searching a node with an alphanumeric keypad

The previous examples suppose that the controller provides the user with a full keyboard. Sometimes the user is limited to a small alphanumeric keypad similar to that found on a telephone.

| | | |
|---|---|---|
| 1 | 2 (ABC) | 3 (DEF) |

| | | |
|---|---|---|
| 4 (GHI) | 5 (JKL) | 6 (MNO) |
| 7 (PQRS) | 8 (TUV) | 9 (WXYZ) |

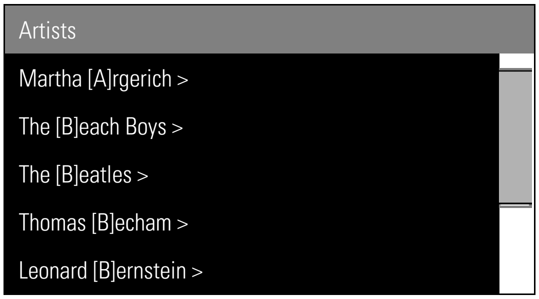
In this case, the user can begin searching for Bob Dylan by typing 2. The controller represents this 2 as `[abc]` and sends the appropriate filter to the music zone.

Controller sends:

```
01.01/1/BROWSE:1.7.1.2.0::1-5:filter="[abc]":
```

Kaleidescape System sends:

```
01.01/1/000:BROWSE_RESULTS_OVERVIEW:1.7.1.2.0:Artists:5:25:/08
01.01/1/000:BROWSE_RESULT:1:1:Martha [A]rgerich:
0:1:1:1.7.1.2.2.Q_9457.0:0:3:3:1.Q_9457;1=1:0:::::::::::::/66
01.01/1/000:BROWSE_RESULT:2:2:The [B]each Boys:
0:1:1:1.7.1.2.2.P_3640.0:0:3:3:1.P_3640;1=1:0:::::::::::::/33
01.01/1/000:BROWSE_RESULT:3:3:The [B]eatles:
0:1:1:1.7.1.2.2.P_3644.0:0:3:3:1.P_3644;1=1:0:::::::::::::/35
01.01/1/000:BROWSE_RESULT:4:4:Thomas [B]eecham:
0:1:1:1.7.1.2.2.Q_8214.0:0:3:3:1.Q_8214;1=1:0:::::::::::::/39
01.01/1/000:BROWSE_RESULT:5:5:Leonard [B]ernstein:
0:1:1:1.7.1.2.2.Q_7057.0:0:3:3:1.Q_7057;1=1:0:::::::::::::/99
```



Artists

Martha [A]rgerich >

The [B]each Boys >

The [B]eatles >

Thomas [B]echam >

Leonard [B]ernstein >

Bob Dylan is not present, and the `BROWSE_RESULTS_OVERVIEW` message shows 25 items that match this string. The user continues the search by adding a 6, representing MNO.

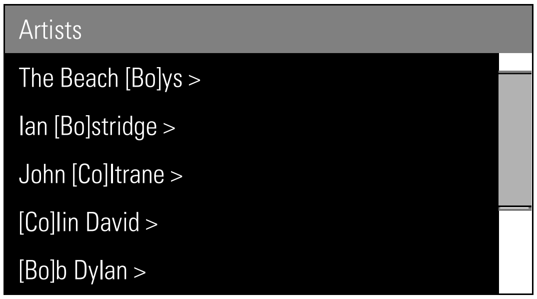
Controller sends:

```
01.01/1/BROWSE:1.7.1.2.0::1-5:filter="[abc][mno]":
```

Kaleidescape System sends:

```
01.01/1/000:BROWSE_RESULTS_OVERVIEW:1.7.1.2.0:Artists:5:5:/58
01.01/1/000:BROWSE_RESULT:1:1:The Beach [Bo]ys:
0:1:1:1.7.1.2.2.P_3640.0:0:3:3:1.P_3640;1=1:0:::::::::::::/31
01.01/1/000:BROWSE_RESULT:2:2:Ian [Bo]stridge:
0:1:1:1.7.1.2.2.Q_14299.0:0:3:3:1.Q_14299;1=1:0:::::::::::::/65
01.01/1/000:BROWSE_RESULT:3:3:John [Co]ltrane:
0:1:1:1.7.1.2.2.P_65851.0:0:3:3:1.P_65851;1=1:0:::::::::::::/77
01.01/1/000:BROWSE_RESULT:4:4:[Co]lin Davis:
0:1:1:1.7.1.2.2.Q_8172.0:0:3:3:1.Q_8172;1=1:0:::::::::::::/52
```

```
01.01/1/000:BROWSE_RESULT:5:5:[Bo]b Dylan:
0:1:1:1.7.1.2.2.P_4147.0:0:3:3:1.P_4147;1=1:0:::::::::::::/23
```



Artists

The Beach [Bo]ys >

Ian [Bo]stridge >

John [Co]ltrane >

[Co]lin David >

[Bo]b Dylan >

The user has found Bob Dylan. Notice that even though the user intends to type Bo, the music zone is returning results for Co as well.

## Keypad collections and presets

PLAY_FIRST_IN_MUSIC_COLLECTION

| | |
|---|---|
| Affects | Any zone |
| Command | `PLAY_FIRST_IN_MUSIC_COLLECTION:`*`collection`*`:` |
| Response | *`status`*`:` |

Plays the first item in a collection which always corresponds to the first `BROWSE_RESULT`, usually the topmost selectable item on the onscreen display. Because that selection is usually a music mix, the order of the songs from that collection is random.

| | |
|---|---|
| *`collection`* | is the name of the collection whose next item is to be played, for example, `Artists` or `Albums by Title`. |

This command is useful for programming keypads that only support direction and select keys (and not presets). Direction keys can be used to obtain the name of the next/previous collection, and the `PLAY_FIRST_IN_MUSIC_COLLECTION` can be used to play the contents of that collection.

Example

Controller sends:

```
01.01/0/PLAY_FIRST_IN_MUSIC_COLLECTION:Artists:
```

Kaleidescape System sends:

```
01.01/0/000:/31
03/!/000:MUSIC_NOW_PLAYING_STATUS:00000:00000:1:0:0000000022::/72
03/!/000:MUSIC_NOW_PLAYING_STATUS:00051:00000:1:1:0000000023::/80
03/!/000:PLAYING_MUSIC_INFORMATION:0-a_690008:All Music:/16
```

This example shows the `PLAY_FIRST_IN_MUSIC_COLLECTION` command used to start playback of the Artists collection. Because the first item in the collection is Play all music, this command has the effect of beginning playback of all music, as shown by the `PLAYING_MUSIC_INFORMATION` event message generated. The `MUSIC_NOW_PLAYING_STATUS` event messages confirm that music is playing back.

### PLAY_NEXT_IN_MUSIC_COLLECTION

| | |
|---|---|
| Affects | Any zone |
| Command | `PLAY_NEXT_IN_MUSIC_COLLECTION:`*`collection`*`:` |
| Response | *`status`*`:` |

Plays the item after either the currently playing or the last item played in a collection.

*`collection`* is the name of the collection whose first item is to be played, for example, `Artists` or `Albums by Title`.

This command is useful for keypads that have up and down keys. `PLAY_NEXT_IN_MUSIC_COLLECTION` can be used for the up key, and `PLAY_PREVIOUS_IN_MUSIC_COLLECTION` can be used for the down key.

> Note: To determine the music currently playing, see the `GET_PLAYING_MUSIC_INFORMATION` command.

Example

Controller sends:

```
01.01/1/PLAY_NEXT_IN_MUSIC_COLLECTION:Artists:
```

Kaleidescape System sends:

```
01.01/1/000:/32
01.01/!/000:MUSIC_NOW_PLAYING_STATUS:00051:00000:1:1:0000001732::/29
01.01/!/000:PLAYING_MUSIC_INFORMATION:P_3640:The Beach Boys:/48
01.01/!/000:MUSIC_TITLE:Pet Sounds:The Beach Boys:Pet
Sounds:1.b9bca9a6f224fb54-t30c_36:1.R_485904:2.2007c:/07
01.01/!/000:MUSIC_PLAY_STATUS:2:0:00143:+00000:000.00:/48
```

In this example, the next item in the Artists collection is activated. The `PLAYING_MUSIC_INFORMATION` event message is generated to indicate that The Beach Boys are the music item playing, and the `MUSIC_TITLE`, `MUSIC_PLAY_STATUS`, and `MUSIC_NOW_PLAYING_INFORMATION` event messages are generated with new playback information for this content.

### PLAY_PREVIOUS_IN_MUSIC_COLLECTION

| | |
|---|---|
| Affects | Any zone |
| Command | `PLAY_PREVIOUS_IN_MUSIC_COLLECTION:`*`collection`*`:` |
| Response | *`status`*`:` |

Plays the next item before either the item currently playing or last item played in the collection.

| | |
|---|---|
| *`collection`* | is the name of the collection, for example, `Albums by Artist`. |

This command is useful for keypads that have up and down keys. `PLAY_NEXT_IN_MUSIC_COLLECTION` can be used for the up key, and `PLAY_PREVIOUS_IN_MUSIC_COLLECTION` can be used for the down key.

Example

Controller sends:

```
01.01/1/PLAY_PREVIOUS_IN_MUSIC_COLLECTION:Artists:
```

Kaleidescape System sends:

```
01.01/1/000:/32
01.01/!/000:MUSIC_NOW_PLAYING_STATUS:00051:00000:1:1:0000001738::/35
01.01/!/000:PLAYING_MUSIC_INFORMATION:Q_9457:Martha Argerich:/70
01.01/!/000:MUSIC_TITLE:Piano Concerto No. 3 in C minor, Op. 37\: 2.
Largo:Ludwig van Beethoven:Piano Concerto No. 3 in C minor, Op.
37:1.b9bca9a6f224fb54-t302_6d:1.W_121665:2.200b2:/66
01.01/!/000:MUSIC_PLAY_STATUS:2:0:00621:+00000:000.00:/49
```

In this example, the previous item in the Artists collection is activated. The `PLAYING_MUSIC_INFORMATION` event message is generated identifying Martha Argerich as the music playing. The `MUSIC_NOW_PLAYING_STATUS`, `MUSIC_TITLE`, and `MUSIC_PLAY_STATUS` event messages are generated with new playback information for the new track.

### ASSIGN_PLAYING_MUSIC_TO_PRESET

| | |
|---|---|
| Affects | Any zone |
| Command | `ASSIGN_PLAYING_MUSIC_TO_PRESET:`*`tag`*`:` |
| Response | *`status`*`:` |

This command is used to store a music item into a preset by associating the currently playing music item to a preset tag that can be accessed from anywhere in the system.

The tag is an arbitrary string that a controller uses to recall the preset with the `PLAY_MUSIC_PRESET` command. The tag can indicate which button is pressed, for example, Bedroom Button 1, or the nature of the preset, for example, Bob's Favorites. These tags might be made available to the user through the onscreen display later, so name the tags appropriately.

Music items that can be assigned are artists, genres, albums, tracks, mix albums, or even all music.

Kaleidescape suggests that buttons using this command be implemented like a car stereo, i.e., when the button is held down for a period of time, the controller sends the command.

When a music preset is changed, a `MUSIC_PRESET_INFORMATION` event message is generated with the new settings.

| | |
|---|---|
| *tag* | is an identifier to use in PLAY_MUSIC_PRESET later. |

Example

Controller sends:

```
01.01/8/ASSIGN_PLAYING_MUSIC_TO_PRESET:Fav:
```

Kaleidescape System sends:

```
01/!/000:MUSIC_PRESET_INFORMATION:Fav:R_650407:Puffy AmiYumi - Nice:/12
01.01/8/000:/39
```

In this example, the ASSIGN_PLAYING_MUSIC_TO_PRESET command is used to assign the music currently playing to the preset tag Fav. Because the currently playing information is the Puffy AmiYumi album Nice, this is assigned to the preset. The MUSIC_PRESET_INFORMATION event message confirms the event change.

### PLAY_MUSIC_PRESET

| | |
|---|---|
| Affects | Any zone |
| Command | PLAY_MUSIC_PRESET:*tag*: |
| Response | *status*: |

Plays the music item associated with the preset tag.

Presets can be set to play back all music, genres, artists, albums, tracks, and mix albums using the ASSIGN_PLAYING_MUSIC_TO_PRESET command.

Music preset tags are usually associated to music items that play as music mixes.

When the PLAY_MUSIC_PRESET command is sent for a given tag, a new random element from the music item is selected for playback. Therefore, to skip an undesired song in a preset, simply send the command again.

Kaleidescape recommends this command be implemented like a car stereo, i.e., when the button is held down for a period of time, the controller sends the command.

| | |
|---|---|
| *tag* | indicates which preset to play, must correspond to the tag assigned earlier with the ASSIGN_PLAYING_MUSIC_TO_PRESET command. |

Example

Controller sends:

```
01.01/0/PLAY_MUSIC_PRESET:Fav:
```

Kaleidescape System sends:

```
01.01/0/000:/31
01/!/000:PLAYING_MUSIC_INFORMATION:R_650407:Puffy AmiYumi - Nice:/02
```

In this example, the preset Fav is recalled by the `PLAY_MUSIC_PRESET` command. This causes the information stored in the preset, specifically the album Nice by Puffy AmiYumi, to begin playback. This is seen in the `PLAYING_MUSIC_INFORMATION` event message.

### GET_MUSIC_PRESET_INFORMATION

| | |
|---|---|
| Affects | Any zone |
| Command | `GET_MUSIC_PRESET_INFORMATION:`*`tag`*`:` |
| Response | *`status`*`:MUSIC_PRESET_INFORMATION:`*`tag`*`:`<br>*`handle`*`:`*`label`*`:` |

This message provides detailed information about a preset. The event message is sent when any preset tag is reassigned to a new music item.

*tag* is a controller-assigned unique identifier for a preset. Must match the string used in the `ASSIGN_PLAYING_MUSIC_TO_PRESET` command to identify the preset.

*handle* is a short text string that identifies the associated music item. The controller can compare this handle against the handle returned by the `PLAYING_MUSIC_INFORMATION` command to determine if the preset music item is playing. A controller can use this information to identify a preset button as active.

*label* is a plain text description of the music item assigned to the preset (for example, Jazz, The Beatles, Tattoo You – The Rolling Stones, or Bach).

Label text can be provided on keypad character displays and touch panels.

Example

Controller sends:

```
01.01/2/GET_MUSIC_PRESET_INFORMATION:Fav:
```

Kaleidescape System sends:

```
01.01/2/000:MUSIC_PRESET_INFORMATION:Fav:R_650407:
Puffy AmiYumi - Nice:/72
```

In this example, information is requested for the music preset tag `Fav`. The `MUSIC_PRESET_INFORMATION` response indicates that the preset is associated with the album Nice by Puffy AmiYumi. The response also indicates that the handle is `R_650407`, which can be compared with the handle returned by the `PLAYING_MUSIC_INFORMATION` message to determine if the preset Fav is active.

GET_PLAYING_MUSIC_INFORMATION

| | |
|---|---|
| Affects | Any zone |
| Command | `GET_PLAYING_MUSIC_INFORMATION:` |
| Response | *`status`*`:PLAYING_MUSIC_INFORMATION:`*`handle`*`:`*`label`*`:` |

This message provides keypad related information about the music currently playing. This message can be used with the `MUSIC_PRESET_INFORMATION` message to identify which presets are active and which are not.

The event message is generated when the currently playing music item changes.

The music being played back can be a specific genre, artist, album, or mix album. The music item can also be an individual track or represent all music. For example, when playing back the genre Jazz, the music item currently playing is Jazz, not the specific song currently playing.

| | |
|---|---|
| *`handle`* | is a unique identifier that represents the associated music item. The controller can compare this *`handle`* against handles returned in `MUSIC_PRESET_INFORMATION` events to determine which presets are assigned to the music item currently playing. |
| *`label`* | is a plain text description of the item currently playing (for example, Jazz, The Beatles, Tattoo You – The Rolling Stones, or Bach).<br>Label text can be provided on keypad character displays and touch panels. |

Example

Controller sends:

```
01.01/5/GET_PLAYING_MUSIC_INFORMATION:
```

Kaleidescape System sends:

```
01.01/5/000:PLAYING_MUSIC_INFORMATION:R_650407:Puffy AmiYumi - Nice:/65
```

In this example, the currently playing information is returned as the album Nice by Puffy AmiYumi, which has the handle `R_650407`. The controller can match this handle against presets used by the controller to determine if any match and can be set as active.

## Advanced Integration

These commands provide additional capabilities that are useful when integrating a Kaleidescape component with lighting systems, masking systems, projectors, and other third party equipment. Table 12 lists onscreen display control commands

Table 12: Advanced integration command summary

| Command/Event | Description |
|---|---|
| Lighting, screen masking, video settings | |
| GET_MOVIE_LOCATION | Identifies the location in the movie, whether in the main content, intermission, or end credits. |
| GO_CALIBRATE_MASKING | Calibrates the top and bottom values for screen masking. |
| GO_CALIBRATE_MASKING_OVERSCAN | Defines the position of the overscan in a movie zone. |
| GET_CINEMASCAPE_MASK | Provides aspect ratio when a player is in a CinemaScape mode. |
| GET_SCREEN_MASK | Provides aspect ratio and masking information for the current video image. |
| GET_SCREEN_MASK2 | Provides masking information based on aspect ratio and overscan area. |
| SET_SCREEN_MASK | Used to inform the Kaleidescape System that a masking system is in use. |
| GET_VIDEO_MODE | Identifies the video mode currently active. |
| GET_CINEMASCAPE_MODE | Identifies the CinemaScape mode currently active. |
| SET_CINEMASCAPE_MODE | Sets the CinemaScape mode. Useful for players whose video output can be switched from a 2:35 theater to another room with a non-CinemaScape friendly display. |
| GET_SCALE_MODE | Indicates whether the image from the player requires horizontal scaling, vertical and horizontal scaling, or does not require scaling. |
| Scripts | |
| PLAY_SCRIPT | Executes one of the scripts created in the browser interface. |
| User-defined events | |
| USER_DEFINED_EVENT | Custom event message that can be set to be generated by scripts created in the browser interface, sent from another controller, or triggered by system events. |
| SEND_EVENT | Emits a user-defined event to controllers with enabled event messages. |

| Command/Event | Description |
|---|---|
| SELECT_KALEIDESCAPE_INPUT | Sent when selecting Kaleidescape input, for example, when the child user interface is activated. |
| Child mode | |
| GET_CHILD_MODE_STATE | Used to determine if the onscreen display is displaying the child user interface. |
| ENTER_CHILD_MODE | Displays the child user interface. |
| LEAVE_CHILD_MODE | Exits the child user interface and displays covers view. |

## Lighting, screen masking, and video settings

### GET_MOVIE_LOCATION

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GET_MOVIE_LOCATION:` |
| Response/Event | *status*:MOVIE_LOCATION:*location*: |

Provides information about the current location in the movie, whether playback is in the main content, intermission, or end credits. The event message is generated when this location changes.

This command is useful for triggering lighting systems and other events based on the current location within the movie. For example, during intermission, the controller could raise the lights and activate a popcorn machine.

It should be noted that the response `06` DVD/Blu-ray Disc Menu has limited functionality for Blu-Ray discs. Only a small subset of discs emits underlying information that is sufficient to report the disc menu state correctly. For Blu-ray discs unable to report the disc menu state properly the command will return `03` Main content.

*location* is a two-digit numeric code indicating the general location of movie playback.

| | |
|---|---|
| 00 | In the Kaleidescape interface or location unknown |
| 01 | unused |
| 02 | unused |
| 03 | Main content (feature, episode, bonus material, DVD/Blu-ray Disc menu) |
| 04 | Intermission (see INTERMISSION_ON) |
| 05 | End credits |
| 06 | DVD/Blu-ray Disc Menu |

Example

Controller sends:

```
01/6/GET_MOVIE_LOCATION:
```

Kaleidescape System sends:

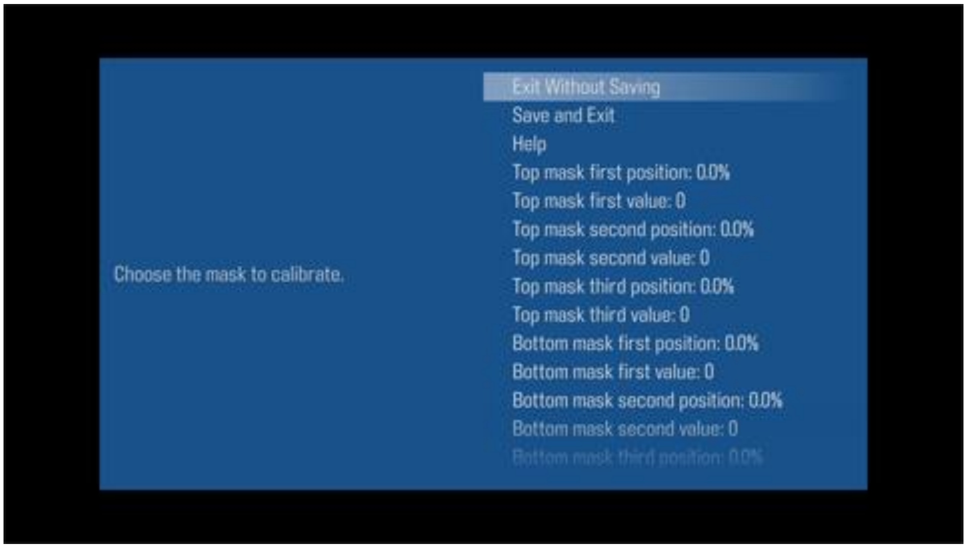
```
01/6/000:MOVIE_LOCATION:03:/89
```

This response to GET_MOVIE_LOCATION shows that movie playback is currently in the main content.

### GO_CALIBRATE_MASKING

| | |
|---|---|
| Affects | Any movie zone |
| Command | GO_CALIBRATE_MASKING: |
| Response | *status*: |

Displays the Calibrate Masking page. This page is used to calibrate the *top_calibrated* and *bottom_calibrated* values in the GET_SCREEN_MASK2 command.



Choose the mask to calibrate.

Top mask first position: 0.0%
Top mask first value: 0
Top mask second position: 0.0%
Top mask second value: 0
Top mask third position: 0.0%
Top mask third value: 0
Bottom mask first position: 0.0%
Bottom mask first value: 0
Bottom mask second position: 0.0%
Bottom mask second value: 0
Bottom mask third position: 0.0%

Figure 11: Calibrate Masking page

To calibrate masking, enter values for three positions in the range of motion of the top and bottom screen masks and assign a value to each position. For example, if the screen mask expects 0 for fully closed and 128 for fully extended, set the top mask first value to 0, the top mask second value to a midpoint of 64, and the top mask third value to 128. The zone interpolates the value of all positions in this range from these three values and returns the exact value required by the masking system without additional calculations by the controller.

To get accurate information, first calibrate the masking overscan using the `GO_CALIBRATE_MASKING_OVERSCAN` command.

> Note: This page is not accessible through the menus on the onscreen display. The only way to display this page is to use this command.

Example

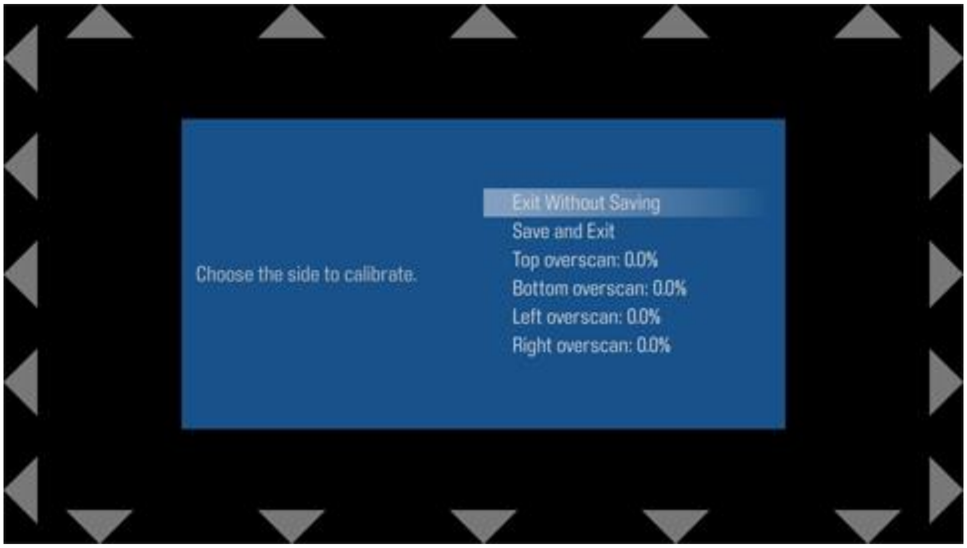
Controller sends:
`01/7/GO_CALIBRATE_MASKING:`

Kaleidescape System sends:
`01/7/000:/95`

### GO_CALIBRATE_MASKING_OVERSCAN

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GO_CALIBRATE_MASKING_OVERSCAN:` |
| Response | *`status`*`:` |

Displays the Calibrate Masking Overscan page. This page is used to define the position of the overscan extension into the Kaleidescape movie zone. This overscan data is used in calculating the values returned by the `GET_SCREEN_MASK2` command. These values are not used to adjust the position of the screen.



Choose the side to calibrate.

Top overscan: 0.0%
Bottom overscan: 0.0%
Left overscan: 0.0%
Right overscan: 0.0%

Figure 12: Calibrate Masking Overscan page

To calibrate the masking overscan, adjust the top, bottom, left, and right overscan percentage values until the associated set of arrows is directly at the edge of the viewing area. When in the correct position, the entire arrow should be visible at the edge of the screen with no blank space between the arrow and the edge of the screen.

Masking overscan must be calibrated before calibrating the masking values with `GO_CALIBRATE_MASKING` to get accurate values.

> Note: This page is not accessible through the menus on the onscreen display. The only way to display this page is to use this command.

Example

Controller sends:

```
01/3/GO_CALIBRATE_MASKING_OVERSCAN:
```

Kaleidescape System sends:

```
01/3/000:/91
```

## GET_CINEMASCAPE_MASK

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GET_CINEMASCAPE_MASK:` |
| Response/Event | *`status`*`:CINEMASCAPE_MASK:`*`frame_ratio`*`:` |

When in CinemaScape mode, provides information about the frame aspect ratio. The event message is generated whenever the CinemaScape mode is changed or when the player is set to any of the CinemaScape modes and the aspect ratio changes, e.g., starting/ending movie playback and displaying the Kaleidescape user interface.

| | |
|---|---|
| *frame_ratio* | is a zero-padded, three-digit number between 000 and 999, in hundredths of the ratio of the frame width to the frame height. The parameter can have the following values: `133`, `166`, `178`, `237`, `240`. Future revisions of kOS may support additional aspect ratios. |

> Note: This event and the command only work when in CinemaScape mode. If `GET_CINEMASCAPE_MASK` is issued when not in a CinemaScape mode, the response will be error code `028`, with the text `Incompatible video configuration`.

Example 1

Controller sends:

01/1/GET_CINEMASCAPE_MASK:

Kaleidescape System sends:

```
01/1/000:CINEMASCAPE_MASK:133:/23
```

In this example, the `CINEMASCAPE_MASK` is set for 4:3 video.

Example 2

Kaleidescape System sends:

```
01/!/000:CINEMASCAPE_MASK:178:/53
```

In this example, the `CINEMASCAPE_MASK` is set for 16:9 video. A *`frame_ratio`* value of 178 is also emitted for movies with an aspect ratio of 1.85 or 2.20.

### GET_SCREEN_MASK

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GET_SCREEN_MASK:` |
| Response/Event | *status*:SCREEN_MASK:*image_ratio*:<br>*top_trim_rel*:*bottom_trim_rel*:<br>*conservative_ratio*:*top_mask_abs*:<br>*bottom_mask_abs*: |

This message contains information about the aspect ratio and masking for the current video image. The aspect ratio information is kept simple for basic mask controller programming, while the trim/mask fields contain precise information for fine adjustments to masks. If the controller adjusts screen masks, the controller should react to this event message with a `SET_SCREEN_MASK` message.

The event message is generated whenever the aspect ratio of the video output changes.

This message provides all information needed by masking controllers of varying capabilities, some of which can be redundant depending on the masking controller.

Generally, a controller needs to use only a subset of the information, depending on the capabilities of the masking system.

- Masking controllers with presets and fine-tuning adjustments should use *`image_ratio`*, *`top_trim_rel`*, and *`bottom_trim_rel`* together.
- Masking controllers with absolute positioning should use *`top_mask_abs`* and *`bottom_mask_abs`*.
- Masking controllers with presets alone will choose based upon the type of masks.
- 16:9 or 4:3 screens with top and bottom masks should use just *`conservative_ratio`*.
- 2.35:1 screens with side masks should use just *`image_ratio`*.

Note: `GET_SCREEN_MASK` does not take overscan into account. This command provides values based on the full frame of the video image. See `GET_SCREEN_MASK2` for more information.

*image_ratio*

is the actual aspect ratio of the video content (as opposed to the full-frame content stored on the DVD). This parameter can have the following values.

| | |
|---|---|
| 00 | No image aspect ratio specified (not in playback, Kaleidescape user interface is displayed), or image aspect ratio unknown (often the case with trailers and supplemental material). The controller cannot make any assumptions about the projected video, and so should open up the masks. |
| 01 | Image aspect ratio is 1.33 (4:3) |
| 02 | Image aspect ratio is 1.66 |
| 03 | Image aspect ratio is 1.78 (16:9) |
| 04 | Image aspect ratio is 1.85 |
| 05 | Image aspect ratio is 2.35 |

*top_trim_rel*

*bottom_trim_rel*

indicate top and bottom trim values, relative to the aspect ratio specified by *image_ratio*. The value of each is a signed, zero-padded, three-digit number between -999 and +999, in tenths of a percent of the screen height.

Positive values indicate adjustment towards the center of the screen, negative toward the edge. For example, `+010` means adjust a mask inward by 1% of the screen height, and `-005` means adjust the mask outward by 0.5% of the screen height.

The value always includes a plus or minus sign.

*conservative_ratio*

has the same possible values as the *image_ratio* field, but represents a more conservative estimate of the image aspect ratio. This value never goes too far into the actual picture.

Note: This field is designed for 16:9 and 4:3 screens that have top and bottom masks without trim capability. This field should not be used on 2.35:1 screens with side masks, because it can cause the masks to move too far into the picture.

*top_mask_abs*

*bottom_mask_abs*

describe the position for the top and bottom masks in absolute terms, measured from the top and bottom of the screen, respectively. These fields are each zero-padded, four-digit numbers between 0000 and 1000, in tenths of a percent of the screen height.

For example, a *top_mask_abs* value of `0200`, means that the corresponding mask should be located 20% from the top of the screen.

Example 1 A 2.35 movie on a 16:9 screen

Kaleidescape System sends:

```
01/!/000:SCREEN_MASK:05:+000:+000:05:0121:0121:/90
```

This event message example shows that the image ratio is `05`, which represents a 2.35 aspect ratio movie. No relative trim is indicated, and the conservative image ratio is also set to `05`. The absolute positions indicate that the masks should be in 12.1% from both the top and bottom.

Example 2 A 2.35 movie on a 4:3 screen

Kaleidescape System sends:

```
01/!/000:SCREEN_MASK:05:+000:+000:05:0216:0216:/00
```

This event message is similar to the previous example, except that the absolute positions are 21.6% instead of 12.1%. This is because a 2.35 movie on a 4:3 frame has larger letterboxes. The adjustments (the relative ratios) are not affected by the frame ratio.

GET_SCREEN_MASK2

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GET_SCREEN_MASK2:` |
| Response/Event | *`status`*`:SCREEN_MASK2:`<br>*`top_mask_abs`*`:`*`bottom_mask_abs`*`:`<br>*`top_calibrated`*`:`*`bottom_calibrated`*`:` |

This message provides masking information based on aspect ratio and overscan area. This information depends on the calibration parameters entered in the Calibrate Masking Overscan and Calibrate Masking pages accessed when the `GO_CALIBRATE_MASKING_OVERSCAN` and `GO_CALIBRATE_MASKING` commands are issued.

Note: To receive meaningful data, set calibration parameters first.

*`top_mask_abs`*

*`bottom_mask_abs`* describe the position for the top and bottom masks in absolute terms, relative to the aspect ratio and accounting for overscan as set in on the Calibrate Masking Overscan page accessed when a `GO_CALIBRATE_MASKING_OVERSCAN` command is issued.

These fields are zero-padded, four-digit numbers between 0000 and 1000, in tenths of a percent of the screen height.

For example, a *`top_mask_abs`* value of `0200`, means that the corresponding mask should be located 20% from the top of the screen.

| | |
|---|---|
| *top_calibrated* | |
| *bottom_calibrated* | contain masking positions calculated by the movie zone based on values entered in the Calibrate Masking page accessed through the `GO_CALIBRATE_MASKING` command.<br>These fields return the top and bottom trim values as a zero-padded, five-digit number between 00000 and 99999. |

To calibrate the masking overscan, adjust the top, bottom, left, and right overscan percentage values until the associated set of arrows is directly at the edge of the viewing area. When in the correct position, the entire arrow is visible at the edge of the screen with no blank space between the arrow and the edge of the screen.

> Note: Masking overscan should be calibrated using the `GO_CALIBRATE_MASKING_OVERSCAN` command before calibrating the masking with the `GO_CALIBRATE_MASKING` command.

To calibrate masking, enter values for three positions in the range of motion of the top and bottom screen masks and assign a value to each position. For example, if the screen mask expects 0 for fully closed and 128 for fully extended, set the top mask first value to 0, the top mask second value to a midpoint of 64, and the top mask third value to 128. The zone interpolates the value of all positions in this range from these three values and returns the exact value required by the masking system without additional calculations by the controller.

Example

Controller sends:

```
01/7/GET_SCREEN_MASK2:
```

Kaleidescape System sends:

```
01/7/000:SCREEN_MASK2:0121:0123:00000:00000:/52
```

This response to the `GET_SCREEN_MASK2` command indicates that the absolute position for the top mask would be 12.1% from the top of the screen. The bottom mask should be 12.3% from the bottom. There are no calibrated masking values in this response.

### SET_SCREEN_MASK

| | |
|---|---|
| Affects | Any movie zone |
| Command | `SET_SCREEN_MASK:`*`flag`*`:` |
| Response | *status*`:SCREEN_MASK:`*image_ratio*`:`<br>*top_trim_rel*`:`*bottom_trim_rel*`:`<br>*conservative_ratio*`:`<br>*top_mask_abs*`:`*bottom_mask_abs*`:` |

This command is used to tell the movie zone whether or not a masking system is being used.

If a masking system is in use, the movie zone automatically adjusts the position of onscreen elements such as drop-down menus to appear in the unmasked portion of the display. If a masking system is not in use, these elements are displayed at the edges of the screen, possibly in letterbox areas.

*`flag`* indicates whether or not a masking system is in use.

`0` Instructs the movie zone not to compensate for masking.

`1` Instructs the movie zone to compensate for masking.

A controller should send this message when the controller starts up, or when the player restarts. The response to this command includes the current screen masking as described under `GET_SCREEN_MASK`.

Example

Controller sends:

```
01/0/SET_SCREEN_MASK:1:
```

Kaleidescape System sends:

```
01/0/000:SCREEN_MASK:05:+000:+001:05:0121:0123:/08
```

In this example, the controller is instructing the movie zone to compensate for masking system when positioning various elements of the screen. The response provides the latest screen masking information.

### GET_VIDEO_MODE

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GET_VIDEO_MODE:` |
| Response/Event | *`status`*`:VIDEO_MODE:`*`composite`*`:`*`component`*`:`*`HDMI`*`:` |

This message provides information about the video outputs for the composite, component, and HDMI video outputs. Controllers can use this information to configure external scalers, video processors, display devices, and screen masks.

The event message is generated whenever the video mode of any video output changes.

*`composite`* is the video mode of the composite and S-Video analog video outputs.

*`component`* is the video mode of the component analog video output.

*`HDMI`* is video mode of the HDMI digital video output.

These fields can have the following values:

| | |
|---|---|
| `00` | No output |
| `01` | 480i60 4:3 |
| `02` | 480i60 16:9 |

| | |
|---|---|
| 03 | 480p60 4:3 |
| 04 | 480p60 16:9 |
| 05 | 576i50 4:3 |
| 06 | 576i50 16:9 |
| 07 | 576p50 4:3 |
| 08 | 576p50 16:9 |
| 09 | 720p60 NTSC HD |
| 10 | 720p50 PAL HD |
| 11 | 1080i60 16:9 |
| 12 | 1080i50 16:9 |
| 13 | 1080p60 16:9 |
| 14 | 1080p50 16:9 |
| 15 | reserved |
| 16 | reserved |
| 17 | 1080p24 16:9 |
| 18 | reserved |
| 19 | 480i60 64:27 |
| 20 | 576i50 64:27 |
| 21 | 1080i60 64:27 |
| 22 | 1080i50 64:27 |
| 23 | 1080p60 64:27 |
| 24 | 1080p50 64:27 |
| 25 | 1080p24 64:27 |
| 26 | 1080p24 64:27 |
| 27 | 3840x 2160p24 16:9 |
| 28 | 3840x 2160p24 64:27 |
| 29 | 3840x 2160p30 16:9 |
| 30 | 3840x 2160p30 64:27 |
| 31 | 3840x 2160p60 16:9 |
| 32 | 3840x 2160p60 64:27 |
| 33 | 3840x 2160p25 16:9 |
| 34 | 3840x 2160p25 64:27 |
| 35 | 3840x 2160p50 16:9 |
| 36 | 3840x 2160p50 64:27 |
| 37 | 3840x 2160p24 16:9 |
| 38 | 3840x 2160p24 64:27 |

Example

Controller sends:

```
01/4/GET_VIDEO_MODE:
```

Kaleidescape System sends:

```
01/4/000:VIDEO_MODE:02:02:04:/83
```

The response to this `GET_VIDEO_MODE` command shows that both the composite and component video outputs are displaying 480i60 16:9. The HDMI video output is outputting 480p60 16:9.

### GET_VIDEO_COLOR

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GET_VIDEO_COLOR:` |
| Response/Event | *status*:VIDEO_COLOR:*EOTF*:*color_space*:<br>*color_depth*:*color_sampling*: |

This message provides color information about the current video output. Controllers can use this information to configure external scalers, video processors, and display devices.

The event message is generated whenever the video mode of any video output changes.

*EOTF* is the Electro-Optical Transfer Function standard of the currently playing content.

This field can have the following values:

| | |
|---|---|
| 00 | unknown |
| 01 | SDR |
| 02 | HDR |
| 03 | SMTPE ST 2084 |

*color_space* is the standard which defines the color space of the playing media.

This field can have the following values:

| | |
|---|---|
| 00 | default |
| 01 | RGB |
| 02 | BT.601 |
| 03 | BT.709 |
| 04 | BT.2020 |

*color_depth* is the number of bits used to define the color of each image pixel.

This field can have the following values:

| | |
|---|---|
| 24 | 24 bits |
| 30 | 30 bits |
| 36 | 36 bits |

*color_sampling* is the chroma color sampling standard of the currently playing media.

This field can have the following values:

| | |
|---|---|
| 00 | NONE |
| 01 | RGB |
| 02 | YCbCr 4:2:2 |
| 03 | YCbCr 4:4:4 |
| 04 | YCbCr 4:2:0 |

Example

Controller sends:

```
01/4/GET_VIDEO_COLOR:
```

Kaleidescape System sends:

```
01/4/000:VIDEO_COLOR:01:00:24:03:/93
```

The response to this `GET_VIDEO_COLOR` command shows that video output is Standard Dynamic Range using the default color space. It has a color depth of 24 bits and uses the color sampling standard of YCbCr 4:4:4.

GET_CONTENT_COLOR

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GET_CONTENT_COLOR:` |
| Response/Event | *status*:CONTENT_COLOR:*EOTF*:*color_space*:<br>*color_depth*:*color_sampling*: |

This message provides color details encoded in the currently playing content. Controllers can use this information to configure external scalers, video processors, and display devices.

*EOTF* is the Electro-Optical Transfer Function standard of the currently playing content. This field can have the following values:

| | |
|---|---|
| 00 | unknown |
| 01 | SDR |
| 02 | HDR |
| 03 | SMTPE ST 2084 |

*color_space* is the standard which defines the color space of the playing media. This field can have the following values:

| | |
|---|---|
| `00` | default |
| `01` | RGB |
| `02` | BT.601 |
| `03` | BT.709 |
| `04` | BT.2020 |

*color_depth* is the number of bits used to define the color of each image pixel. This field can have the following values:

| | |
|---|---|
| `24` | 24 bits |
| `30` | 30 bits |
| `36` | 36 bits |

*color_sampling* is the chroma color sampling standard of the currently playing media. This field can have the following values:

| | |
|---|---|
| `00` | NONE |
| `01` | RGB |
| `02` | YCbCr 4:2:2 |
| `03` | YCbCr 4:4:4 |
| `04` | YCbCr 4:2:0 |

Example

Controller sends:

```
01/1/GET_CONTENT_COLOR:
```

Kaleidescape System sends:

```
01/1/000:CONTENT_COLOR:03:04:30:04:/94
```

The response to this `GET_CONTENT_COLOR` command shows that the content is encoded to the SMTPE ST 2084 standard with BT.2020 color space. It has a color depth of 30 bits with a chroma mode of YCbCr of 4:2:0.

### GET_CINEMASCAPE_MODE

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GET_CINEMASCAPE_MODE:` |
| Response/Event | *`status`*`:CINEMASCAPE_MODE:`*`mode`*`:` |

The command provides information about the CinemaScape mode of the zone. This information is useful for installations that can switch player video output from a 2:35 theater to another room with a non-CinemaScape friendly display.

*`mode`* This parameter can have the following values.

| | |
|---|---|
| `0` | Not in CinemaScape mode |
| `1` | CinemaScape 2.35 Anamorphic |
| `2` | CinemaScape 2.35 Letterbox |
| `3` | CinemaScape Native 2.35 Display |

Example

Controller sends:

```
01/1/GET_CINEMASCAPE_MODE:/4C
```

Kaleidescape System sends:

```
01/1/000:CINEMASCAPE_MODE:0:/59
```

### SET_CINEMASCAPE_MODE

| | |
|---|---|
| Affects | Any movie zone |
| Command | `SET_CINEMASCAPE_MODE:`*`mode`*`:` |
| Response/Event | *`status`*`:CINEMASCAPE_MODE:`*`mode`*`:` |

The command sets the CinemaScape mode of the zone. This command is useful for installations that can switch player video output from a 2:35 theater to another room with a non-CinemaScape friendly display.

*`mode`* This parameter can have the following values.

| | |
|---|---|
| `0` | Not in CinemaScape mode |
| `1` | CinemaScape 2.35 Anamorphic |
| `2` | CinemaScape 2.35 Letterbox |
| `3` | CinemaScape Native 2.35 Display |

Example

Controller sends:

```
01/1/SET_CINEMASCAPE_MODE:2:/2B
```

Kaleidescape System sends:

```
01/1/000:CINEMASCAPE_MODE:2:/59
```

In this example, the CinemaScape mode is set to CinemaScape 2.35 Letterbox.

### GET_SCALE_MODE

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GET_SCALE_MODE:` |
| Response/Event | *status*:SCALE_MODE:*mode*: |

This command provides information about the video image output from an M-Class player, which allows triggering of an anamorphic lens or zoom mode when using a native 2.35:1 projector when CinemaScape mode is enabled. This command returns 0 when CinemaScape mode is not enabled.

*Mode* This parameter can have the following values.

| | |
|---|---|
| 0 | No scaling required |
| 1 | Image requires anamorphic scaling |
| 2 | Reserved |
| 3 | Image requires zoom |

## Scripts

### PLAY_SCRIPT

| | |
|---|---|
| Affects | Any movie zone |
| Command | PLAY_SCRIPT:*script_name*: |
| Response | *status*: |

Begins playback of the script named by the *script_name*. Scripts are set up in the user pages of the browser interface and can be used to combine movies, scenes, cover art, and other actions into a single playback item.

*script_name* is the case-sensitive name of the script to be played.

Example

Controller sends:

```
01/4/PLAY_SCRIPT:Great Vistas:
```

Kaleidescape System sends:

```
01/4/000:/92
01/!/000:TITLE_NAME::/59
```

In this example, the script named Great Vistas is set to begin playback. This command causes various event messages to occur as the script runs through its steps.

## User-defined events

### SEND_EVENT

| | |
|---|---|
| Affects | Any zone |
| Command | `SEND_EVENT:`*`message`*`:` |
| Response | *`status`*`:` |

This command causes a `USER_DEFINED_EVENT` to be emitted to all controllers with enabled event messages from the component.

*`message`* is the string emitted in the `USER_DEFINED_EVENT`.

Example

Controller sends:

```
02/1/SEND_EVENT:my_custom_event:
```

Kaleidescape System sends:

```
02/1/000:/90
02/!/000:USER_DEFINED_EVENT:my_custom_event:/13
```

### USER_DEFINED_EVENT

| | |
|---|---|
| Affects | Any zone, any controller |
| Event | *`status`*`:USER_DEFINED_EVENT:`*`event_message`*`:` |

This special event message can be used to trigger lighting, switch input sources, and more. User-defined events can be used to facilitate communication between controllers. Often *`event_message`* is a command for the controller. The Kaleidescape System sends a `USER_DEFINED_EVENT` event message when one of five cases occurs.

1. A player executes a script step configured to send a command to the control system (scripts are created and managed in the user pages of the browser interface).

   *`event_message`* is the command string configured in a script step.

2. The child user interface is activated, whether by a button press on the Child Remote or by sending a child command (e.g., `CHILD_PLAY`, `ENTER_CHILD_MODE`). Controllers can listen for this message and select the Kaleidescape player as the active source device.

   *`event_message`* is hard coded to `SELECT_KALEIDESCAPE_INPUT`.

3. A user presses volume buttons on a Kaleidescape Remote. Kaleidescape Remotes (KREMOTE-10 and KREMOTE-20) shipped prior to July 2011 do not have the IR codes programmed for volume or mute functions. For information on adding on adding IR codes to these remotes consult www.kaleidescape.com/go/remote-volume

   *`event_message`* is hard coded to `VOLUME_DOWN_PRESS`, `VOLUME_UP_PRESS`, `VOLUME_DOWN_RELEASE`, `VOLUME_UP_RELEASE`, or `TOGGLE_MUTE`.

4. A controller issues a SEND_EVENT command.

   *event_message* is the string provided by the controller issuing the command.

5. The volume capabilities of one of the Kaleidescape Apps are enabled.

   *event_message* is hard coded to VOLUME_QUERY, VOLUME_UP, VOLUME_DOWN, TOGGLE_MUTE.

Example 1 Script step event

After invoking a user-created script named User Event, the script executed the step “Send command to control system: My User Event.”

Controller sends:

```
01/2/PLAY_SCRIPT:User Event:
```

Kaleidescape System sends:

```
01/2/000:/90
01/!/000:UI_STATE:12:00:00:0:/40
01/!/000:UI_STATE:07:00:00:0:/44
01/!/000:USER_DEFINED_EVENT:My User Event:/72
```

Example 2 SELECT_KALEIDESCAPE_INPUT

A user presses a button on the Child Remote. The onscreen display switches to the child user interface.

Kaleidescape System sends:

```
01/!/000:UI_STATE:01:00:00:0:/38
01/!/000:USER_DEFINED_EVENT:SELECT_KALEIDESCAPE_INPUT:/76
01/!/000:UI_STATE:03:00:00:0:/40
```

Example 3 Kaleidescape Remote volume control event

A user presses (and releases) the Volume Down button on the Kaleidescape Remote.

Kaleidescape System sends:

```
02/!/000:USER_DEFINED_EVENT:VOLUME_DOWN_PRESS:/51
02/!/000:USER_DEFINED_EVENT:VOLUME_DOWN_RELEASE:/67
```

Example 4 SEND_EVENT command

A controller sends the command SEND_EVENT:Bedroom Controller Started: to the player, and the player issues the USER_DEFINED_EVENT to all listening controllers.

Controller sends:

```
02/4/SEND_EVENT:Bedroom Controller Started:
```

Kaleidescape System sends:

```
02/4/000:/93
02/!/000:USER_DEFINED_EVENT:Bedroom Controller Started:/43
```

## Kaleidescape Mobile App and Kaleidescape App for iPad

Kaleidescape provides two portable apps for controlling a Kaleidescape System: The new Kaleidescape Mobile App and the Kaleidescape App for iPAD. These will be referred to as the Kaleidescape Apps unless an event or command is specific to one app or another.

To provide volume control and feedback, these Kaleidescape Apps use `USER_DEFINED_EVENTS` events to communicate with a control system. The controller responds with `SEND_EVENT` commands.

Event `USER_DEFINED_EVENT:VOLUME_QUERY:`

This event message is sent by the Kaleidescape Apps when a new zone is selected or playback on the selected zone is initiated. The controller should respond to a volume query with `SEND_EVENT` commands issuing `VOLUME_CAPABILITIES`, `VOLUME_LEVEL`, and `MUTE_ON_FB`/`MUTE_OFF_FB` messages. If two modules are controlling the same zone, and the audio path is the same for both uses of the player’s zone, then only one module should respond to the volume query.

Event `USER_DEFINED_EVENT:VOLUME_UP:`
`USER_DEFINED_EVENT:VOLUME_DOWN:`

These event messages are sent by the Kaleidescape Apps when a volume button (Volume Up or Volume Down) is pressed on the app. The event message repeats until the volume button is released.

Event `USER_DEFINED_EVENT:SET_VOLUME_LEVEL=`*`Level`*`:`

This event message is sent by the Kaleidescape Mobile App when the volume slider is adjusted upwards or downwards on the app.

| | |
|---|---|
| *`Level`* | Is the desired volume level to set, specified as a percentage. |

Event `USER_DEFINED_EVENT:TOGGLE_MUTE:`

This event message is sent by the Kaleidescape Apps when the Mute button is tapped on the app.

Command `SEND_EVENT:VOLUME_CAPABILITIES=`*`flag`*`:`

This command is used by the controller to set the volume capabilities of the Kaleidescape Apps.

*flag* Is the sum of capabilities of the audio system:

| | |
|---|---|
| 0 | No volume control or feedback |
| 1 | Volume up/down control |
| 2 | Mute Control |
| 4 | Volume level feedback |
| 8 | Mute feedback |
| 16 | Volume level set |

For example, if the control system supports adjusting the volume level up and down, supports mute, and can report feedback for the volume level and mute, flag would be set to: 1+2+4+8 = 15

Command `SEND_EVENT:VOLUME_LEVEL=`*`vol_percent`*`:`

This command is used by the controller to send the volume level to the Kaleidescape Apps. The value of *vol_percent* should be between 0 and 100.

Command

```
SEND_EVENT:MUTE_ON_FB:
SEND_EVENT:MUTE_OFF_FB:
```

These commands are used by the controller to show the active state of the Mute button on the Kaleidescape Apps when mute feedback is enabled (*flag* = 8 above).

The controller should update the feedback on the Kaleidescape Apps using VOLUME_LEVEL and MUTE_ON_FB/MUTE_OFF_FB messages whenever the audio processor indicates a change in volume level or mute status.

Example

The controller is connected directly to a player (CPDID 01). The controller uses command routing to communicate with other players (e.g., CPDID 02). The controller is listening for event messages from all players (see ENABLE_EVENTS). The controller communicates with other A/V components (e.g., the audio processors).

A user opens one of the Kaleidescape Apps to control the directly connected player (CPDID 01). The app sends a volume query about the player.

Kaleidescape System sends:

```
01/!/000:USER_DEFINED_EVENT:VOLUME_QUERY:/52
```

The controller is listening for volume queries and responds to the volume query with SEND_EVENT commands (with VOLUME_CAPABILITIES, VOLUME_LEVEL, and MUTE_ON_FB/MUTE_OFF_FB messages) to initialize volume settings. First, the controller tells the app about the volume capabilities. In this case, the controller can control volume and mute with feedback.

Controller sends:

```
01/1/SEND_EVENT:VOLUME_CAPABILITIES=15:
```

Kaleidescape System sends:

```
01/1/000:/89
```

```
01/!/000:USER_DEFINED_EVENT:VOLUME_CAPABILITIES=15:/83
```

Next, the controller provides feedback to the app about the state of the Mute button. In this case, the Mute button is off.

Controller sends:

```
01/1/SEND_EVENT:MUTE_OFF_FB:
```

Kaleidescape System sends:

```
01/1/000:/89
01/!/000:USER_DEFINED_EVENT:MUTE_OFF_FB:/39
```

Next, the controller provides feedback about the volume level. In this case, the volume level is 25%.

Controller sends:

```
01/1/SEND_EVENT:VOLUME_LEVEL=25:
```

Kaleidescape System sends:

```
01/1/000:/89
01/!/000:USER_DEFINED_EVENT:VOLUME_LEVEL=25:/86
```

Now the user starts to use the app. The user taps the Mute button. The Kaleidescape System sends an event.

Kaleidescape System sends:

```
01/!/000:USER_DEFINED_EVENT:TOGGLE_MUTE:/39
```

The controller should use the Kaleidescape event as a trigger to send commands to another device (the audio processor) to control the volume. The controller sends feedback to the app that the Mute button should be on.

Controller sends:

```
01/1/SEND_EVENT:MUTE_ON_FB:
```

Kaleidescape System sends:

```
01/1/000:/89
01/!/000:USER_DEFINED_EVENT:MUTE_ON_FB:/77
```

The app displays the active state of the Mute button indicating the zone is muted.

Next, the user selects a different zone. The app sends a volume query.

Kaleidescape System sends:

```
02/!/000:USER_DEFINED_EVENT:VOLUME_QUERY:/53
```

The controller should now initialize the volume settings in the new zone with `SEND_EVENT` commands to device ID 02 with `VOLUME_CAPABILITIES`, `VOLUME_LEVEL`, and `MUTE_ON_FB`/`MUTE_OFF_FB` messages.

## Child user interface

### GET_CHILD_MODE_STATE

| | |
|---|---|
| Affects | Any movie zone |
| Command | `GET_CHILD_MODE_STATE:` |
| Response | *`status`*`:CHILD_MODE_STATE:`*`child_mode`*`:` |

This command is used to determine if the onscreen display is displaying the child user interface.

*`child_mode`* indicates whether or not the child user interface is active.

| | |
|---|---|
| `0` | The child user interface is not active. |
| `1` | The child user interface is active. |

Example

Controller sends:

```
02/1/GET_CHILD_MODE_STATE:
```

Kaleidescape System sends:

```
02/1/000:CHILD_MODE_STATE:0:/78
```

The child user interface is not active because the *`child_mode`* parameter is `0`.

### ENTER_CHILD_MODE

| | |
|---|---|
| Affects | Any movie zone |
| Command | `ENTER_CHILD_MODE:` |
| Response | *`status`*`:` |

This command causes the onscreen display to show the child user interface. Does not stop movie playback.

Example

Controller sends:

```
02/1/ENTER_CHILD_MODE:
```

Kaleidescape System sends:

```
02/1/000:/90
02/!/000:CHILD_MODE_STATE:1:/63
02/!/000:USER_DEFINED_EVENT:SELECT_KALEIDESCAPE_INPUT:/77
```

### LEAVE_CHILD_MODE

| | |
|---|---|
| Affects | Any movie zone |
| Command | `LEAVE_CHILD_MODE:` |
| Response | *`status`*`:` |

If the child user interface is displayed, the onscreen display exits the child user interface and shows the Movie Covers view. Does not stop movie playback.

Example

Controller sends:

```
02/1/LEAVE_CHILD_MODE:
```

Kaleidescape System sends:

```
02/1/000:/90
02/!/000:CHILD_MODE_STATE:0:/62
```

## Other Commands

These commands are not described in other sections and can be useful in some installations. Table 13 lists these commands.

Table 13: Command summary for other commands

| Command | Description |
|---|---|
| GET_NETWORK_SETTINGS | Returns component network settings. |
| SET_NETWORK_SETTINGS | Changes network settings. |
| GET_SYSTEM_CAPABILITIES | Returns whether or not the system has movie/music licenses. |
| GET_TIME | Displays current date and time. |
| GET_ZONE_CAPABILITIES | Provides summary of zone features. |

### GET_NETWORK_SETTINGS

| | |
|---|---|
| Affects | All components |
| Command | `GET_NETWORK_SETTINGS:` |
| Response | *status*:NETWORK_SETTINGS:*static*:*ip_address*:<br>*subnet_mask*:*gateway*:*dns1*:*dns2*: |

Parameters:

| | |
|---|---|
| *static* | is the component IP address (dynamic or static).<br>`0` Component uses DHCP to obtain an IP address.<br>`1` Component has a statically-assigned IP address. |
| *ip_address* | is the IP address of the component formatted as a string with four zero-padded, three-digit numbers between 000 and 255, separated by periods. For example, the IP address 10.200.1.120, is given as the string `010.200.001.120`. |
| *subnet_mask* | is the subnet mask used by the component in the same format as the *ip_address*. |
| *gateway* | is the IP address of the default gateway used by the component, in the same format as the *ip_address*. |
| *dns1* | is the IP address of the primary DNS server currently used by the component, in the same format as the *ip_address*. |
| *dns2* | is the IP address of the secondary DNS server used by the component, in the same format as the *ip_address*. If no secondary DNS server is set, this field is ???.???.???.???. |

Example

Controller sends:

```
01/1/GET_NETWORK_SETTINGS:
```

Kaleidescape System sends:

```
01/1/000:NETWORK_SETTINGS:0:010.100.012.194:255.255.252.000:
010.100.012.001:010.100.000.092:010.100.000.018:/69
```

This example shows a setup where the component is set to DHCP and currently has an IP address of 10.100.12.194. The subnet mask is 255.255.252.000 and the default gateway is 10.100.12.1. This component is using 10.100.0.92 and 10.100.0.18 as the DNS servers.

### SET_NETWORK_SETTINGS

| | |
|---|---|
| Affects | All components |
| Command | `SET_NETWORK_SETTINGS:`*`static`*`:`*`ip_address`*`:`*`subnet`*`:`<br>*`gateway`*`:`*`dns1`*`:`*`dns2`*`:` |
| Response | *`status`*`:NETWORK_SETTINGS:`*`static`*`:`*`ip_address`*`:`<br>*`subnet_mask`*`:`*`gateway`*`:`*`dns1`*`:`*`dns2`*`:` |

This command is used to change network settings for a component. The `NETWORK_SETTINGS` response contains all the new network settings for the component. See `GET_NETWORK_SETTINGS` for more information on this command.

Blank fields in this command are ignored, allowing only partial changes to be made.

| | | |
|---|---|---|
| *`static`* | `0` | Sets component to use DHCP to obtain an IP address. When DHCP is used, all other network fields in the command are ignored. |
| | `1` | Sets component to use a static IP address. Use the value of the *`ip_address`* in this field. |

*`ip_address`* is a static IP address, as a string with four numbers between 0 and 255, separated by periods. No need to use zero-padding for the numbers. Leave this field blank if *`static`* is `0`, or to keep the static IP address at its current setting.

*`subnet_mask`* is the subnet mask used by the component, in the same format as the *`ip_address`*. Leave this field blank if *`static`* is `0`, or to keep the subnet mask at its current setting.

*`gateway`* is the IP address of the default gateway, in the same format as the *`ip_address`*. Leave this field blank if *`static`* is `0`, or to keep the default gateway address at the current setting.

*`dns1`* is the IP address of the primary DNS server, in the same format as the *`ip_address`*. Leave this field blank if *`static`* is `0`, or to keep the primary DNS server address at the current setting.

*`dns2`* is the IP address of the secondary DNS server, in the same format as the *`ip_address`*. Leave this field blank if *`static`* is `0`, or to keep the secondary DNS server address at the current setting.

Example

Controller sends:

```
01/1/SET_NETWORK_SETTINGS:1:10.100.12.194::10.100.12.1:
10.100.0.92:10.100.0.18:
```

Kaleidescape System sends:

```
01/1/000:NETWORK_SETTINGS:1:010.100.012.194:255.255.252.000:
010.100.012.001:010.100.000.092:010.100.000.018:/70
```

In this example, the network settings are set to static IP address 10.100.12.194. The gateway is set to 10.100.12.1, and the DNS addresses are 10.100.0.92 and 10.100.0.18. Note that the subnet field has been left blank, telling the Kaleidescape System to use the prior settings.

### GET_SYSTEM_CAPABILITIES

| | |
|---|---|
| Affects | All components |
| Command | `GET_SYSTEM_CAPABILITIES:` |
| Response | *status*:SYSTEM_CAPABILITIES:*movies*:*music*:<br>:::::::: |

Provides information about the capabilities of the system.

| | | |
|---|---|---|
| *movies* | indicates whether the system supports movies. | |
| | Y | System supports movies. |
| | N | System does not support movies. |
| *music* | indicates whether the system supports music. | |
| | Y | System supports music. |
| | N | System does not support music. |

The remaining eight fields are blank in this version of the protocol, and are reserved for future use. The response to this command indicates system capabilities, and is therefore identical on all components in a system.

Example

Controller sends:

```
01/1/GET_SYSTEM_CAPABILITIES:
```

Kaleidescape System sends:

```
01/1/000:SYSTEM_CAPABILITIES:Y:Y:::::::::/59
```

This is a typical response for most systems, indicating support for both movies and music.

### GET_ZONE_CAPABILITIES

| | |
|---|---|
| Affects | Any Encore Movie Zone |
| Command | `GET_ZONE_CAPABILITIES:` |
| Response | *status*:ZONE_CAPABILITIES:*osd:movies:music:store:*<br>:::::: |

Provides information about the capabilities of the current zone.

*osd* indicates whether the zone has OSD available

`Y` Zone has OSD capabilities.

`N` Zone has no OSD capabilities.

*movies* indicates whether the zone supports movies.

`Y` Zone supports movies.

`N` Zone does not support movies.

*music* indicates whether the zone supports music.

`Y` Zone supports music.

`N` Zone does not support music.

*store* indicates whether the zone supports OSD store.

`Y` Zone supports OSD store.

`N` Zone does not support OSD store.

The remaining six fields are blank in this version of the protocol and are reserved for future use. The response to this command indicates zone capabilities and is therefore specific to each player in a system.

Example

Controller sends:

```
01/1/GET_ZONE_CAPABILITIES:
```

Kaleidescape System sends:

```
01/1/000:ZONE_CAPABILITIES:Y:Y:N:Y:::::::/59
```

This is a response for an Encore player that has OSD capabilities, supports movies, does not support music and supports the OSD store.

### GET_TIME

| | |
|---|---|
| Affects | All components |
| Command | `GET_TIME:` |
| Response | *`status`*`:TIME:`*`yyyy`*`:`*`mm`*`:`*`dd`*`:`*`hh`*`:`*`mm`*`:`*`ss`*`:`*`timezone`*`:` |

The response contains the current date and time in the following fields.

*yyyy* is a four-digit year.

*mm* is a zero-padded two digit month, starting at `01` for January.

*dd* is a zero-padded, two digit day of the month, starting at `01` for the first day.

*hh* is a 24-hour clock hour as a zero-padded, two-digit number; `00` is midnight or 12 a.m., 23 is 11 p.m.

*mm* is a zero-padded, two-digit minute from 0 to 59.

*ss* are the clock seconds as a two-digit number from 0 to 59.

*timezone* is a three-letter time zone abbreviation, including the daylight savings letter. For example, `PST` is Pacific Standard Time and `PDT` is Pacific Daylight Time. The time zone can be set in the browser interface.

Example

Controller sends:

```
01/1/GET_TIME:
```

Kaleidescape System sends:

```
01/1/000:TIME:2015:05:01:11:23:00:PDT:/93
```

This response returns the current date is May 1, 2015 and that the current time is 11:23:00 a.m. in Pacific Daylight Time.

> Note: This command replaces GET_DATE_TIME command which is still available, but GET_TIME is preferred.

# Getting Additional Support

Kaleidescape provides software for AMX, Crestron, Control4, Philips Pronto, Universal Remote Control, and ELAN control systems, including drivers, modules, touch panel templates and sample programs, as well as databases and codes for popular IR remotes. Software and documentation are available for download. See www.kaleidescape.com/support/control-systems.

If a problem occurs when programming a controller, or for additional help, contact Kaleidescape Support.

- See online resources at www.kaleidescape.com/support.
- Contact Kaleidescape Support at support@kaleidescape.com
  or +1 (650) 625-6160.

When contacting Kaleidescape Support, be prepared to provide the serial number of the Kaleidescape server. The serial number label is located behind the front panels of servers, and on the back of all components except mini players – it is on the bottom of these players.

# Appendix A: Command Summary and Status Codes

## Commands

| A | |
|---|---|
| ALPHABETIZE_COVER_ART | Arranges covers alphabetically. |
| ANGLE_NEXT | Changes to the next camera angle defined for playback. |
| ANGLE_PREVIOUS | Changes to the previous camera angle defined for playback. |

| | |
|---|---|
| Arrow commands<br>CHILD_UP<br>CHILD_UP_PRESS<br>CHILD_UP_RELEASE<br>CHILD_DOWN<br>CHILD_DOWN_PRESS<br>CHILD_DOWN_RELEASE<br>CHILD_LEFT<br>CHILD_LEFT_PRESS<br>CHILD_LEFT_RELEASE<br>CHILD_RIGHT<br>CHILD_RIGHT_PRESS<br>CHILD_RIGHT_RELEASE<br>DOWN<br>DOWN_PRESS<br>DOWN_RELEASE<br>LEFT<br>LEFT_PRESS<br>LEFT_RELEASE<br>RIGHT<br>RIGHT_PRESS<br>RIGHT_RELEASE<br>UP<br>UP_PRESS<br>UP_RELEASE | Used to navigate the onscreen display. |
| ASSIGN_PLAYING_MUSIC_TO_PRESET | Assigns a preset tag to the music item current playing. |
| AUDIO_NEXT | Changes to the next audio stream during movie playback. |
| B<br>BACKSPACE | Erases the last character entered. |
| Blu-ray color buttons<br>BLUE<br>BLUE_PRESS<br>BLUE_RELEASE<br>GREEN_PRESS<br>GREEN_RELEASE<br>GREEN<br>RED<br>RED_PRESS<br>RED_RELEASE<br>YELLOW_PRESS<br>YELLOW_RELEASE<br>YELLOW | Performs actions associated with color buttons. |
| BLURAY_POPUP_MENU_TOGGLE | Toggles display of Blu-ray Disc pop-up menu. |
| BLURAY_SPECIAL_STOP | Stops Blu-ray Disc playback. Use with caution. |

| | |
|---|---|
| BROWSE | Used to navigate the text-based music browsing interface (SATP). |
| C<br>CANCEL | Dismisses a page, dialog, or text entry. |
| CHILD_PLAY | If the child user interface is already active, plays the selected movie. If not, activates the child user interface. |
| CHILD_PAUSE | If the child user interface is already active, pauses the movie. If not, activates the child user interface. |
| CHILD_SELECT | If child user interface is already active, plays the highlighted movie. If not, activates the child user interface. |
| CHILD_SHUFFLE_COVER_ART | If the child user interface is already active, shuffles the cover art. If not, activates the child user interface. |
| CHILD_STOP | If the child user interface is already active, stops movie playback. If not, activates the child user interface. |
| D<br>DEFAULT_LEVEL | Changes the parental control level to the default level. |
| DETAILS | Toggles between the details page and the current display. |
| DISABLE_EVENTS | Disables event messages from a specified movie or music zone. |
| DISC_IN_TRAY_TOGGLE | Toggles the disc in player sheet in the OSD. |
| DISC_MENU | Displays DVD or Blu-ray Disc menu for the current playback. |
| DISC_OR_KALEIDESCAPE_MENU | Behaves like DISC_MENU during movie or music playback.<br>Behaves like KALEIDESCAPE_MENU_TOGGLE in user interface. |
| DISC_RESUME | Resumes playback from the point of interruption. |
| DISC_TOP_MENU | Displays the top menu for the DVD or Blu-ray Disc. |

| | |
|---|---|
| E<br>ENABLE_EVENTS | Enables event messages from a specified movie or music zone. |
| ENTER_CHILD_MODE | Displays the child user interface. |
| ENTER_STANDBY | Puts component into standby. |
| F<br>FILTER_LIST | Filters the list view to search criteria. |
| G<br>GET_ACTIVE_PROTOCOL | Returns the control protocol version set for the current session. |
| GET_AVAILABLE_DEVICES | Returns a list of device IDs for all system components powered on. |
| GET_AVAILABLE_DEVICES_BY_SERIAL_NUMBER | Returns a list containing the serial number device identifiers of all components in the system. |
| GET_CAMERA_ANGLE | Provides information about the current camera angle. |
| GET_CINEMASCAPE_MASK | Provides aspect ratio when a player is in a CinemaScape mode. |
| GET_CINEMASCAPE_MODE | Identifies the CinemaScape mode currently active. |
| GET_CONTENT_DETAILS | Provides information about a movie color mode and related color details. |
| GET_CONTENT_DETAILS | Provides information about a movie or album selected on the user interface. |
| GET_CHILD_MODE_STATE | Used to determine if the onscreen display is displaying the child user interface. |
| GET_CONTROLLED_ZONE | Returns the music zone currently under control. |
| GET_DEVICE_INFO | Returns component device type, serial number, device ID, and IP address. |
| GET_DEVICE_POWER_STATE | Returns power state of a component. |
| GET_DEVICE_TYPE_NAME | Returns component type. |

| | |
|---|---|
| GET_FRIENDLY_NAME | Returns name of component or music zone. |
| GET_FRIENDLY_SYSTEM_NAME | Returns the name of the System the component is part of. |
| GET_HIGHLIGHTED_SELECTION | Specifies the handle of the movie or album currently selected on the user interface. |
| GET_MOVIE_LOCATION | Identifies the location in the movie, whether in the main content, intermission, or end credits. |
| GET_MOVIE_MEDIA_TYPE | Identifies the type of media being played. |
| GET_MUSIC_NOW_PLAYING_STATUS | Sends the state of the Now Playing list. |
| GET_MUSIC_PLAY_STATUS | Provides playback information for the currently playing music. |
| GET_MUSIC_PRESET_INFORMATION | Provides detailed information about a music preset. |
| GET_MUSIC_TITLE | Provides detailed information about the music currently playing. |
| GET_NETWORK_SETTINGS | Returns component network settings. |
| GET_NUM_ZONES | Returns number of zones in a component. |
| GET_PLAY_STATUS | Identifies movie play mode, speed, location and chapter. |
| GET_PLAYING_MUSIC_INFORMATION | Provides information about music currently playing. |
| GET_PLAYING_TITLE_NAME | Provides the title of the movie currently playing. |
| GET_PROTOCOL | Returns protocol version number. |
| GET_SCALE_MODE | Indicates whether the image from the player requires horizontal scaling, vertical and horizontal scaling, or does not require scaling. |
| GET_SCREEN_MASK | Provides aspect ratio and masking information for the current video image. |
| GET_SCREEN_MASK2 | Provides masking information based on aspect ratio and overscan area. |
| GET_SYSTEM_CAPABILITIES | Returns whether or not the system has movie/music licenses. |

| | |
|---|---|
| GET_SYSTEM_READINESS_STATE | Returns the idle mode of Encore players or Cinema One (2nd generation). |
| GET_SYSTEM_VERSION | Returns protocol version number and the version of kOS. |
| GET_TIME | Displays current date and time. |
| GET_UI_STATE | Provides details about the current state of the user interface. |
| GET_USER_INPUT | Provides information about user input requested from the user interface. |
| GET_USER_INPUT_PROMPT | Returns user input prompt info and properties currently shown on screen. |
| GET_VIDEO_COLOR | Returns details of the current video output. |
| GET_VIDEO_MODE | Identifies the video mode currently active. |
| GET_ZONE_CAPABILITIES | Provides information about the current zone. |
| GO_CALIBRATE_MASKING | Calibrates the top and bottom values for screen masking. |
| GO_CALIBRATE_MASKING_OVERSCAN | Defines the position of the overscan in a movie zone. |
| GO_MOVIE_COLLECTION | Displays a specific collection in the Movie Collections view. |
| GO_MOVIE_COLLECTIONS | Displays the Movie Collections view. |
| GO_MOVIE_COVERS | Displays the Movie Covers view. |
| GO_MOVIE_LIST | Displays the Movie List view. |
| GO_MOVIE_STORE | Displays the Movie Store view. |
| GO_MOVIES | Changes the interface from a music view to a movie view. |
| GO_MUSIC | Changes the interface from a movie view to a music view. |
| GO_MUSIC_COLLECTION | Displays a specific collection in the Music Collections view. |
| GO_MUSIC_COLLECTIONS | Displays the Music Collections view. |

| | |
|---|---|
| GO_MUSIC_COVERS | Displays the Music Covers view. |
| GO_MUSIC_LIST | Displays the Music List view. |
| GO_NOW_PLAYING | Displays the Now Playing view. |
| GO_PARENTAL_CONTROL | Displays the Parental Control view. |
| GO_SCREEN_SAVER | Displays the screen saver. |
| GO_SYSTEM_STATUS | Displays the System Status view. |
| GO_VAULT_SUMMARY | Displays the Vault Summary view |
| I<br>INTERMISSION_OFF | Removes the intermission screen and resumes playback. |
| INTERMISSION_ON | Pauses playback and displays intermission screen. |
| INTERMISSION_TOGGLE | Toggles intermission screen on and off. |
| K<br>KALEIDESCAPE_MENU_OFF | Removes Kaleidescape menu. |
| KALEIDESCAPE_MENU_ON | Displays Kaleidescape menu. |
| KALEIDESCAPE_MENU_TOGGLE | Toggles Kaleidescape menu on and off. |
| KEYBOARD_CHARACTER | Sends a single character to the onscreen display. |
| KEYBOARD_LITERAL | Sends a single character to the onscreen keyboard. |
| L<br>LEAVE_CHILD_MODE | Exits the child user interface and displays covers view. |
| LEAVE_IDLE_MODE | Takes Encore players or Cinema One (2nd generation) out of idle mode. |
| LEAVE_STANDBY | Takes component out of standby. |
| M<br>MUSIC_RANDOM_OFF | Turns off random playback for music. |

| | |
|---|---|
| MUSIC_RANDOM_ON | Turns on random playback for music. |
| MUSIC_RANDOM_TOGGLE | Toggles on/off random playback for music. |
| MUSIC_REPEAT_OFF | Turns off repeat playback for music. |
| MUSIC_REPEAT_ON | Turns on repeat playback for music. |
| MUSIC_REPEAT_TOGGLE | Toggles on/off repeat playback for music. |
| N<br>NEXT and PREVIOUS | NEXT command skips forward through chapters or songs. |
| P<br>Page up/down commands<br>PAGE_DOWN<br>PAGE_DOWN_PRESS<br>PAGE_DOWN_RELEASE<br>PAGE_UP<br>PAGE_UP_PRESS<br>PAGE_UP_RELEASE | Used to navigate by pages on the onscreen display. |
| Paging and skipping<br>PAGE_DOWN_OR_NEXT<br>PAGE_DOWN_OR_NEXT_PRESS<br>PAGE_DOWN_OR_NEXT_RELEASE<br>PAGE_DOWN_OR_PREVIOUS<br>PAGE_DOWN_OR_PREVIOUS_PRESS<br>PAGE_DOWN_OR_PREVIOUS_RELEASE<br>PAGE_UP_OR_NEXT<br>PAGE_UP_OR_NEXT_PRESS<br>PAGE_UP_OR_NEXT_RELEASE<br>PAGE_UP_OR_PREVIOUS<br>PAGE_UP_OR_PREVIOUS_PRESS<br>PAGE_UP_OR_PREVIOUS_RELEASE | Behaves like NEXT or PREVIOUS during movie playback<br>Behaves like PAGE_UP or PAGE_DOWN in the user interface. |
| PAUSE | Toggles pause. |
| PERFORM_ACTION | Performs a specified action on a music handle. |
| PLAY | Begins playback of movies and music. |
| PLAY_FIRST_IN_MUSIC_COLLECTION | Plays the first item in a music collection. |
| PLAY_MUSIC_PRESET | Plays the music item associated with a preset tag. |
| PLAY_NEXT_IN_MUSIC_COLLECTION | Plays the next item in the collection. |

| | |
|---|---|
| PLAY_PREVIOUS_IN_MUSIC_COLLECTION | Plays the item previous item in the collection. |
| PLAY_SCRIPT | Executes one of the scripts created in the browser interface. |
| PLAYER_RESTART | Event message stating that a player has just been restarted. |
| POSITION_SELECT | Transmits touch screen interaction to the onscreen display. |
| PREVIOUS (see NEXT and PREVIOUS) | PREVIOUS command skips backward through chapters or songs. |
| R<br>REPLAY | Skips back five seconds during movie playback. |
| S<br>SAFE_LEVEL | Changes parental control to highest level without a passcode. |
| SCAN_FORWARD and SCAN_REVERSE | Cycles through fast-forward or fast-reverse. |
| SELECT | Selects the highlighted item in the onscreen display. |
| SELECT_KALEIDESCAPE_INPUT | Selects the Kaleidescape input. |
| SEND_EVENT | Emits a user-defined event to controllers with enabled event messages. |
| SEND_TO_SYSLOG | Posts a message to Kaleidescape System logs. |
| SET_CINEMASCAPE_MODE | Sets the CinemaScape mode. Useful when video output is distributed to displays with different aspect ratios. |
| SET_CONTROLLED_ZONE | Changes the music zone controlled by the onscreen display. |
| SET_FAVORITE_SCENE_END | Records a bookmark for the end of a scene. |
| SET_FAVORITE_SCENE_START | Records a bookmark for the start of a scene. |
| SET_FRIENDLY_NAME | Renames component or music zone. |
| SET_NETWORK_SETTINGS | Changes network settings. |
| SET_PROTOCOL_SETTINGS | Changes protocol syntax. |

| | |
|---|---|
| SET_SCREEN_MASK | Used to inform the Kaleidescape System that a masking system is in use. |
| SET_STATUS_CUE_PERIOD | Sets the frequency of PLAY_STATUS and MUSIC_PLAY_STATUS event messages. |
| SET_SUPPORTED_PROTOCOL | Sets the control protocol version to use for a session. |
| SET_USER_INPUT_ENTRY | Sends string to the current on screen user entry prompt. |
| SHOW_NAVIGATION_OVERLAY | During playback, opens the navigation option of the movie overlay to the chapter/title navigation page. |
| SHUFFLE_COVER_ART | Shuffles cover art on covers view. |
| START_CHAPTER_ENTRY | Displays a tab to enter chapter number to skip directly to. |
| START_DISC_TITLE_ENTRY | Displays a tab to enter title number to skip directly to a title. |
| START_SEND_NUMBER_TO_DISC_ENTRY | Displays a tab to enter a number key to send to a DVD or Blu-ray Disc. |
| STATUS_AND_SETTINGS | During playback, toggles the display of the navigation option of the movie overlay. Otherwise, brings up the System Status page. |
| STOP | Stops playback. |
| STOP_OR_CANCEL | Behaves like STOP during movie playback or when sent directly to a music zone.<br>Behaves like CANCEL in the user interface. |
| STOP_SCREEN_SAVER | Removes screen saver. |
| SUBTITLES_NEXT | Changes to the next subtitle track during playback. |
| U<br>USER_DEFINED_EVENT | Custom event message that can be set to be generated by scripts created in the browser interface, sent by another controller, initiated by an infrared volume command, or automatically generated based on system events. |

## Status codes

| Status Code | Name | Description |
|---|---|---|
| 000 | Success | Command was accepted and executed without error. |
| 001 | Message too long | Command is larger than the maximum frame size of 1024 characters. |
| 002 | Message contains invalid character | Character in the command is outside the supported character set from decimal ASCII value 32 (space) to decimal ASCII value 126 (tilde (~)). |
| 003 | Checksum error | Message checksum does not match calculated checksum. |
| 004 | Invalid device | Device identifier is invalid. |
| 005 | Device unavailable | There is no component available with that device identifier. Component can be turned off, disconnected, or misconfigured; or the device identifier in the message is incorrect. |
| 006 | Invalid zone syntax | Music zone identifier (the part of the device identifier after the period) was not specified correctly and must be two digits. |
| 007 | Invalid zone | Music zone identifier (the part of the device identifier after the period) is invalid. Use `GET_NUM_ZONES` to determine the range of valid identities for a given component. |
| 010 | Invalid command | Command name is unknown to the target device (possibly corrupted). |
| 011 | Invalid number of parameters | Command body contains an incorrect number of fields for the message name. |
| 012 | Invalid parameter | One or more values in the command field is invalid. |
| 013 | Device identifier conflict | More than one Kaleidescape component in the server group has the same routable (02–99) device identifier in the command. Note that a message to device ID 01 always goes to the directly connected component. |
| 014 | Invalid sequence number | Sequence number is invalid. |
| 015 | Unused | |

| Status Code | Name | Description |
|---|---|---|
| 016 | Invalid passcode | Passcode required to access specified content is invalid. |
| 017 | Invalid content handle | Content handle specified is not valid. This situation occurs if content is deleted, or if the server with the content is turned off. |
| 018 | Network error | Network connectivity issues exist between Kaleidescape components. Check network configuration. |
| 019 | Invalid serial syntax | Serial number specified for the device identifier is improperly formatted. |
| 020 | Device is in standby | Component specified by the device identifier is currently in standby and cannot respond to most control protocol requests. |
| 999 | Other error | Undetermined error occurred, or the command cannot be handled because of hardware limitations. |

# Appendix B: Revision History

## Changes in document revision 10
## (Kaleidescape software version 10.4 and 8.4 )

### New commands

The following new commands have been added:

```
SET_SUPPORTED_PROTOCOL
GET_FRIENDLY_SYSTEM_NAME
GET_USER_INPUT_PROMPT
SET_USER_INPUT_ENTRY
GET_ZONE_CAPABILITIES
```

### Other

- Updated `GET_CONTENT_DETAILS` documentation to properly set Last_played and Last_played_relative as future use as they have not been implemented yet. Also added HiRes_overURL Entry.
- Updated `GET_MOVIE_LOCATION` to include the protocol rev 16 change defining 06 as Disc Menu. Prior revisions included it in 03.
- Updated Kaleidescape App for iPad section to be inclusive of the new Kaleidescape Mobile App.

## Changes in document revision 9 (Kaleidescape software version 10.2 and 8.1)

### New commands

The following new commands have been added:

```
GET_CONTENT_COLOR
GET_FRIENDLY_SYSTEM_NAME
GET_VIDEO_COLOR
GO_MOVIE_STORE
KEYBOARD_LITERAL
```

### Other

- Updated `UI_STATE` to include additional reponse codes.
- Updated `GET_VIDEO_MODE` with new status codes.
- Changed references of Alto to be inclusive of all Encore players where appropriate.

## Changes in document revision 8 (Kaleidescape software version 6.1)

### New commands

No new commands or messages have been added.

### Other

- Updated `GET_DEVICE_TYPE_NAME`, `GET_FRIENDLY_NAME`, and `SET_FRIENDLY_NAME` for Alto.
- Clarifications to differentiate between Cinema One (1st generation) and Cinema One (2nd generation).

## Changes in document revision 7 (Kaleidescape software version 6.0)

### New commands

The following new commands have been added:

```
GET_SYSTEM_READINESS_STATE
LEAVE_IDLE_MODE
DISC_IN_TRAY_TOGGLE
```

The following new messages have been added:

```
SYSTEM_READINESS_STATE
```

GET_SCALE_MODE

### Other

- The protocol version as reported by GET_PROTOCOL is now 13.

## Changes in document revision 6 (Kaleidescape software version 4.3)

### New commands

The following new commands have been added:

GET_CINEMASCAPE_MODE
SET_CINEMASCAPE_MODE
GET_CINEMASCAPE_MASK
GET_AVAILABLE_DEVICES_BY_SERIAL_NUMBER
GET_SYSTEM_VERSION

### Other

- The protocol version as reported by GET_PROTOCOL is now 11.
- Updated GET_VIDEO_MODE with outputs for CinemaScape mode.

## Changes in document revision 5 (Kaleidescape software version 4.1)

Commands have been reorganized depending on application. A command usage section was added with brief explanations of how commands work together. All parameters are defined with command examples.

### New commands

The following new commands have been added:

SEND_TO_SYSLOG
GO_VAULT_SUMMARY
SEND_EVENT
TOGGLE_MUTE
VOLUME_UP
VOLUME_DOWN
VOLUME_UP_PRESS
VOLUME_UP_RELEASE
VOLUME_DOWN_PRESS
VOLUME_DOWN_RELEASE
GET_CHILD_MODE_STATE
ENTER_CHILD_MODE
LEAVE_CHILD_MODE
CHILD_PLAY
CHILD_STOP
CHILD_PAUSE

`SHUFFLE_COVER_ART`

`CHILD_SELECT`

`CHILD_UP`

`CHILD_UP_PRESS`

`CHILD_UP_RELEASE`

`CHILD_DOWN`

`CHILD_DOWN_PRESS`

`CHILD_DOWN_RELEASE`

`CHILD_LEFT`

`CHILD_LEFT_PRESS`

`CHILD_LEFT_RELEASE`

`CHILD_RIGHT`

`CHILD_RIGHT_PRESS`

`CHILD_RIGHT_RELEASE`

### Commands removed

The following commands have been removed:

`GET_ASPECT_RATIO`

`GO_DEMO`

`GO_DEMO_LOOP`

`GET_SQUEEZE_OF_4X3_ON_16X9`

`SET_SQUEEZE_OF_4X3_ON_16X9`

### Other

- Status codes have been added to Appendix A.
- The `BROWSE` command has been expanded with examples.

## Changes in document revision 4.6 (Kaleidescape software version 4.0)

### New commands

The following new commands have been added:

`GO_CALIBRATE_MASKING`

`GO_CALIBRATE_MASKING_OVERSCAN`

`BLURAY_SPECIAL_STOP`

`BLURAY_POPUP_MENU_TOGGLE`

`GET_MOVIE_MEDIA_TYPE`

`RED`

`BLUE`

`RED_PRESS`

`RED_RELEASE`

`BLUE_PRESS`

`BLUE_RELEASE`

`GREEN_PRESS`

`GREEN_RELEASE`

`YELLOW_PRESS`

GREEN

YELLOW

YELLOW_RELEASE

### Renamed commands

The following commands have been renamed to support both DVDs and Blu-ray Discs in the Kaleidescape library. The old commands are still supported, but are deprecated.

| New Command | Deprecated Command |
|---|---|
| DISC_MENU | DVD_MENU |
| DISC_TOP_MENU | DVD_TOP_MENU |
| DISC_RESUME | DVD_RESUME |
| START_DISC_TITLE_ENTRY | START_DVD_TITLE_ENTRY |
| START_SEND_NUMBER_TO_DISC_ENTRY | START_SEND_NUMBER_TO_DVD_ENTRY |
| DISC_OR_KALEIDESCAPE_MENU | DVD_OR_KALEIDESCAPE_MENU |

### Other

- Changed protocol version as reported by GET_PROTOCOL to 10.
- Changed terminology based on introduction of M-Class players.
- Updated GET_VIDEO_MODE with new status codes.

## Changes in document revision 4.5 (Kaleidescape software version 3.7)

### New commands

The following new commands have been added:

GET_NUM_ZONES

GET_DEVICE_TYPE_NAME

GET_DEVICE_POWER_STATE

### New status codes

The following new status code was added:

| Status Code | Name | Description |
|---|---|---|
| 020 | Device is in standby | The component specified by the device identifier is currently in standby. Component does not respond to most control protocol requests. |

### Other

- The protocol version as reported by `GET_PROTOCOL` is now 09.
- The `device_type` field in the response to `GET_DEVICE_INFO` has been deprecated in favor of using `GET_NUM_ZONES` and `GET_DEVICE_TYPE_NAME`.
- Added `filter` and `suggest` flags to `BROWSE` command.
- Added play action to `BROWSE_RESULT` responses.
- Numerous terminology changes with the introduction of the Kaleidescape Cinema One.

## Changes in document revision 4.4 (Kaleidescape software version 3.6)

### New commands

The following new commands have been added:

| | |
|---|---|
| GO_MOVIES_COLLECTION | GET_SCREEN_MASK2 |
| GO_MUSIC_COLLECTION | SET_PROTOCOL_SETTINGS |

## Changes in document revision 4.3 (Kaleidescape software version 3.5)

### New commands

The following new commands have been added:

| | |
|---|---|
| GO_MOVIES | GET_MUSIC_PRESET_INFORMATION |
| GO_MUSIC | GET_PLAYING_MUSIC_INFORMATION |

## Changes in document revision 4.2 (Kaleidescape software version 3.4)

### New commands

The following new commands have been added:

ASSIGN_PLAYING_PLAYING_MUSIC_TO_PRESET

PLAY_MUSIC_PRESET

PLAY_FIRST_IN_MUSIC_COLLECTION

PLAY_NEXT_IN_MUSIC_COLLECTION

PLAY_PREVIOUS_IN_MUSIC_COLLECTION

## Changes in document revision 4.1 (Kaleidescape software version 3.3)

### New commands

The following new commands have been added:

GET_CONTROLLED_ZONE

SET_CONTROLLED_ZONE

## Changes in document revision 4.0

### New commands

The following new commands have been added:

BROWSE

GET_CONTENT_DETAILS

PERFORM_ACTION

STOP_OR_CANCEL

GET_MUSIC_TITLE

GET_MUSIC_PLAY_STATUS

MUSIC_REPEAT_ON

PREVIOUS

NEXT

PAGE_UP_OR_NEXT

PAGE_DOWN_OR_PREVIOUS

PAGE_UP_OR_PREVIOUS

PAGE_DOWN_OR_NEXT

MUSIC_RANDOM_ON

MUSIC_REPEAT_OFF

MUSIC_REPEAT_TOGGLE

GET_USER_INPUT

GO_MUSIC_LIST

GO_MUSIC_COVERS

GO_MUSIC_COLLECTIONS

MUSIC_RANDOM_OFF

MUSIC_RANDOM_TOGGLE

GET_SYSTEM_CAPABILITIES

GET_MUSIC_NOW_PLAYING_STATUS

GO_NOW_PLAYING

### Deprecated commands

The following commands have been deprecated. Use the context-sensitive page up/down commands instead (e.g., PAGE_UP_OR_NEXT).

SKIP_FORWARD

SKIP_REVERSE

### Other

- The protocol version as reported by GET_PROTOCOL is now 05.
- The device ID can be specified as a serial number preceded by a pound sign.
- The device ID can include a zone ID to control a specific music zone.

## Changes in document revision 3.3.1

### Commands that have been replaced

The following commands have been replaced. The GET_SCREEN_MASK and GET_VIDEO_MODE commands provide more detailed information. New control programs should use these commands instead.

ASPECT_RATIO

GET_ASPECT_RATIO

### Other

- The protocol version as reported by GET_PROTOCOL is now 04.
- The instructions for configuring the control protocol device ID were corrected.

## Changes in document revision 3.3 (Kaleidescape software version 2.4)

### New commands

The following new commands have been added:

GET_FRIENDLY_NAME

GET_VIDEO_MODE

PLAY_SCRIPT

SET_FRIENDLY_NAME

USER_DEFINED_EVENT

USER_INPUT

VIDEO_MODE

### Commands that have been replaced

The following commands have been replaced. The scripts feature provides more flexibility in organizing and presenting automated sequences of favorite scenes. New control programs should use the PLAY_SCRIPT command instead.

GO_DEMO

GO_DEMO_LOOP

## Changes in document revision 3.2 (Kaleidescape software version 2.2)

### New commands

The following new command has been added:

POSITION_SELECT

### Other

- The protocol version as reported by GET_PROTOCOL is now 03.

## Changes in document revision 3.1 (Kaleidescape software version 2.1)

### New commands

The following new commands have been added:

ANGLE_NEXT

DEFAULT_LEVEL

| | |
|---|---|
| ANGLE_PREVIOUS | GET_CAMERA_ANGLE |
| START_SEND_NUMBER_TO_DVD_ENTRY | CAMERA_ANGLE |

### Other

- The second value of the ASPECT_RATIO event command now represents the aspect ratio of the video frame that the Kaleidescape Movie Player outputs, rather than that of the frame in the DVD. These values will be different if the Movie Player is performing internal adaptation between 4:3 and 16:9. The new behavior is more useful as it allows a controller to use the value directly to run an external scaler or projector.
- The GET_SQUEEZE_OF_4X3_ON_16X9 and SET_SQUEEZE_OF_4X3_ON_16X9 commands are now deprecated, as the Movie Player provides options for many more combinations of media format and video aspect ratio, controllable through the browser interface. Changing this setting has no effect.
- The SQUEEZE_OF_4X3_ON_16X9 command description has been removed. This event message is never sent by the Movie Player.

## Changes in document revision 3.0.1

### Corrections

- The PLAY_STATUS command listed incorrect values for the mode field. The values have been corrected
- The UI_STATE command listed the screen field value 06 as “Screen Saver”. This value is unused.

## Changes in document revision 3.0 (Kaleidescape software version 2.0)

### Changes

- Control Protocol version incremented

  The control protocol version, as returned by the GET_PROTOCOL command, is now 02 to reflect the added features of TCP/IP and addressable devices.
- Support for TCP/IP

  The Kaleidescape device control protocol is now supported over Ethernet using TCP/IP connections.
- Support for addressable devices

A single controller connection, whether serial or TCP/IP, can now be used to send commands to any device in a Kaleidescape System. Device identifier 01 has been defined to target the device to which a connection has been directly established; device identifiers above 01 can be assigned to individual devices.

- ISO Latin-1 character set support

  The control protocol now accepts unescaped, accented Latin-1 characters (those from decimal 128 to 255).

- New Status Codes

  The following new status codes were added:

| Status Code | Name | Description |
|---|---|---|
| 005 | Device Unavailable | There is no device available with the indicated device identifier. It can be turned off, disconnected, or configured incorrectly; or the device identifier in the command was incorrect. |
| 013 | Device Identifier Conflict | Multiple Kaleidescape components in the server group share the routable (02–99) device identifier in the command. Note that a command to device ID 01 always goes to the local device (the one with a direct connection). |
| 014 | Invalid sequence number | The sequence number is invalid. |
| 999 | Other error | An undetermined error occurred or the request could not be handled because of hardware limitations. |

## New and deprecated commands

The following new commands have been added, in some cases replacing existing commands with ambiguous responses:

| New Command | Deprecated Command |
|---|---|
| `GET_PLAYING_TITLE_NAME` | `GET_TITLE_NAME` |
| `GET_PROTOCOL` | `GET_PROTOCOL_VERSION` |
| `GET_TIME` | `GET_DATE_TIME` |
| `GO_COLLECTIONS` | `GO_FAVORITES` |

SET_SCREEN_MASK

SET_STATUS_CUE_PERIOD

SHOW_NAVIGATION_OVERLAY

SCREEN_MASK_USED

ENABLE_STATUS_CUES

### Other

The following corrections and amendments have been made to this document:

- There is no NETWORK_SETTINGS event command; this is a response only to a GET_NETWORK_SETTINGS request command.
- The full-frame aspect ratio in the ASPECT_RATIO response now indicates the aspect ratio of the video frame output by the Movie Player, and not necessarily that encoded on the DVD itself. The two aspect ratios can be different, as when the Movie Player is doing internal scaling from 4:3 to 16:9 or vice versa.

# Notices

Document Title: Kaleidescape System Control Protocol Reference Manual
Document Number: 101-0005 Rev 10
Publication Date: October 2018
Permanent Link: www.kaleidescape.com/go/control-protocol

This document revision corresponds to Kaleidescape software version 10.4.2 & 8.4.2.

This document is for informational purposes only. Kaleidescape makes no representations or warranties, express or implied, regarding the accuracy or completeness of the information contained herein and Kaleidescape shall have no obligation to provide updates to this information in the future.