

HEOS CLI Bash Scripts (Version 2)

Author: Norman Carver ©2025-2026

Contents

1	Introduction	3
2	Copyright Notice	4
3	Scripts Devices Terminology	5
3.1	Master	5
3.2	Players	5
3.3	Groups	6
4	Some Script Usage Examples	7
5	The Scripts	9
5.1	Master Initialization	9
5.2	HEOS Devices Information	9
5.3	Player/Group Play/Pause/Next/Prev	10
5.4	Player/Group Volume and Muting	10
5.5	Player/Group Repeat and Shuffle	10
5.6	Player/Group Queue	10
5.7	HEOS Favorites	11
5.8	HEOS History	11
5.9	HEOS Playlists	11
5.10	Local Music (DLNA) Server	11
5.11	Group Control	12
5.12	Menu-Based Control	12
6	The HEOS CLI Protocol	13
7	Technical Requirements for the Scripts	14

8	Required Setup for the Script Programs	15
8.1	Step 1: Understand Home Network Configuration	15
8.2	Step 2: Select and Configure Master Device	15
8.3	Step 3: Create Directory and Unzip Scripts	16
8.4	Step 4: Edit Network Information File	16
8.5	Step 5: Edit HEOS Account Information File (optional)	17
9	HEOS System Details	18
9.1	Players vs Rooms vs Zones	18
9.2	Groups vs Stereo Pairs	18
9.3	Volume and Mute Settings	18

1 Introduction

The HEOS CLI Scripts are a set of Bash scripts plus one C program. They provide the ability to control local HEOS players from a *Linux* or (Intel) *ChromeOS* (Crostini) *command line* (i.e., terminal window). They make use of **The HEOS CLI Protocol** to interact with HEOS players.

The scripts allow one to control basic player functions: play, pause/stop, change volume, mute, etc. They also allow one to query for a list of players and groups. HEOS Favorites, History, and Playlists, plus player Queues can be listed out and items selected for playing. One can also query a local DLNA server and play “albums” from it.

The design of the script programs is in line with the Unix design philosophy of creating a set of simple, modular programs that do one thing well (rather than single “monolithic” programs that require a GUI or complex options to control their functionality). Each script implements one particular function: e.g., start a player playing its current selection, query/change a player’s volume, etc.

Another design goal was avoiding the use of utilities that are not part of base Linux installations. For example, while **jq**, a utility for parsing and manipulating JSON, simplifies some operations, it is not part of most base Linux installs (though it can be added). Instead, clever use of Bash capabilities has allowed the scripts to require only the standard Linux utility **grep** (and **sort** if player/group name sorting is wanted).

Finally, be aware that the HEOS CLI protocol is somewhat limited in the functionality it provides, so it is in no way a substitute for the HEOS app. Nonetheless, these scripts accomplished their author’s goal of being able to provide most of the desired player control from his laptops, without needing constant access to a smartphone.

2 Copyright Notice

The author, Norman Carver, retains copyright on these programs. Permission is granted for *individuals* to make *unrestricted use* of the programs, and to be allowed to *modify* them as desired. **Permission is not granted for scraping of the source code, or for use of the programs by corporations or other organizations!** *These actions are a violation of the author's exclusive rights under US and International Copyright Law.*

3 Scripts Devices Terminology

The script programs use the terms *Master*, *Player*, and *Group* when referring to HEOS devices. This section explains what these terms mean.

3.1 Master

HEOS CLI commands are intended to be sent to one HEOS device. This device is referred to as the *Master* in the script programs. The master device is used to control any of the (online) HEOS devices (by sending commands via the LAN).

The scripts are setup to use one particular HEOS device as the *default master*. The default master device must be configured prior to using the scripts. Be sure to read [Required Setup for the Scripts](#).

Should a master device other than the default master need to be used, the script programs allow an alternative master device to be specified. An alternative master is specified via its “*IPNUM*.” An IPNUM is the *host portion* of a device IP address, and will be a number (integer) between 1 and 254 (see **HEOS Devices Setup** for more about this).

The master device may have to be “*initialized*.” See [Master Initialization](#) for information about how to do this.

3.2 Players

There are a number of different types of HEOS-enabled devices that can play music: Denon HEOS/Home speakers, Denon/Marantz Links and Amps, and Denon/Marantz AVRs. All these devices are referred to as *Players* in the script programs. (See [HEOS System Details](#) for more about HEOS devices.)

HEOS devices all have *names*, which are assigned using the HEOS app during setup or later. These are often location names like “**Living Room**”. The scripts allow players to be identified using their names. Name matching is done in a *case insensitive* and *substring* manner to make it easier to type player names. E.g., the player **Living Room** could be selected using any of: **living**, **Living**, **living room**, etc.

If multiple player names are matched by an entered name substring, the scripts automatically use the first player matched. This is useful with device/player names like **Living Room Left** and **Living Room Right**, which are in a *stereo pair*, since only one of these players will be available for controlling the pair, and one may not know/recall which it is. Of course this means that using “**room**” as a script’s player name argument is not a good choice if you have players named as, e.g., **Living Room**, **Bedroom**, etc.

Players can also be selected via their “*IPNUM*.” An IPNUM is the host portion of a device IP address, and will be a number (integer) between 1 and 254 (see **HEOS Devices Setup** for more about this).

[HEOS Devices Information](#) lists the scripts that can provide information about the currently online HEOS players: **heos-players**, **heos-names**, and **heos-info**.

3.3 Groups

HEOS players can be *grouped*. Groups have names just like HEOS players. Most of the scripts can deal with groups as well as players. However, a script must be informed that a name identifies a group rather than a player. This is done by including the “-g” option when invoking a script.

Groups can also be identified by their HEOS *Group ID (GID)*. The script **heos-groups** shows current HEOS groups info. Each group consists of one player/device that is the “*leader*” and one or more “*member*” players/devices. The script **heos-names** with -g option shows the names of the currently active groups.

Groups are best created via the HEOS app, using either the *Group Presets* or by dropping players on others to create a (temporary) group. Groups can also of course be disbanded from the HEOS app. Two script programs allow one to make/disband groups, but do not allow naming (see **Group Control** for more about this).

4 Some Script Usage Examples

- Get the names of the players (currently online):
`heos-players -n`
- Get the names of the groups (currently active):
`heos-groups -n`
- Get status info for a player named “HiFi”:
`heos-status hifi`
- Start a player named “HiFi” playing (its current selection):
`heos-play hifi`
- See what a player named “HiFi” has as its current selection:
`heos-nowplaying hifi`
- Start a group named “Kitchen” playing:
`heos-play -g kitchen`
- Increase the volume of a player named “HiFi”:
`heos-volume hifi +8`
- Mute a player named “HiFi”:
`heos-mute hifi on`
- List items in HEOS Favorites:
`heos-favlist`
- Start a player named “HiFi” playing HEOS Favorites item #1:
`heos-favplay hifi 1`
- Run the menu-driven program for a player named “HiFi”:
`heos-menu hifi`
- Run the menu-driven program for a group named “Kitchen”:
`heos-menu -g kitchen`

Example session using heos-menu:

```
> heos-menu hifi
1) Play          3) Next          5) Fav Item      7) Now Playing  9) Mute          11) State
2) Pause         4) Prev           6) Queue Item   8) Volume       10) UnMute       12) Info
Enter choice (^-d to Quit): 7
type: song
song: Autumn of the Seraphs
album: Autumn of the Seraphs - Bonus EP
artist: Pinback
qid: 3
Enter choice (^-d to Quit): 1
Playing...
Enter choice (^-d to Quit): 8
Enter volume level (0--50) or Return for current: 20
Enter choice (^-d to Quit): 8
Enter volume level (0--50) or Return for current:
level=20
Enter choice (^-d to Quit): 3
Enter choice (^-d to Quit): 7
type: song
song: Scent (Bonus Track)
album: Summer in Abaddon
artist: Pinback
qid: 4
Enter choice (^-d to Quit): 5
Enter Favorites item (>=1) or Return to list:
  1) station: Radio Paradise
  2) station: Radio Paradise Mellow Mix
  3) station: Radio Paradise Global Mix
  4) station: Radio Paradise Rock Mix
  5) station: Lord Huron
  6) station: Future Islands
Enter choice (^-d to Quit): 5
Enter Favorites item (>=1) or Return to list: 1
Playing: 1) station: Radio Paradise
Enter choice (^-d to Quit): 2
Paused!
Enter choice (^-d to Quit): ^d
```

5 The Scripts

The CLI scripts are designed as a set of Bash script programs, each of which accomplishes one task. E.g. `heos-play` to start a player playing its current selection. The top-level scripts to take actions are named as `heos-*`. They make use of two “utility” programs, named as `heosutil-*`. See [Required Setup for the Scripts](#) for how the scripts should be configured on your system.

Some scripts do not require a player be specified, e.g., `heos-names`. Others require a particular player (or group) be specified, e.g., `heos-play`. See sections [Players](#) and [Groups](#) for more information on players and groups.

5.1 Master Initialization

The script programs communicate with a single HEOS device, referred to as the [Master](#). That device may need to be *initialized*: activate its CLI Module and login to the user’s HEOS account. Initialization is accomplished with the following program:

- `heos-login` – Login a device to HEOS account, Master by default.

Note that the login program can take several seconds to run, since it may need to repeatedly send login commands in order to get the master fully initialized.

The program obviously requires the HEOS username and password. Without any setup, one will be prompted to enter these. Since this can become tedious over time, one can enter the information into a file, which will be sourced by the login program. See [Script Files Setup](#) for details.

If the master device has been in use via the HEOS app, it should not be necessary to login/initialize it. However, should any script program result in a “HEOS CLI command failed” error, use the above command to initialize the intended HEOS master device.

The `heos-login` program not only activates the master, but makes sure it can access online HEOS information such as Favorites, History, and Playlists. The command can also be used to login other HEOS devices. This can be handy for CEOL units and AVR’s, so they have access to HEOS info via their remote.

5.2 HEOS Devices Information

Scripts that give information about (currently online) HEOS Players and Groups:

- `heos-players` – Print information about the (online) HEOS Players.
- `heos-groups` – Print information about the current HEOS Groups.
- `heos-info` – Print information about a particular (online) HEOS Player.
- `heos-state` – Print out the play state, mute state, etc. for a HEOS Player/Group.
- `heos-nowplaying` – Print the Now Playing info for a HEOS Player/Group.

5.3 Player/Group Play/Pause/Next/Prev

Play/pause/stop state info and control:

- **heos-next** – Play the next track in the Queue for a HEOS Player/Group.
- **heos-pause** – Pause playing for a HEOS Player/Group.
- **heos-play** – Start playing (the current item) on a HEOS Player/Group.
- **heos-prev** – Play the previous track in the Queue on a HEOS Player/Group.
(Note: resets to start of current if >4secs in, so may have to use twice.)
- **heos-stop** – Stop playing on a HEOS Player/Group.

5.4 Player/Group Volume and Muting

Volume info and control:

- **heos-volume** – Print out or set/change the Volume on a HEOS Player/Group.
- **heos-mute** – Print out or set the Mute state on a HEOS Player/Group.

heos-volume by default (no **ACTION** argument) prints out the current volume level. **ACTION** arguments can set volume to any specific level, or increase or decrease the level.

heos-mute by default (no **ACTION** argument) prints out the current mute state. **ACTION** arguments can mute, unmute, or toggle muting.

5.5 Player/Group Repeat and Shuffle

Repeat and shuffle info and control:

- **heos-repeat** – Print out or set the Repeat mode for a HEOS Player/Group.
- **heos-shuffle** – Print out or set the Shuffle mode for a HEOS Player/Group.

heos-repeat by default (no **ACTION** argument) prints out the current repeat state. **ACTION** arguments can set repeat to all, one, or off.

heos-shuffle by default (no **ACTION** argument) prints out the current shuffle state. **ACTION** arguments can set shuffle to on or off.

5.6 Player/Group Queue

Queue scripts require a player/group be specified, as queues are specific to each player/group:

- **heos-qclear** – Clear the Queue list of a HEOS Player/Group.
- **heos-qlist** – Print the Queue list of a HEOS Player/Group.
- **heos-qplay** – Play a particular HEOS Queue item on a HEOS Player/Group.

5.7 HEOS Favorites

HEOS Favorites (i.e., Presets) are set in the HEOS App, and are system wide:

- **heos-favlist** – Print a list of the HEOS Favorites.
- **heos-favplay** – Play a particular HEOS Favorites item on a HEOS Player/Group.

HEOS Favorites are streaming stations. Playing them does not affect a player’s Queue.

5.8 HEOS History

HEOS History is a list of *Tracks* or *Stations* that have been played on any player/group:

- **heos-histlist** – Print a list of the HEOS History Tracks or Stations.
- **heos-histplay** – Play a particular HEOS History item on a HEOS Player/Group.

Note that items must have actually been *played* on a player/group to appear in History (not simply have been queued).

Recents in the HEOS App provides the same information, but also includes Albums and Playlists categories, which are not available via the HEOS CLI.

Playing a History *track* will *add* the track to the player’s Queue. Playing a History *station* does not affect a player’s Queue.

5.9 HEOS Playlists

HEOS Playlists are created in the HEOS App, and are system wide:

- **heos-pllist** – Print a list of the HEOS Playlists.
- **heos-plplay** – Play a particular HEOS Playlists item on a HEOS Player/Group.

By default, a selected playlist to play *replaces* the player’s Queue contents, but the **-a** option will *add* the playlist to the current Queue contents instead.

5.10 Local Music (DLNA) Server

If a Local Music Server (DLNA Server) is running on the LAN, these scripts allow one to list and play albums by some artist. They are interactive, searching for and listing artists, then listing albums by selected artists, for final play selection:

- **heos-servlist** – Print a list of artists and then albums available on the server.
- **heos-servplay** – Locate and play a particular artist’s album on a HEOS Player/Group.

By default, a selected album to play *replaces* the player’s Queue contents, but the **-a** option *adds* the album to the current Queue contents.

Note that these scripts work down to the “album” level only, they cannot query or play individual tracks (unless an “album” consists of a single track).

5.11 Group Control

Make and remove a HEOS devices group:

- `heos-mkgroup` – Make a new group from some (online) HEOS Players.
- `heos-ungroup` – UnGroup the players from a current HEOS Group.

Note that the HEOS CLI Protocol does not allow for *naming* a group when creating it via the Protocol, so it is generally going to be better to use the HEOS app to create groups (the default group names are long and not meaningful).

5.12 Menu-Based Control

The script `heos-menu` provides a simple menu-driven program (using Bash' `select`). It can run a number of the provided actions on one particular Player/Group. It carries out the actions by calling the other scripts. This provides a simpler way to control one device for an extended time.

`heos-menu` supports the following actions:

- play/pause current selection;
- next/prev queue items;
- volume/muting control;
- nowplaying and state info;
- list/play Favorites, Playlists, Queue, Server;

6 The HEOS CLI Protocol

The script programs make use of the **HEOS Command Line Interface (CLI) Protocol** (specification version 1.17). The protocol documentation can be easily found online.

The HEOS CLI Protocol involves sending command strings to port 1255 of a HEOS device. The device responds by sending one or two responses in *JSON format*. The HEOS CLI Protocol intends that commands be sent to one particular HEOS device, and that device contact other HEOS devices as necessary. The single device to be contacted is referred to as the **Master** in the script programs.

A single device to be sent commands is used in part because HEOS devices do not keep their CLI modules active. The HEOS Protocol document says: “HEOS Speaker runs CLI module in a dormant mode. HEOS speaker spawns CLI core modules when the controller establishes the first socket connection to the speaker.” Being in dormant mode can cause: “Inability of CLI module to process player commands. This is because, by nature of UPnP, CLI module need some time to discover all players....”

While the Protocol document suggests keeping an open socket/connection to the Master device to avoid the CLI module going dormant, this is not practical given the design of the script programs as a set of independent scripts. A small C program was written to open a TCP socket connection to a HEOS device, send a command string, and then read one or more JSON responses. (There appears to be no suitable Linux standard program to accomplish this socket send-receive only task.)

No issues have been encountered when using these programs for an extended period of time, except perhaps for an initial call to the master. If a script program results in a “HEOS CLI command failed” error, use the `heos-login` command on the intended HEOS master device (see **Master Initialization**).

HEOS players/devices are identified to each other via a *Player ID (PID)*. A master maps player names or IPNUMs to PIDs via the script `heosutil-run-command`. Groups are identified by a *Group ID (GID)*. Each group consists of one player/device that is the “*leader*” and one or more “*member*” players/devices. The group GID is the PID of the group leader.

7 Technical Requirements for the Scripts

The script programs are all written for the *BASH* shell, which is the standard shell (CLI) on Linux systems. Other UNIX-like OS's may use different default shells, but Bash should be available, and would have to be installed to run these programs.

The one C program, `heosuntil-send-command-toipaddr.c`, must be compiled to create an executable version that can be run on the target system. The supplied Zip contains an executable compiled with *GCC* (GNU Compiler Collection) for X86-64 Linuxes. Other UNIX-like OS's may be compatible, though at the very least the C source code will need to be recompiled to produce a suitable executable. This should be fairly straightforward, as no libraries other than the standard C library (e.g., `glibc`) are required, and the socket-related system calls are all POSIX standard calls.

8 Required Setup for the Script Programs

To make use of the HEOS CLI Bash scripts, users must complete the following steps:

1. Understand their home network configuration (IP addressing scheme);
2. Select one HEOS device to be the default Master and configure it to use a *static IP address*;
3. Create a directory/folder on the control computer, copy the downloaded Zip file to that directory, and unzip it there;
4. Edit the file `heosinfo-network` with their home network info and the “IPNUM” of the selected default Master;
5. Optional: Edit the file `heosinfo-account` with your HEOS account info;

8.1 Step 1: Understand Home Network Configuration

The first step is to make sure you understand your home network’s configuration. This primarily means knowing the IP address range HEOS devices use. You will also need to know what device on your network is handling DHCP (dynamic IP address assignments), and the IP address range it assigns from.

The scripts are designed assuming a standard home network configuration: all HEOS devices have IPv4 addresses where the first three quads are the *network portion/prefix* and the last quad (lowest order) is the *host portion*. E.g., a device might have IP address `192.168.0.101`, where `192.168.0` is the network portion/prefix, and `101` is the host portion/number.

Home networks with different configurations will likely not work with the scripts as written! The main issue will be with using *IPNUMs* (see below) to identify HEOS devices, since full IP addresses may not be properly constructed. Note, though, that the laptop/computer running these script must be able to communicate with only the *Master device*, since the Master communicates with (controls) the other HEOS devices.

A *DHCP server* is what provides IP addresses to devices using dynamic addresses. For most people, their *router* is probably also their DHCP server. You will need to check its web interface to determine (and maybe modify) the range of IP addresses it assigns from (more on this below).

8.2 Step 2: Select and Configure Master Device

One particular HEOS device must be chosen as the default master device (see section [Master](#) to understand the master). Ideally, this should be a HEOS device that is either always powered on, or at least always powered when CLI-based device control is desired. A HEOS device connected via Ethernet instead of WiFi is also preferred.

The default HEOS master device must be assigned a static IP address! By default, HEOS devices use DHCP to obtain network IP addresses. However, they can be set to use *static IP addresses*. Use of static IP addresses for all HEOS devices is strongly encouraged, as it reduces potential network connection issues.

Static IP addresses can be configured via the HEOS app or through your router (or other DHCP server device). To setup a static IP address via the HEOS app, Click the gear icon to goto **Settings > MyDevices > the device > Advanced > IP Address Settings**. Change **Configuration Type** to “Manual (Static)” and enter the needed info. Use one of the well known public DNS servers instead of your ISP’s, e.g., 1.1.1.1 or 8.8.8.8.

Static IP addresses can also effectively be setup via your *DHCP server* (i.e., via DHCP “reservations”). For most LANs, the DHCP server function is supplied by their router. One complication of setting up static IPs via the DHCP server is that exactly how this is done and the terminology used varies among router brands.

A key point is that the static IP address you assign to the Master cannot fall within the DHCP server’s IP address range! This may require that you adjust the IP address range of your DHCP server (e.g., the router).

In the script documentation, the *host portion* value of an IP address is referred to as an IPNUM, e.g., 101 in 192.168.0.101. IPNUMs can be used in the scripts to identify a master and players. Host portions of IP addresses can potentially range from 1 to 254 (0 and 255 have special meanings). A LAN’s *gateway* device is typically assigned as host 1 or 254.

8.3 Step 3: Create Directory and Unzip Scripts

The downloaded scripts must be put into a *single directory* (i.e., folder). That directory can be placed anywhere and can have any name.

If the directory is in the *shell search path* (PATH environment variable), the scripts can be run simply by typing their names, so that is recommended. A directory can be added to PATH by doing like: `PATH=$PATH:heosdir` (where `heosdir` might be like: `~/Download/heos/`). Of course all of the scripts must be *executable* (`chmod +x`).

Unzip the downloaded `.zip` file directly into the created directory (i.e., “unzip here” rather than “unzip into subdirectory”). This should result in the set of scripts being placed into the directory you created. It should also result in the creation of a single *subdirectory*, named `heosutils`.

All `heos-*` script programs go into the main directory. The `heosutil-*` and `heosinfo-*` programs/files go into the `heosutils` subdirectory. The `heosutil-*` files: are `heosutil-run-command` and `heosutil-send-command-toipaddr` (compiled from `heosutil-send-command-toipaddr.c`). The `heosinfo-*` files: are `heosinfo-account` and `heosinfo-network`.

8.4 Step 4: Edit Network Information File

Users must edit the file `heosinfo-network` in the `heosutils` subdirectory. Information specific to their home network and master device must be entered. This requires setting/modifying the values assigned to two variables:

```
# HEOS LAN info (must be set according to local network config):
networkprefix=192.168.0 #LAN network prefix (e.g.: 192.168.0)
masteripnum=101        #default master device IP address host portion (e.g.: 101)
```

8.5 Step 5: Edit HEOS Account Information File (optional)

Users may also wish to edit the file `heosinfo-account` in the `heosutils` subdirectory, if they wish to be able to login HEOS devices automatically. This requires setting/modifying the values assigned to two variables:

```
heosuser=user@mail    #HEOS account email/username
heospwd=abc123         #HEOS account password
```

9 HEOS System Details

9.1 Players vs Rooms vs Zones

HEOS terminology for devices can be a bit confusing, in part because changes have been made in what terms are used in the app. The HEOS CLI Protocol spec uses the terms “player” and “group.” The HEOS app now refers to “players” as “rooms”, and uses this term when creating of *groups*.

The “room” terminology can be confusing if you have multiple, grouped players in what you consider a single room in your house. For example, having two speakers at opposite ends of a large living room, with names “Living Room Left” and “Living Room Right,” which you have grouped into a group named “Living Room.” The HEOS app will still list the two speakers as individual rooms.

Zones are an element of only AVR’s: different sets of wired speakers that can be enabled/disabled separately. A “zone” is not the same as a HEOS room or group. The script programs do not provide any control over which AVR zones are active.

9.2 Groups vs Stereo Pairs

Another possible source of confusion is groups versus *stereo pairs*. When a pair of HEOS devices (possibly including a sub too) are grouped, they can be set as a stereo pair (via the HEOS app). Once set as a stereo pair, they are *no longer* considered as a HEOS *group*. When the HEOS CLI command `get_players` is executed, only *one* of the speakers in a stereo pair gets listed, and this device does not show a GID.

A stereo pair can be controlled by the scripts, using the single identifying device name or its IPNUM. However, you must know which of the devices is being used by HEOS to identify the stereo pair. To avoid having to remember this, it makes sense to name the players that are going to be part of a stereo pair such that they share a common substring, e.g.: “Living Room Left” and “Living Room Right.” If you now just use “living room” as the player/pair name, the scripts will find whichever player is in control.

9.3 Volume and Mute Settings

The script programs allow one to get, set, and change the volumes on local HEOS devices, as well as mute and unmute devices. However, HEOS volumes and muting are not as straightforward as one might think.

Some HEOS devices may be set to have *fixed output*. While you can get and set volumes on these devices via the scripts, the volume settings are meaningless (do not determine output). However, muting/unmuting commands do work as expected.

Most confusing is how volumes work with *groups*. Each group has a *group volume*, but each individual player/device in a group also has its own *player volume*. When a group is created, the group volume is set to be the player volume of the player that is *group leader*. Each (non-leader) group member’s volume remains unchanged. The scripts can be used to

set/change the group volume, but one can also set/change the volume of each group player specifically. This is handy when group members need different volume settings for balance.

So what happens to player volumes when one sets/changes the group volume? The “group volume” (and group leader’s volume) change as directed. The volumes of each other member player get changed by the *delta* of the change to the group volume. E.g., if the group volume changes from 20 to 25 (delta of +5), a member’s volume that was 17, will change to 22. Note, though, that if group volume is set to 0 (zero), the volume of all players in the group is also set to zero. Thus, if one wishes to set the volume of all players in a group to the same exact volume value, one must first set the group volume to 0 and then set the group volume to the desired value.