

# HEOS CLI Protocol Specification

- 1. Overview
- 2. Connection
  - 2.1 Controller Design Guidelines
    - 2.1.1 Driver Initialization
    - 2.1.2 Caveats
    - 2.1.3 Miscellaneous
- 3. Command and Response Overview
  - 3.1 Commands
  - 3.2 Responses
- 4. Command and Response Details
  - 4.1 System Commands
    - 4.1.1 Register for Change Events
    - 4.1.2 HEOS Account Check
    - 4.1.3 HEOS Account Sign In
    - 4.1.4 HEOS Account Sign Out
    - 4.1.5 HEOS System Heart Beat
    - 4.1.6 HEOS Speaker Reboot
    - 4.1.7 Prettify JSON response
  - 4.2 Player Commands
    - 4.2.1 Get Players
    - 4.2.2 Get Player Info
    - 4.2.3 Get Play State
    - 4.2.4 Set Play State
    - 4.2.5 Get Now Playing Media
    - 4.2.6 Get Volume
    - 4.2.7 Set Volume
    - 4.2.8 Volume Up
    - 4.2.9 Volume Down
    - 4.2.10 Get Mute
    - 4.2.11 Set Mute
    - 4.2.12 Toggle Mute
    - 4.2.13 Get Play Mode
    - 4.2.14 Set Play Mode
    - 4.2.15 Get Queue
    - 4.2.16 Play Queue Item
    - 4.2.17 Remove Item(s) from Queue
    - 4.2.18 Save Queue as Playlist
    - 4.2.19 Clear Queue
    - 4.2.20 Play Next
    - 4.2.21 Play Previous
  - 4.3 Group Commands
    - 4.3.1 Get Groups
    - 4.3.2 Get Group Info
    - 4.3.3 Set Group
    - 4.3.4 Get Group Volume
    - 4.3.5 Set Group Volume
    - 4.2.6 Group Volume Up
    - 4.2.7 Group Volume Down
    - 4.3.8 Get Group Mute
    - 4.3.9 Set Group Mute
    - 4.3.10 Toggle Group Mute
  - 4.4 Browse Commands
    - 4.4.1 Get Music Sources
    - 4.4.2 Get Source Info
    - 4.4.3 Browse Source
    - 4.4.4 Browse Source Containers
    - 4.4.5 Get Source Search Criteria
    - 4.4.6 Search
    - 4.4.7 Play Station
    - 4.4.8 Play Input source
    - 4.4.9 Add Container to Queue with Options
    - 4.4.10 Add Track to Queue with Options
    - 4.4.11 Get HEOS Playlists
    - 4.4.12 Rename HEOS Playlist
    - 4.4.13 Delete HEOS Playlist
    - 4.4.14 Get HEOS History
    - 4.4.15 Retrieve Album Metadata
    - 4.4.16 Get Service Options for now playing screen - OBSOLETE
    - 4.4.17 Set service option

- 5. Change Events (Unsolicited Responses)
  - 5.1 Sources Changed
  - 5.2 Players Changed
  - 5.3 Group Changed
  - 5.4 Source Data Changed
  - 5.5 Player State Changed
  - 5.6 Player Now Playing Changed
  - 5.7 Player Now Playing Progress
  - 5.8 Player Playback Error
  - 5.9 Player Queue Changed
  - 5.10 Player Volume Changed
  - 5.11 Player Mute Changed
  - 5.12 Player Repeat Mode Changed
  - 5.13 Player Shuffle Mode Changed
  - 5.14 Group Status Changed
  - 5.15 Group Volume Changed
  - 5.16 Group Mute Changed
  - 5.17 User Changed
- 6.0 Error Codes
  - 6.1 General Error Response
  - 6.2 Error Code and Text Table

Version	HEOS Version	Modifications	Date	Author
1.0	1.280.96	Initial release	12/20/2014	Prakash Mortha
1.1	1.304.61	Add set service option command	05/27/2015	Prakash Mortha
1.2	1.310.170	Remove support for play url	08/06/2015	Prakash Mortha
1.3	TBD	Add reboot command	TDB	Prakash Mortha

## 1. Overview

The Denon HEOS is a network connected, wireless, multi-room music system. The HEOS Command Line Interface (CLI) allows external control systems to manage, browse, play, and get status from the Denon HEOS products. The HEOS CLI is accessed through a telnet connection between the HEOS product and the control system. The control system sends commands and receives responses over the network connection. The CLI commands and responses are in human readable (ascii) format. The command is a text string and the responses are in JSON format. The commands and responses for browsing music servers and services use a RESTFUL like approach while other commands and responses are more static.

## 2. Connection

The HEOS products can be discovered using the UPnP SSDP protocol. Through discovery, the IP address of the HEOS products can be retrieved. Once the IP address is retrieved, a telnet connection to port 1255 can be opened to access the HEOS CLI and control the HEOS system. The HEOS product IP address can also be set statically and manually programmed into the control system. Search target name (ST) in M-SEARCH discovery request is 'urn:schemas-denon-com:device:ACT-Denon:1'.

The control system should use various Get commands to determine the players and groups currently in the HEOS system.

Controller software can control all HEOS speakers in the network by establishing socket connection with just one HEOS speaker. It is recommended not to establish socket connection to each HEOS speaker. This is to decrease network traffic caused by establishing socket connection to each HEOS speaker. Controller software can open multiple socket connections to the single HEOS speaker. Typically controllers will use one connection to listen for change events and one to handle user actions.

### 2.1 Controller Design Guidelines

#### 2.1.1 Driver Initialization

In order to reduce number of UPnP devices running on the network, 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. What it all means for controller?

- Inability of CLI module to process player commands. This is because, by nature of UPnP, CLI module need some time to discover all

- players before they can be identified by their unique Id (pid)
- Spew of events when controller initially connects to the speaker. In order to avoid excessive event handling in a event driven controller system, the following initialization sequence is suggested:
  1. Un-register for change events. By default speaker doesn't send unsolicited events but still it is a good idea to send un-register command. This is done through 'register\_for\_change\_events' command.
  2. If user credentials are available, sign-in to HEOS user account. This is done through 'sign\_in' command.
  3. Retrieve current HEOS ecosystem status. This is done through commands like 'get\_players', 'get\_sources', 'get\_groups', 'get\_queue', 'get\_now\_playing\_media', 'get\_volume', 'get\_play\_state' etc.
  4. Register for change events. This is done through 'register\_for\_change\_events' command.
- If controller design involves disconnect and reconnect to HEOS speakers through CLI, it is recommended to keep a idle connection to HEOS Speaker thus avoiding CLI module to set back to dormant mode.

## 2.1.2 Caveats

Please take a look at the following suggestions to avoid breaking controller code due to future enhancements

- The 'message' field part of HEOS response is a string. The attribute value pair in this message string is delimited by '&'. Further the attribute name and value is separated by '=' sign. Please note that new arguments can be added in the future.
- New JSON objects may be added to the 'payload' as part of future enhancements.

## 2.1.3 Miscellaneous

- Controllers can add custom argument SEQUENCE=<number> in browse commands to associate command and response. This is possible because the 'message' field in the response packet includes all the arguments sent in the command. Please let us know if you need additional custom argument other than 'SEQUENCE'. This is to avoid accidentally using HEOS command arguments for special purpose.
- Maximum number of simultaneous socket connections supported by HEOS speaker is 32.

# 3. Command and Response Overview

## 3.1 Commands

HEOS CLI commands are in the following general format:

heos://command\_group/command?attribute1=value1&attribute2=value2...&attributeN=valueN

Command string delimiter is "\n".

## 3.2 Responses

The responses to commands are in JSON format and use the following general structure:

```
{
  "heos": {
    "command": "command_group/command",
    "result": "success' or 'fail",
    "message": "other result information"
  },
  "payload": {
    'Rest of response data'
  }
}
```

Some command responses will not include a payload.

If the "result" of the command is "fail" then the "message" information contains the error codes for the failure. The error codes can be found in Section TBD.

Some commands will also cause unsolicited responses. For example, sending the 'player/clear\_queue' command will also cause the Player/Group Queue Changed response and could also cause the Player/Group Status Changed response.

When the actual response can't be populated immediately, a special response will be sent back as shown below. This usually occurs during browse/search as CLI needs to retrieve data from remote media server or online service.

```
{
  "heos": {
    "command": "command_group/command",
    "result": "success",
    "message": "command under process"
  }
}
```

JSON command response delimiter is "\r\n".

## 4. Command and Response Details

### 4.1 System Commands

#### 4.1.1 Register for Change Events

By default HEOS speaker does not send Change events. Controller needs to send this command with enable=on when it is ready to receive unsolicit responses from CLI. Please refer to "Driver Initialization" section regarding when to register for change events.

Command: heos://system/register\_for\_change\_events?enable='on\_or\_off'

Attribute	Description	Enumeration
enable	Register or unregister for change events.	on,off

Response:

```
{
  "heos": {
    "command": "system/register_for_change_events",
    "result": "success",
    "message": "enable='on_or_off'"
  }
}
```

Example: heos://system/register\_for\_change\_events?enable=on

#### 4.1.2 HEOS Account Check

Command: heos://system/check\_account

This command returns current user name in its message field if the user is currently signed in.

Response:

```
{
  "heos": {
    "command": "system/check_account",
    "result": "success",
    "message": "signed_out" or "signed_in&un=<current user name>"
  }
}
```

Example: heos://system/check\_account

#### 4.1.3 HEOS Account Sign In

Command: heos://system/sign\_in?un=heos\_username&pw=heos\_password

Attribute	Description	Enumeration
un	HEOS account username	N/A
pw	HEOS account password	N/A

Response:

```
{
```

```

"heos": {
  "command": "system/sign_in ",
  "result": "success",
  "message": "signed_in&un=<current user name>"
}
}

```

Example: heos://system/sign\_in?un=user@gmail.com&pw=12345

#### 4.1.4 HEOS Account Sign Out

Command: heos://system/sign\_out

Response:

```

{
  "heos": {
    "command": "system/sign_out ",
    "result": "success",
    "message": "signed_out"
  }
}

```

Example: heos://system/sign\_out

#### 4.1.5 HEOS System Heart Beat

Command: heos://system/heart\_beat

Response:

```

{
  "heos": {
    "command": "system/heart_beat ",
    "result": "success"
    "message": ""
  }
}

```

Example: heos://system/heart\_beat

#### 4.1.6 HEOS Speaker Reboot

Using this command controllers can reboot HEOS device. This command can only be used to reboot the HEOS device to which the controller is connected through CLI port.

Command: heos://system/reboot

Response:

```

{
  "heos": {
    "command": "system/reboot",
    "result": "success"
    "message": ""
  }
}

```

Example: heos://system/reboot

#### 4.1.7 Prettify JSON response

Helper command to prettify JSON response when user is running CLI controller through telnet.

Command: heos://system/prettify\_json\_response?enable='on\_or\_off'

Attribute	Description	Enumeration
enable	Enable or disable prettification of JSON response.	on,off

Response:

```

{
  "heos": {

```

```

        "command": "system/prettify_json_response",
        "result": "success",
        "message": "enable='on_or_off'"
    }
}

```

Example: heos://system/prettify\_json\_response?enable=on

## 4.2 Player Commands

### 4.2.1 Get Players

Command: heos://player/get\_players

Attribute	Description	Enumeration
pid	Player id	N/A
gid	pid of the Group leader	N/A
lineout	LineOut level type	1 - variable 2 - Fixed
control	Only valid when lintout level type is Fixed (2).	1 - None 2 - IR 3 - Trigger 4 - Network

Note: The group id field (gid) is optional. The 'gid' field will only be appeared if the player(s) is part of a group.

Note: control field is only populated when lineout level type is Fixed (lineout = 2)

Response:

```

{
  "heos": {
    "command": "player/get_players",
    "result": "success",
    "message": ""
  },
  "payload": [
    {
      "name": "player name 1",
      "pid": "player id 1",
      "gid": "group id",
      "model": "player model 1",
      "version": "player verison 1"
      "lineout": "level type"
      "control": "control option"
    },
    {
      "name": "player name 2",
      "pid": "player id 2",
      "gid": "group id",
      "model": "player model 2",
      "version": "player verison 2"
      "lineout": "level type"
      "control": "control option"
    },
    .
    .
    .
    {
      "name": "player name N",
      "pid": "player id N",
      "gid": "group id",
      "model": "player model N",
      "version": "player verison N"
      "lineout": "level type"
    }
  ]
}

```

```

        "control": "control option"
    }
}

```

Example: heos://player/get\_players

## 4.2.2 Get Player Info

Command: heos://player/get\_player\_info?pid=player\_id

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A
gid	pid of the Group leader	N/A
lineout	LineOut level type	1 - variable 2 - Fixed
control	Only valid when lintout level type is Fixed (2).	1 - None 2 - IR 3 - Trigger 4 - Network

Note: The group id field (gid) is optional. The 'gid' field will only be appeared if the player(s) is part of a group.

Note: control field is only populated when lineout level type is Fixed (lineout = 2)

Response:

```

{
  "heos": {
    "command": "player/get_player_info",
    "result": "success",
    "message": "pid=player_id"
  },
  "payload": {
    "name": "player name",
    "pid": "player id",
    "gid": "group id",
    "model": "player model",
    "version": "player verison",
    "lineout": "level type",
    "control": "control option"
  }
}

```

Example: heos://player/get\_player\_info?pid=1

## 4.2.3 Get Play State

Command: heos://player/get\_play\_state?pid=player\_id

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A

Response:

```

{
  "heos": {
    "command": " player/get_play_state ",
    "result": "success",
    "message": "pid=player_id&state=play_state"
  }
}

```

Example: heos://player/get\_playe\_state?pid=1

## 4.2.4 Set Play State

Command: heos://player/set\_play\_state?pid=player\_id&state=play\_state

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A
state	Player play state	play, pause, stop

Response:

```
{
  "heos": {
    "command": " player/set_play__state ",
    "result": "success",
    "message": "pid='player_id'&state='play_state'"
  }
}
```

Example: heos://player/set\_play\_state?pid=1&state=play

## 4.2.5 Get Now Playing Media

Command: heos://player/get\_now\_playing\_media?pid=player\_id

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A
id (options)	Options available for now playing media	Following options are currently supported for now playing media <ul style="list-style-type: none"><li>• 11 - Thumbs Up (Pandora)</li><li>• 12 - Thumbs Down (Pandora)</li><li>• 19 - Add station to HEOS Favorites</li></ul>

Response:

The following response provides example when the speaker is playing a song.

**Note:** For local music and DLNA servers sid will point to Local Music Source id.

```
{
  "heos": {
    "command": "player/get_now_playing_media",
    "result": "success",
    "message": "pid='player_id'"
  },
  "payload": {
    "type": "song",
    "song": "song name",
    "album": "album name",
    "artist": "artist name",
    "image_url": "image url",
    "mid": "media id",
    "qid": "queue id",
    "sid": source_id
    "album_id": "Album Id"
  }
}
```

The following response provides example when the speaker is playing a station.

```
{
  "heos": {
    "command": "player/get_now_playing_media",
    "result": "success",
```



```

        "message": "pid=player_id"
    },
    "payload": {
        "type": "station",
        "song": "song name",
        "station": "station name",
        "album": "album name",
        "artist": "artist name",
        "image_url": "image url",
        "mid": "media id",
        "qid": "queue id",
        "sid": source_id
    }
    "options": [
        {
            "play": [
                {
                    "id": 19,
                    "name": "Add to HEOS Favorites"
                }
            ]
        }
    ]
}

```

Example: heos://player/get\_now\_playing\_media?pid=1

## 4.2.6 Get Volume

Command: heos://player/get\_volume?pid='player\_id'

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A

Response:

```

{
    "heos": {
        "command": " player/ get_volume ",
        "result": "success",
        "message": "pid='player_id'&level='vol_level'"
    }
}

```

Example: heos://player/get\_volume?pid=1

## 4.2.7 Set Volume

Command: heos://player/set\_volume?pid=player\_id&level=vol\_level

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A
level	Player volume level	0 to 100

Response:

```

{
    "heos": {
        "command": " player/ set_volume ",
        "result": "success",
        "message": "pid='player_id'&level='vol_level'"
    }
}

```

Example: heos://player/set\_volume?pid=2&level=30

## 4.2.8 Volume Up

Command: heos://player/volume\_up?pid=player\_id&step=step\_level

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A
step	Player volume step level	1 to 10(default 5)

Response:

```
{
  "heos": {
    "command": " player/ volume_up ",
    "result": "success",
    "message": "pid=player_id&step=step_level"
  }
}
```

Example: heos://player/volume\_up?pid=2&step=5

## 4.2.9 Volume Down

Command: heos://player/volume\_down?pid=player\_id&step=step\_level

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A
level	Player volume step level	1 to 10(default 5)

Response:

```
{
  "heos": {
    "command": " player/ volume_down ",
    "result": "success",
    "message": "pid=player_id&step=step_level"
  }
}
```

Example: heos://player/volume\_down?pid=2&step=5

## 4.2.10 Get Mute

Command: heos://player/get\_mute?pid=player\_id

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A

Response:

```
{
  "heos": {
    "command": " player/ get_mute ",
    "result": "success",
    "message": "pid=player_id&state=on_or_off"
  }
}
```

Example: heos://player/get\_mute?pid=1

### 4.2.11 Set Mute

Command: heos://player/set\_mute?pid=player\_id&state=on\_or\_off

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A
state	Player mute state	on, off

Response:

```
{
  "heos": {
    "command": " player/ set_mute ",
    "result": "success",
    "message": "pid='player_id'&state='on_or_off'"
  }
}
```

Example: heos://player/set\_mute?pid=3&state=off

### 4.2.12 Toggle Mute

Command: heos://player/toggle\_mute?pid=player\_id

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A

Response:

```
{
  "heos": {
    "command": " player/ toggle_mute ",
    "result": "success",
    "message": "pid=player_id"
  }
}
```

Example: heos://player/toggle\_mute?pid=3

### 4.2.13 Get Play Mode

Command: heos://player/get\_play\_mode?pid=player\_id

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A

Response:

```
{
  "heos": {
    "command": " player/get_play_mode",
    "result": "success",
    "message": "pid='player_id'&&repeat=on_all_or_on_one_or_off&shuffle=on_or_off"
  }
}
```

Example: hoes://player/get\_play\_mode?pid=1

## 4.2.14 Set Play Mode

Command: heos://player/set\_play\_mode?pid=player\_id&repeat=on\_all\_or\_on\_one\_or\_off&shuffle=on\_or\_off

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A
repeat	Player repeat state	on_all, on_one, off
shuffle	Player shuffle state	on, off

Response:

```
{
  "heos": {
    "command": " player/set_play_mode",
    "result": "success",
    "message": "pid=player_id&repeat=on_all_or_on_one_or_off&shuffle=on_or_off"
  }
}
```

Example: heos://player/set\_play\_mode?pid=1&repeat=on\_all&shuffle=off

## 4.2.15 Get Queue

Command: heos://player/get\_queue?pid=player\_id&range=start#, end#

Range is start and end record index to return. Range parameter is optional. Omitting range parameter returns all records but a maximum of 100 records are returned per response.

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A
range	Range is start and end record index to return. Range parameter is optional. Omitting range parameter returns all records up to a maximum of 100 records per response.	range starts from 0

Response:

```
{
  "heos": {
    "command": "player/get_queue",
    "result": "success",
    "message": "pid=player_id&range=start#, end#"
  },
  "payload": [
    {
      "song": "song name 1",
      "album": "album name 1",
      "artist": "artist name 1",
      "image_url": "image_url 1",
      "qid": "queue id 1",
      "mid": "media id 1"

      "album_id": "AlbumId 1"
    },
    {
      "song": "song name 2",
      "album": "album name 2",
      "artist": "artist name 2",
      "image_url": "image_url 2",
      "qid": "queue id 2",
      "mid": "media id 2"

      "album_id": "AlbumId 2"
    },
    .
  ]
}
```

```

    .
    .
    {
      "song": "song name N",
      "album": "album name N",
      "artist": "artist name N",
      "image_url": "image_url N",
      "qid": "queue id N",
      "mid": "media id N"

      "album_id": "AlbumId N"
    }
  ]
}

```

Example: heos://player/get\_queue?pid=1&range=0,10

## 4.2.16 Play Queue Item

Command: heos://player/play\_queue?pid=player\_id&qid=queue\_song\_id

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A
qid	Queue id for song returned by 'get_queue' command	N/A

Response:

```

{
  "heos": {
    "command": " player/play_queue",
    "result": "success",
    "message": "pid='player_id'&qid='queue_id'"
  }
}

```

Example: heos://player/play\_queue?pid=2&qid=9

## 4.2.17 Remove Item(s) from Queue

Command: heos://player/remove\_from\_queue?pid=player\_id&qid=queue\_id\_1,queue\_id\_2,...,queue\_id\_n

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A
qid	List of comma separated queue_id's where each queue id for song is returned by 'get_queue' command	N/A

Response:

```

{
  "heos": {
    "command": "player/remove_from_queue ",
    "result": "success",
    "message": "pid='player_id'&qid=queue_id_1, queue_id_2,...,queue_id_n"
  }
}

```

Example: heos://player/remove\_from\_queue? pid=1&qid=4,5,6

## 4.2.18 Save Queue as Playlist

Command: heos://player/save\_queue?pid=player\_id&name=playlist\_name

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A
name	String for new playlist name limited to 128 unicode characters	N/A

Response:

```
{
  "heos": {
    "command": "player/save_queue ",
    "result": "success",
    "message": "pid=player_id&name=playlist_name"
  }
}
```

Example: heos://player/save\_queue?pid=1&name=great playlist

## 4.2.19 Clear Queue

Command: heos://player/clear\_queue?pid=player\_id

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A

Response:

```
{
  "heos": {
    "command": "player/clear_queue ",
    "result": "success",
    "message": "pid=player_id"
  }
}
```

Example: heos://player/clear\_queue

## 4.2.20 Play Next

Command: heos://player/play\_next?pid=player\_id

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A

Response:

```
{
  "heos": {
    "command": " player/play_next",
    "result": "success",
    "message": "pid=player_id"
  }
}
```

Example: heos://player/play\_next?pid=1

## 4.2.21 Play Previous

Command: heos://player/play\_previous?pid=player\_id

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A

Response:

```
{
  "heos": {
    "command": " player/play_previous",
    "result": "success",
    "message": "pid=player_id"
  }
}
```

Example: heos://player/play\_previous?pid=1

## 4.3 Group Commands

### 4.3.1 Get Groups

Command: heos://group/get\_groups

Response:

```
{
  "heos": {
    "command": "player/get_groups",
    "result": "success",
    "message": ""
  },
  "payload": [
    {
      "name": "group name 1",
      "gid": "group id 1",
      "players": [
        {
          "name": "player name 1",
          "pid": "player id 1",
          "role": "player role 1 (leader or member)"
        },
        {
          "name": "player name 2",
          "pid": "player id 2",
          "role": "player role 2 (leader or member)"
        },
        .
        .
        .
        {
          "name": "player name N",
          "pid": "player id N",
          "role": "player role N (leader or member)"
        }
      ]
    },
    {
      "name": "group name 2",
      "gid": "group id 2",
      "players": [
        {
          "name": "player name 1",
          "pid": "player id 1",
          "role": "player role 1 (leader or member)"
        },
        {
          "name": "player name 2",
          "pid": "player id 2",
          "role": "player role 2 (leader or member)"
        },
        .
        .
        .
        {
          "name": "player name N",
          "pid": "player id N",
          "role": "player role N (leader or member)"
        }
      ]
    }
  ]
}
```

```

        }
    ]
},
.
.
.
{
    "name": "group name N",
    "gid": "group id N",
    "players": [
        {
            "name": "player name 1",
            "pid": "player id 1",
            "role": "player role 1 (leader or member)"
        },
        {
            "name": "player name 2",
            "pid": "player id 2",
            "role": "player role 2 (leader or member)"
        },
        .
        .
        .
        {
            "name": "player name N",
            "pid": "player id N",
            "role": "player role N (leader or member)"
        }
    ]
}
]
}

```

Example: heos://group/get\_groups

### 4.3.2 Get Group Info

Command: heos://group/get\_group\_info?gid=group\_id

Attribute	Description	Enumeration
gid	Group id returned by 'get_groups' command	N/A

Response:

```

{
    "heos": {
        "command": "player/get_groups",
        "result": "success",
        "message": "gid=group_id"
    },
    "payload": {
        "name": "group name 1",
        "gid": "group id 1",
        "players": [
            {
                "name": "player name 1",
                "pid": "player id 1",
                "role": "player role 1 (leader or member)"
            },
            {
                "name": "player name 2",
                "pid": "player id 2",
                "role": "player role 2 (leader or member)"
            },
            .
            .
            .
            {
                "name": "player name N",

```



```

        "pid": "player id N",
        "role": "player role N (leader or member)"
    }
}
}

```

Example: heos://group/get\_group\_info?gid=1

### 4.3.3 Set Group

This command is used to perform the following actions:

Create new group:

Creates new group. First player id in the list is group leader.

Ex: heos://group/set\_group?pid=3,1,4

Modify existing group members:

Adds or delete players from the group. First player id should be the group leader id.

Ex: heos://group/set\_group?pid=3,1,5

Ungroup all players in the group

Ungroup players. Player id (pid) should be the group leader id.

Ex: heos://group/set\_group?pid=3

Command: heos://group/set\_group?pid=player\_id\_leader, player\_id\_member\_1,...,player\_id\_member\_n

Attribute	Description	Enumeration
pid	List of comma separated player_id's where each player id is returned by 'get_players' or 'get_groups' command; first player_id in list is group leader	N/A

Response:

The following response provides example when a group is created/modified.

```

{
  "heos": {
    "command": "player/set_group ",
    "result": "success",
    "message": "gid=new group_id&name='group_name'&pid=player_id_1, player_id_2,...,player_id_n"
  }
}

```

The following response provides example when all the speakers in the group are un-grouped.

```

{
  "heos": {
    "command": "player/set_group ",
    "result": "success",
    "message": "pid=player_id"
  }
}

```

Example: heos://group/set\_group?pid=3,1,4

### 4.3.4 Get Group Volume

Command: heos://group/get\_volume?gid=group\_id

Attribute	Description	Enumeration
gid	Group id returned by 'get_groups' command	N/A

Response:

```
{
  "heos": {
    "command": "group/get_volume ",
    "result": "success",
    "message": "gid=group_id&level=vol_level"
  }
}
```

Example: heos://group/get\_volume?gid=1

### 4.3.5 Set Group Volume

Command: heos://group/set\_volume?gid=group\_id&level=vol\_level

Attribute	Description	Enumeration
gid	Group id returned by 'get_groups' command	N/A
level	Group volume level	0 to 100

Response:

```
{
  "heos": {
    "command": "group/set_volume ",
    "result": "success",
    "message": "gid=group_id&level=vol_level"
  }
}
```

Example: heos://group/set\_volume?gid=1&level=30

### 4.2.6 Group Volume Up

Command: heos://group/volume\_up?gid=group\_id&step=step\_level

Attribute	Description	Enumeration
gid	Group id returned by 'get_groups' command	N/A
step	Group volume step level	1 to 10(default 5)

Response:

```
{
  "heos": {
    "command": " group/ volume_up ",
    "result": "success",
    "message": "gid=group_id&step=step_level"
  }
}
```

Example: heos://group/volume\_up?gid=1&step=5

### 4.2.7 Group Volume Down

Command: heos://group/volume\_down?gid=group\_id&step=step\_level

Attribute	Description	Enumeration
gid	Group id returned by 'get_groups' command	N/A
level	Group volume step level	1 to 10(default 5)

Response:

```
{
  "heos": {
    "command": " group/ volume_down ",
    "result": "success",
    "message": "gid='group_id'&step='step_level'"
  }
}
```

Example: heos://group/volume\_down?gid=1&step=5

### 4.3.8 Get Group Mute

Command: heos://group/get\_mute?gid=group\_id

Attribute	Description	Enumeration
gid	Group id returned by 'get_groups' command	N/A

Response:

```
{
  "heos": {
    "command": "group/ get_mute ",
    "result": "success",
    "message": "gid='group_id'&state='on_or_off'"
  }
}
```

Example: heos://group/get\_mute?gid=1

### 4.3.9 Set Group Mute

Command: heos://group/set\_mute?gid=group\_id&state=on\_or\_off

Attribute	Description	Enumeration
gid	Group id returned by 'get_groups' command	N/A
state	Group mute state	on, off

Response:

```
{
  "heos": {
    "command": "group/ set_mute ",
    "result": "success",
    "message": "gid=group_id'&state='on_or_off'"
  }
}
```

Example: heos://group/set\_mute?gid=1&state=off

### 4.3.10 Toggle Group Mute

Command: heos://group/toggle\_mute?gid=group\_id

Attribute	Description	Enumeration
gid	Group id returned by 'get_groups' command	N/A

Response:

```
{
```

```

    "heos": {
      "command": "group/ toggle_mute ",
      "result": "success",
      "message": "gid=group_id"
    }
  }
}

```

Example: heos://group/toggle\_mute?gid=1

## 4.4 Browse Commands

### 4.4.1 Get Music Sources

Command: heos://browse/get\_music\_sources

Response:

```

{
  "heos": {
    "command": "browse/get_music_sources",
    "result": "success",
    "message": ""
  },
  "payload": [
    {
      "name": "source name 1",
      "image_url": "source logo url 1",
      "type": "source type 1",
      "sid": source_id_1
    },
    {
      "name": "source name 2",
      "image_url": "source logo url 2",
      "type": "source type 2",
      "sid": source_id_2
    },
    {
      "name": "source name N",
      "image_url": "source logo url N",
      "type": "source type N",
      "sid": source_id_N
    }
  ]
}

```

Example: heos://browse/get\_music\_sources

The following are valid source types:

- music\_service
- heos\_service
- heos\_server
- dna\_server

### 4.4.2 Get Source Info

Command: heos://browse/get\_source\_info?sid=source\_id

Attribute	Description	Enumeration
sid	Source id returned by 'get_music_sources' command (Or) Source id returned by 'browse' command when browsing source types 'heos_server' and 'heos_service'	N/A

Response:

```
{
  "heos": {
    "command": "browse/get_source_info",
    "result": "success",
    "message": ""
  },
  "payload": [
    {
      "name": "source name",
      "image_url": "source logo url",
      "type": "source type",
      "sid": source_id
    }
  ]
}
```

Example: heos://browse/get\_source\_info

The following are valid source types:

- music\_service
- heos\_service
- heos\_server
- dna\_server

### 4.4.3 Browse Source

Command: heos://browse/browse?sid=source\_id

Attribute	Description	Enumeration
sid	Source id returned by 'get_music_sources' command (Or) Source id returned by 'browse' command when browsing source types 'heos_server' and 'heos_service'	N/A
id (options)	Options available for current browse level	Following options are currently supported for 'Browse Source' command <ul style="list-style-type: none"> <li>• 13 - Create New Station (Pandora)</li> <li>• 20 - Remove from HEOS Favorites (Favorites)</li> </ul>
range	Range is start and end record index to return. Range parameter is optional. Omitting range parameter returns all records up to a maximum of either 50 or 100 records per response. The default maximum number of records depend on the service type.	range starts from 0  NOTE: Range in Browse source command is only supported while browsing Favorites

This command is used under two scenarios.

Browsing actual media sources of type 'heos\_server' and 'heos\_service'.

The command 'Get Music Sources' lists all music servers (type 'heos\_server') in the network under one virtual source called 'Local Music'. Other virtual source that represents all auxiliary inputs (type 'heos\_service') is 'AUX Input'.

Browsing top music view.

Results of this command depends on the music source selected.

**Note:** Optionally this command returns service 'options' that are available for current browse items. Please refer to 'Get Service Options for now playing screen' for service options available on now playing screen.

**Note:** The following response provides examples of the various service options. The actual response will depend on the service options available for a given source type.

Response while browsing actual media sources of type 'heos\_server' and 'heos\_service'. These includes 'Local Music', 'History', 'AUX Inputs', 'Playlists', and 'Favorites'.

```
{
  "heos": {
    "command": "browse/browse",
    "result": "success",
    "message": "sid=source_id&returned=items_in_current_response&count=total_items_available"
```

```

},
"payload": [
  {
    "name": "source name 1",
    "image_url": "source logo url 1",
    "sid": "source id 1",
    "type": "source type 1"
  },
  {
    "name": "source name 2",
    "image_url": "source logo url 2",
    "sid": "source id 2",
    "type": "source type 2"
  },
  {
    "name": "source name N",
    "image_url": "source logo url N",
    "sid": "source id N",
    "type": "source type N"
  }
],
"options": [
  {
    "browse": [
      {
        "id": 13,
        "name": "create new station"
      }
    ]
  }
]
}

```

Example: <heos://browse/browse?sid=1>

Response when browsing top music view in an actual music server/music services.

**Note:** the following response provides examples of the various media types. The actual response will depend on the source browsed and the hierarchy supported by that source.

```

{
  "heos": {
    "command": "browse/browse",
    "result": "success",
    "message": "sid=source_id&returned=items_in_current_response&count=total_items_available"
  },
  "payload": [
    {
      "container": "yes",
      "playable": "no",
      "type": "artist",
      "name": "artist name",
      "image_url": "artist image url",
      "cid": "container id",
      "mid": "media id"
    },
    {
      "container": "yes",
      "playable": "yes",
      "type": "album",
      "name": "album name",
      "image_url": "album image url",
      "artist": "artist name",
      "cid": "container id",
      "mid": "media id"
    },
    {
      "container": "no",
      "playable": "yes",
      "type": "song",
      "name": "song name",
      "image_url": "album image url",
      "artist": "artist name",
      "album": "album name",

```

```

        "mid": "media id"
    },
    {
        "container": "yes",
        "playable": "no",
        "type": "container",
        "name": "container name",
        "image_url": "container image url",
        "cid": "container id",
        "mid": "media id"
    },
    {
        "container": "no",
        "playable": "yes",
        "type": "station",
        "name": "station name",
        "image_url": "station url",
        "mid": "media id"
    }
}

```

Example: heos://browse/browse?sid=1346442495

Supported Sources: Local Media Servers, Playlists, History, Aux-In, Favorites, TuneIn, Pandora, Rhapsody, Deezer, SiriusXM, iHeartRadio, Napster

#### 4.4.4 Browse Source Containers

Command: heos://browse/browse?sid=source\_id&cid=container\_id&range=start#, end#

Attribute	Description	Enumeration
sid	Source id returned by 'get_music_sources' command	N/A
cid	Container id returned by 'browse' or 'search' command	N/A
range	Range is start and end record index to return. Range parameter is optional. Omitting range parameter returns all records up to a maximum of either 50 or 100 records per response. The default maximum number of records depend on the service type.	range starts from 0
count	Total number of items available in the container. NOTE: count value of '0' indicates unknown container size. Controllers needs to query until the return payload is empty (returned attribute is 0).	0 - unknown >1 - valid count
returned	Number of items returned in current response	N/A
id (options)	Options available for current browse level	Following options are currently supported for 'Browse Source container' command: 1 - Add Track to Library (Rhapsody) 2 - Add Album to Library (Rhapsody) 3 - Add Station to Library (Rhapsody) 4 - Add Playlist to Library (Rhapsody) 5 - Remove Track from Library (Rhapsody) 6 - Remove Album from Library (Rhapsody) 7 - Remove Station from Library (Rhapsody) 8 - Remove Playlist from Library (Rhapsody) 13 - Create New Station (Pandora)

The following are valid media types:

- song
- station
- genre
- artist
- album
- container

**Note:** A "yes" for the "container" field as well as the "playable" field implies that the container supports adding all media items to the play queue. Adding all media items of the container to the play queue is performed through ["Add containers to queue"](#) command.

**Note:** Following response provides examples of the various media types. The actual response will depend on the source browsed and the hierarchy supported by that source.

Response:

```
{
  "heos": {
    "command": "browse/browse",
    "result": "success",
    "message":
      "sid='source_id&cid='container_id'&range='start,end'&returned=items_in_current_response&count=total_items_available"
  },
  "payload": [
    {
      "container": "yes",
      "playable": "no",
      "type": "artist",
      "name": "artist name",
      "image_url": "artist image url",
      "cid": "container id",
      "mid": "media id"
    },
    {
      "container": "yes",
      "playable": "yes",
      "type": "album",
      "name": "album name",
      "image_url": "album image url",
      "artist": "artist name",
      "cid": "container id",
      "mid": "media id"
    },
    {
      "container": "no",
      "playable": "yes",
      "type": "song",
      "name": "song name",
      "image_url": "album image url",
      "artist": "artist name",
      "album": "album name",
      "mid": "media id"
    },
    {
      "container": "yes",
      "playable": "no",
      "type": "container",
      "name": "container name",
      "image_url": "container image url",
      "cid": "container id",
      "mid": "media id"
    },
    {
      "container": "no",
      "playable": "yes",
      "type": "station",
      "name": "station name",
      "image_url": "station url",
      "mid": "media id"
    }
  ],
  "options": [
    {
      "browse": [
        {
          "id": 4,
          "name": "Add Playlist to Library"
        }
      ]
    }
  ]
}
```



```
}  
]
```

Example: heos://browse/browse?sid=2&cid=TopAlbums&range=0,100

Supported Sources: Local Media Servers, Playlists, History, Aux-In, TuneIn, Pandora, Rhapsody, Deezer, SiriusXM, iHeartRadio, Napster

#### 4.4.5 Get Source Search Criteria

Command: heos://browse/get\_search\_criteria?sid=source\_id

Attribute	Description	Enumeration
sid	Source id returned by 'get_music_sources' command	N/A

**Note:** the following response provides examples of the various search criteria types. The actual response will depend on the source and the search types supported by that source.

Response:

```
{  
  "heos": {  
    "command": "browse/ get_search_criteria ",  
    "result": "success",  
    "message": "sid='source_id "  
  },  
  "payload": [  
    {  
      "name": "Artist",  
      "scid": "search_criteria_id",  
      "wildcard": "yes_or_no",  
    },  
    {  
      "name": "Album",  
      "scid": "search_criteria_id",  
      "wildcard": "yes_or_no",  
    },  
    {  
      "name": "Track",  
      "scid": "search_criteria_id",  
      "wildcard": "yes_or_no",  
    },  
    {  
      "name": "Station",  
      "scid": "search_criteria_id",  
      "wildcard": "yes_or_no",  
    }  
  ]  
}
```

Example: heos://browse/get\_search\_criteria?sid=3

Supported Sources: Local Media Servers, TuneIn, Rhapsody, Deezer, SiriusXM, Napster

#### 4.4.6 Search

Command: heos://browse/search?sid=source\_id&search=search\_string&scid=search\_criteria&range=start#, end#

Attribute	Description	Enumeration
sid	Source id returned by 'get_music_sources' command	N/A
search	String for search limited to 128 unicode characters and may contain '*' for wildcard if supported by search criteria id	N/A
scid	Search criteria id returned by 'get_search_criteria' command	artist, album, song, station

count	Total number of items available in the container. NOTE: count value of '0' indicates unknown container size. Controllers needs to query until the return payload is empty (returned attribute is 0).	0 - unknown >1 - valid count
range	Range is start and end record index to return. Range parameter is optional. Omitting range parameter returns all records up to a maximum of 50/100 records per response. The default maximum number of records depend on the service type.	range starts from 0
returned	Number of items returned in current response	N/A

Response:

**Note:** the following response provides examples of the various media types. The actual response will depend on the source searched and the results returned for the search string.

```
{
  "heos": {
    "command": "browse/search",
    "result": "success",
    "message": "sid='source_id&scid='search_criteria_id'&range='start#,
end#'&returned=items_in_current_response&count='total_items_available"
  },
  "payload": [
    {
      "container": "yes",
      "playable": "no",
      "type": "artist",
      "name": "artist name",
      "image_url": "artist image url",
      "cid": "container id",
      "mid": "media id"
    },
    {
      "container": "yes",
      "playable": "yes",
      "type": "album",
      "name": "album name",
      "image_url": "album image url",
      "artist": "artist name",
      "cid": "container id",
      "mid": "media id"
    },
    {
      "container": "no",
      "playable": "yes",
      "type": "song",
      "name": "song name",
      "image_url": "album image url",
      "artist": "artist name",
      "album": "album name",
      "mid": "media id"
    },
    {
      "container": "yes",
      "playable": "no",
      "type": "container",
      "name": "container name",
      "image_url": "container image url",
      "cid": "container id",
      "mid": "media id"
    },
    {
      "container": "no",
      "playable": "yes",
      "type": "station",
      "name": "station name",
      "image_url": "station url",
      "mid": "media id"
    }
  ]
}
```

```
}  
}
```

Example: heos://browse/search?sid=2&search="U2"&scid=1

Supported Sources: Local Media Servers, TuneIn, Rhapsody, Deezer, Napster

## 4.4.7 Play Station

Command: heos://browse/play\_stream?pid=player\_id&sid=source\_id&cid=container\_id&mid=media\_id&name=station\_name

Attribute	Description	Enumeration
sid	Source id returned by 'get_music_sources' command	N/A
cid	Container id returned by 'browse' command. Ignore if container id doesn't exists as in case of playing station obtained through 'Search' command	N/A
mid	Media id returned by 'browse' or 'search' command	N/A
pid	Player id returned by 'get_players' or 'get_groups' command	N/A
name	Station name returned by 'browse' command.	N/A

Note: The mid for this command must be a 'station' media type.

Response:

**Note:** this command will cause a Now Playing Change Event to occur if a new stream is played.

```
{  
  "heos": {  
    "command": " browse/play_stream ",  
    "result": "success",  
    "message": "pid=player_id&sid=source_id&cid=container_id&mid=media_id&name=station_name"  
  }  
}
```

Example: heos://browse/play\_stream?pid=1&sid=2&cid='CID-55'&mid=15376&name=Q95

Supported Sources: History, Favorites, TuneIn, Pandora, Rhapsody, Deezer, SiriusXM, iHeartRadio, Napster

## 4.4.8 Play Input source

Command to play input source on the same speaker:

heos://browse/**play\_input**?pid=player\_id&input=input\_name

Command to play input source on another speaker:

heos://browse/**play\_input**?pid=destination\_player\_id&spid=source\_player\_id&input=input\_name

OBSOLETE command that requires sid:

heos://browse/**play\_stream**?pid=player\_id&sid=source\_id&mid=media id

Attribute	Description	Enumeration
sid	Source id returned by 'get_music_sources' command	N/A
pid	player id of the selected speaker (destination HEOS speaker) Player id returned by 'get_players' or 'get_groups' command	N/A
mid	media id returned by 'browse' command	N/A
spid	player id of the HEOS device which is acting as the source Player id returned by 'get_players' or 'get_groups' command	N/A

input	input source name	"inputs/aux_in_1"
	Note: Validity of Inputs depends on the type of source HEOS device	<inputs aux_in_2"<br=""></inputs> <inputs aux_in_3"<br=""></inputs> <inputs aux_in_4"<br=""></inputs> <inputs line_in_1"<br=""></inputs> <inputs line_in_2"<br=""></inputs> <inputs line_in_3"<br=""></inputs> <inputs line_in_4"<br=""></inputs> <inputs coax_in_1"<br=""></inputs> <inputs coax_in_2"<br=""></inputs> <inputs coax_in_3"<br=""></inputs> <inputs coax_in_4"<br=""></inputs> <inputs optical_in_1"<br=""></inputs> <inputs optical_in_2"<br=""></inputs> <inputs optical_in_3"<br=""></inputs> <inputs optical_in_4"<br=""></inputs> <inputs hdmi_in_1"<br=""></inputs> <inputs hdmi_in_2"<br=""></inputs> <inputs hdmi_arc_1"<br=""></inputs> <inputs <="" hdmi_arc_2"="" td=""></inputs>

Response for command "heos://browse/play\_input?pid=player\_id&input=input\_name" :

**Note:** this command will cause a Now Playing Change Event to occur if an aux in stream is played.

```
{
  "heos": {
    "command": "browse/play_input",
    "result": "success",
    "message": "pid=player_id&input=input_name"
  }
}
```

Examples:

heos://browse/play\_input?pid=1234&input=inputs/aux\_in\_1

heos://browse/play\_input?pid=1234&spid=9876&input=inputs/aux\_in\_1

heos://browse/play\_stream?pid=1&sid=1441320818&mid=inputs/aux\_in\_1

## 4.4.9 Add Container to Queue with Options

Command: heos://browse/add\_to\_queue?pid=player\_id&sid=source\_id&cid=container\_id&aid=add\_criteria

Attribute	Description	Enumeration
sid	Source id returned by 'get_music_sources' command	N/A
cid	Container id returned by 'browse' or 'search' command	N/A
aid	Add criteria id as defined by enumerations ->	1 – play now 2 – play next 3 – add to end 4 – replace and play
pid	Player id returned by 'get_players' or 'get_groups' command	N/A

Note: The cid for this command must be a 'playable' container type.

Response:

**Note:** this command will cause a Now Playing Change Event to occur if a new song is played.

```
{
  "heos": {
    "command": "browse/add_to_queue",
    "result": "success",
    "message": "pid=player_id&sid=source_id&cid=container_id&aid=add_criteria"
  }
}
```

Example: heos://browse/add\_to\_queue?pid=1&sid=5&cid=Artist/All&aid=2

Supported Sources: Local Media Servers, Playlists, History, Rhapsody playable containers, Deezer playable containers, iHeartRadio playable

containers, Napster playable containers

#### 4.4.10 Add Track to Queue with Options

Command: `heos://browse/add_to_queue?pid=player_id&sid=source_id&cid=container_id&mid=media_id&aid=add-criteria`

Attribute	Description	Enumeration
sid	Source id returned by 'get_music_sources' command	N/A
cid	Container id returned by 'browse' or 'search' command	N/A
mid	Media id returned by 'browse' or 'search' command	N/A
aid	Add criteria id as defined by enumerations ->	1 – play now 2 – play next 3 – add to end 4 – replace and play
pid	Player id returned by 'get_players' or 'get_groups' command	N/A

Note: The mid for this command must be a 'track' media type.

Response:

**Note:** this command will cause a Now Playing Change Event to occur if a new song is played.

```
{
  "heos": {
    "command": " browse/add_to_queue",
    "result": "success",
    "message": "pid='player_id'&sid='source_id'&cid='container_id'&mid='media_id'&aid='add_criteria'"
  }
}
```

Example: `heos://browse/add_to_queue?pid=1&sid=8&cid=Artists/All&mid=9&aid=1`

Supported Sources: Local Media Servers, Playlists, History, Rhapsody Tracks, Deezer Tracks, iHeartRadio Tracks, Napster

#### 4.4.11 Get HEOS Playlists

Refer to [Browse Sources](#) and [Browse Source Containers](#)

#### 4.4.12 Rename HEOS Playlist

Command: `heos://browse/rename_playlist?sid=source_id&cid=container_id&name=playlist_name`

Attribute	Description	Enumeration
sid	Source id returned by 'get_music_sources' command; select HEOS source to get HEOS playlists.	N/A
cid	Container id returned by 'browse' command; use Playlist container to get HEOS playlists	N/A
name	String for new playlist name limited to 128 unicode characters	N/A

Response:

```
{
  "heos": {
    "command": "browse/rename_playlist ",
    "result": "success",
    "message": "sid='source_id'&cid='container_id'&name='playlist_name'"
  }
}
```

Example: `heos://browse/rename_playlist?sid=11&cid=234&name=new name`

#### 4.4.13 Delete HEOS Playlist

Command: `heos://browse/delete_playlist?sid=source_id&cid=container_id`

Attribute	Description	Enumeration
sid	Source id returned by 'get_music_sources' command; select HEOS source to get HEOS playlists.	N/A
cid	Container id returned by 'browse' command; use Playlist container to get HEOS playlists	N/A

Response:

**Note:** The HEOS History has two containers: one for songs and another for stations. The following response example is for the songs container. The station container returns the list of stations.

```
{
  "heos": {
    "command": "browse/delete_playlist ",
    "result": "success",
    "message": "sid='source_id'&cid='container_id'"
  }
}
```

Example: heos://browse/delete\_playlist?sid=11&cid=234

#### 4.4.14 Get HEOS History

Refer to Browse Sources and Browse Source Containers

#### 4.4.15 Retrieve Album Metadata

Retrieve image url associated with a given album id. This command facilitates controllers to retrieve and update their UI with cover art, if image\_url in browse/search/get\_queue/get\_now\_playing\_media command response is blank.

Command: [heos://browse/retrieve\\_metadata?sid=source\\_id&cid=album\\_id](heos://browse/retrieve_metadata?sid=source_id&cid=album_id)

Attribute	Description	Comment
sid	Source id returned by 'get_music_sources' command; select HEOS source to get HEOS playlists.	Currently supported media sources are Rhapsody/Napster
cid	Container id returned by 'browse' command or 'get_now_playing_media' command	Rhapsody/Napster album ids

Note: Supported music service is Rhapsody.

Response:

```
{
  "heos": {
    "command": "browse/retrieve_metadata",
    "result": "success",
    "message": "sid=2&cid=album_id&returned=items_in_current_response&count=total_items_available"
  },
  "payload": [
    {
      "album_id": "album_id",
      "images": [
        {
          "image_url": "URL to image file",
          "width": current image width
        },
        .
        .
        .
        {
          "image_url": "URL to image file",
          "width": current image width
        }
      ]
    }
  ]
}
```

```
}
```

Example: [heos://browse/retrieve\\_metadata?sid=2&cid=Alb.184664171](heos://browse/retrieve_metadata?sid=2&cid=Alb.184664171)

#### 4.4.16 Get Service Options for now playing screen - OBSOLETE

Obsolete - Now `get_now_playing_media` command will include supported option for currently playing media.

Command: `heos://browse/get_service_options?sid=source_id`

Attribute	Description	Enumeration
sid	Source id returned by 'get_music_sources' command	N/A
id (options)	Options available on now playing screen	Following options are currently supported for 'Get Service options for now playing screen': 11 - Thumbs Up (Pandora) 12 - Thumbs Down (Pandora)

**Note:** This command returns service options that are only available on 'now playing' screen. Please refer to 'Browse Source' and 'Browse Source Containers' for service options available on various browse levels.

**Note:** the following response provides examples of the various service options. The actual response will depend on the service options available for a given source type.

Response:

```
{
  "heos": {
    "command": "browse/get_service_options",
    "result": "success",
    "message": ""
  },
  "payload": [
    {
      "play": [
        {
          "id": 11,
          "name": "Thumbs Up"
        },
        {
          "id": 12,
          "name": "Thumbs Down"
        }
      ]
    }
  ]
}
```

Example: `heos://browse/get_service_options?sid=5`

#### 4.4.17 Set service option

Set service option is a generic command used to select any of the supported service options provided through 'Get Service Options for now playing screen', 'Browse Sources' and 'Browse Source Containers' command response.

Following service options are currently supported:

Option id	Example Command	Parameter description
1 - Add Track to Library (Rhapsody)	<code>heos://browse/set_service_option?sid=2&amp;option=1&amp;mid=Tra.174684187</code>	mid - track id obtained through 'browse source containers' command
2 - Add Album to Library (Rhapsody)	<code>heos://browse/set_service_option?sid=2&amp;option=2&amp;cid=Alb.174684186</code>	cid - album id obtained through 'browse source containers' command
3 - Add Station to Library (Rhapsody)	<code>heos://browse/set_service_option?sid=2&amp;option=3&amp;mid=sas.6513639</code>	mid - station id obtained through 'browse source containers' command

4 - Add Playlist to Library (Rhapsody)	heos://browse/set_service_option?sid=2&option=4&cid=LIBPLAYLIST-pp.175573149&name=Lupe Fiasco	cid - playlist id obtained through 'browse source containers' command name - name of the playlist obtained through 'browse source container' command
5 - Remove Track from Library (Rhapsody)	heos://browse/set_service_option?sid=2&option=5&mid=Tra.174684187	mid - track id obtained through 'browse source containers' command
6 - Remove Album from Library (Rhapsody)	heos://browse/set_service_option?sid=2&option=6&cid=LIBALBUM-Alb.174684186	cid - album id obtained through 'browse source containers' command
7 - Remove Station from Library (Rhapsody)	heos://browse/set_service_option?sid=2&option=7&mid=sas.6513639	mid - station id obtained through 'browse source containers' command
8 - Remove Playlist from Library (Rhapsody)	heos://browse/set_service_option?sid=2&option=8&cid=LIBPLAYLIST-mp.186017722	cid - playlist id obtained through 'browse source containers' command
11 - Thumbs Up (Pandora)	heos://browse/set_service_option?sid=1&option=11&pid=-409995282	pid - player id obtained through 'get_players' command
12 - Thumbs Down (Pandora)	heos://browse/set_service_option?sid=1&option=12&pid=-409995282	pid - player id obtained through 'get_players' command
13 - Create New Station (Pandora)	heos://browse/set_service_option?sid=1&option=13&name=Love  <a href="heos://browse/set_service_option?sid=1&amp;option=13&amp;name=Love&amp;range=0,10">heos://browse/set_service_option?sid=1&amp;option=13&amp;name=Love&amp;range=0, 10</a>	name - search string for creating new station Note: This command returns station ids. Controllers need to use 'play station' command to play a station.
19 - Add station to HEOS Favorites	heos://browse/set_service_option?option=19&pid=-409995282	pid - player id obtained through 'get_players' command  Note: Adds currently playing station to HEOS Favorites
20 - Remove from HEOS Favorites	heos://browse/set_service_option?option=20&mid=sas.6513639	mid - station id obtained through 'browse source' command on Favorites

Note: Option 13 (Create New Station) supports optional range queries.

Response:

```
{
  "heos": {
    "command": " browse/set_service_option",
    "result": "success",
    "message": "sid=source_id&option=option_id&mid=media_id"
  }
}
```

Example: [heos://browse/set\\_service\\_option?sid=2&option=1&mid=Tra.174684187](heos://browse/set_service_option?sid=2&option=1&mid=Tra.174684187)

## 5. Change Events (Unsolicited Responses)

### 5.1 Sources Changed

Response:

```
{
  "heos": {
    "command": "event/sources_changed",
  }
}
```

### 5.2 Players Changed

Response:

```
{
```



```
    "heos": {
      "command": "event/players_changed",
    }
  }
```

### 5.3 Group Changed

Response:

```
{
  "heos": {
    "command": "event/groups_changed",
  }
}
```

### 5.4 Source Data Changed

Response:

```
{
  "heos": {
    "command": "event/source_data_changed",
    "message": "sid='source_id'"
  }
}
```

### 5.5 Player State Changed

Response:

```
{
  "heos": {
    "command": "event/player_state_changed",
    "message": "pid='player_id'&state='play_state'"
  }
}
```

### 5.6 Player Now Playing Changed

Response:

```
{
  "heos": {
    "command": "event/player_now_playing_changed",
    "message": "pid=player_id"
  }
}
```

### 5.7 Player Now Playing Progress

Response:

```
{
  "heos": {
    "command": "event/player_now_playing_progress",
    "message": "pid=player_id&cur_pos=position_ms&duration=duration_ms"
  }
}
```

### 5.8 Player Playback Error

Response:

```
{
  "heos": {
    "command": " event/player_playback_error",
    "message": "pid=player_id&error=Could Not Download"
  }
}
```

Note: error string represents error type. Controller can directly display the error string to the user.

## 5.9 Player Queue Changed

Response:

```
{
  "heos": {
    "command": " event/player_queue_changed",
    "message": "pid='player_id'"
  }
}
```

## 5.10 Player Volume Changed

Response:

```
{
  "heos": {
    "command": "event/player_volume_changed ",
    "message": "pid='player_id'&level='vol_level'"
  }
}
```

## 5.11 Player Mute Changed

Response:

```
{
  "heos": {
    "command": "event/player_mute_changed",
    "message": "pid='player_id'&state='on_or_off'"
  }
}
```

## 5.12 Player Repeat Mode Changed

Response:

```
{
  "heos": {
    "command": "event/repeat_mode_changed",
    "message": "pid='player_id'&repeat='on_all_or_on_one_or_off'"
  }
}
```

## 5.13 Player Shuffle Mode Changed

Response:

```
{
  "heos": {
    "command": "event/shuffle_mode_changed",
    "message": "pid='player_id'&shuffle='on_or_off'"
  }
}
```

## 5.14 Group Status Changed

Response

```
{
  "heos": {
    "command": "event/group_changed",
    "message": "gid='group_id'"
  }
}
```

## 5.15 Group Volume Changed

Response:

```
{
  "heos": {
    "command": "event/group_volume_changed ",
    "message": "gid='group_id'&level='vol_level'"
  }
}
```

## 5.16 Group Mute Changed

Response:

```
{
  "heos": {
    "command": "event/group_mute_changed",
    "message": "gid='group_id'&state='on_or_off'"
  }
}
```

## 5.17 User Changed

Response:

```
{
  "heos": {
    "command": "event/user_changed",
    "message": "signed_out" or "signed_in&un=<current user name>"
  }
}
```

# 6.0 Error Codes

## 6.1 General Error Response

Response:

```
{
  "heos": {
    "command": "command_group'/command",
    "result": "fail",
    "message": "eid='error_id&text=error text&command_arguments'"
  }
}
```

## 6.2 Error Code and Text Table

Description	Code	Text Example
-------------	------	--------------

Unrecognized Command	1	Command not recognized.
Invalid ID	2	ID not valid
Wrong Number of Command Arguments	3	Command arguments not correct.
Requested data not available	4	Requested data not available.
Resource currently not available	5	Resource currently not available.
Invalid Credentials	6	Invalid Credentials.
Command Could Not Be Executed	7	Command not executed.
User not logged In	8	User not logged in.
Parameter out of range	9	Out of range
User not found	10	User not found
Internal Error	11	System Internal Error
System Error	12	System error&syserrno=-2
Processing Previous Command	13	Processing previous command
Media can't be played	14	cannot play
Option not supported	15	Option not supported