HEOS CLI Protocol Specification

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 where the commands use URL encoding with attribute – value pairs and the responses are 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

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 is a 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:

- 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.
- If user credentials are available, sign-in to HEOS user account. This is done through 'sign_in' command.
- 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.
- Register for change events. This is done through 'register_for_change_events' command.

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
The command strings should be encoded using URL encoding.

Command string delimiter is "\r\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

Controller needs to send this command with enable=on when it comes online, or when it is ready to receive unsolicit responses from CLI.

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 singed 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 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

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

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'"
                },
                {
                    "name": "'player name 2'",
                    "pid": "player id 2'",
                    "gid": "group id'",
                    "model": "'player model 2'",
                    "version": "'player verison 2'"
                },
                .
                .
                .
                .
                {
                    "name": "'player name N'",
                    "pid": "player id N'",
                    "gid": "group id'",
                    "model": "'player model N'",
                    "version": "'player verison N'"
                }
            ]
}
```
Example: heos://player/get_players

4.2.2 Get Player Info

Command: heos://player/get_player_info?pid=player_id

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

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

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'"
            }
}
```
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

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
            }
}
```

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
        }
}
```

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'"
                },
                {
                    "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'"
                },
                .
                .
                .
                .
                {
                    "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'"
                }
            ]
}
```

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%20playlist

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 Set Media Opinion - FEATURE NOT IMPLEMENTED

Command: heos://player/set_opinion?pid=player_id&state=up_or_down

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A
state	State of opinion	up, down

Note: This command is currently only supported for Pandora stations to support the Thumbs Up and Thumbs down feature. An Invalid Command error response will be returned if this command is sent when a Pandora station is not playing.
Response:
```
{
    "heos": {
            "command": "player/ set_opinion",
            "result": "success",
            "message": "pid='player_id'&state='up_or_down'
        }
}
```

Example: heos://player/set_opinion?pid=5&state=down

4.2.21 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

4.2.22 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
- dlna_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
- dlna_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

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.

Response while browsing actual media sources of type 'heos_server' and 'heos_service'. These includes 'Local Music', 'History', 'AUX Inputs' and 'Playlists'.

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

Example: heos://browse/browse?sid=2

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&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, TuneIn, Pandora, Rhapsody, Deezer, SiriusXM

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 100 records per response	range starts from 0
count	Total number of items available in the container	0 - unknown >1 - valid count

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'&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=2&cid=TopAlbums&range=0,100

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

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

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	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 100 records per response	range starts from 0

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#'&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

4.4.7 Play Station

Command: 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
mid	Media id returned by 'browse'or 'search' command	N/A
pid	Player id returned by 'get_players' or 'get_groups' 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&mid='media_id'"
}
}
```

Example: heos://browse/play_stream?pid=1&sid=2&mid=15376

Supported Sources: History, TuneIn, Pandora, Rhapsody, Deezer, SiriusXM

4.4.8 Play Aux In

Command: 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 returned by 'get_players' or 'get_groups' command	N/A
mid	media id returned by 'browse' command	N/A

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

```
{
"heos": {
"command": " browse/play_stream ",
"result": "success",
"message": "pid='palyer_id'&sid='source_id"
}
}
```
Example: heos://browse/play_stream?pid=1&sid=1441320818&mid=inputs/2

4.4.9 Play URL

Command: heos://browse/play_stream?pid=player_id&url=url_path

Attribute	Description	Enumeration
pid	Player id returned by 'get_players' or 'get_groups' command	N/A
url	Absolute path to a playable stream	N/A

Response:
Note: this command will cause a Now Playing Change Event to occur if url is played.

```
{
    "heos": {
            "command": " browse/play_stream ",
            "result": "success",
            "message": "pid='palyer_id'&url='path to stream"
        }
}
```
Example: heos://browse/play_stream?pid=1&url=http://10.110.25.159:49152/web/138.mp3

4.4.10 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 Albums, Deezer Albums

4.4.11 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

4.4.12 Get HEOS Playlists

Refer to Browse Sources and Browse Source Containers

4.4.13 HEOS Playlist Rename - FEATURE NOT IMPLEMENTED

Command: heos://browse/rename_playlist?sid=source_id&cid=contaiiner_id& mid=media _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
mid	Media id of Playlist returned by 'browse'or 'search' command	N/A
name	String for new playlist name limited to 128 unicode characters	N/A

Note: The mid for this command must be a 'HEOS Playlist' media type.
Response:
{
"heos": {
"command": "browse/rename_playlist ",
"result": "success",
"message": "sid='source_id'&cid='containier_id'& mid='media _id'&name='playlist_name'"
}
}
Example: heos://browse/rename_playlist?sid=0&cid=1&mid=4&name=new%20name

4.4.14 HEOS Playlist Delete - FEATURE NOT IMPLEMENTED

Command: heos://browse/delete_playlist/sid=source_id&cid=containiner_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
mid	Media id of Playlist returned by 'browse'or 'search' command	N/A

Note: The cid for this command must be a 'HEOS Playlist' container type.
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='containier_id'& mid='media _id'"
}
}
Example: heos://browse/delete_playlist?sid=0&cid=1&mid=4

4.4.15 Get HEOS History

Refer to Browse Sources and Browse Source Containers

4.4.16 Add Media Item to Service - FEATURE NOT IMPLEMENTED

Command: heos://browse/add_media?sid=source_id&mid=media_id

Attribute	Description	Enumeration
sid	Source id returned by 'get_music_sources' command	N/A
mid	Media id returned by 'browse'or 'search' command	N/A

Note: The mid for this command must be an 'album', 'track' or 'station' media type.
Response:
{
"heos": {
"command": " browse/add_media",
"result": "success",
"message": "sid='source_id'&mid='media_id'"
}
}
Example: heos://browse/add_media?sid=3&mid=271

4.4.17 Remove Media Item from Service - FEATURE NOT IMPLEMENTED

Command: heos://browse/remove_media?sid=source_id&mid=media_id

Attribute	Description	Enumeration
sid	Source id returned by 'get_music_sources' command	N/A
mid	Media id returned by 'browse'or 'search' command	N/A

Note: The mid for this command must be an 'album', 'track' or 'station' media type.
Response:
```
{
"heos": {
"command": " browse/remove_media",
"result": "success",
"message": "sid='source_id'&mid='media_id'"
}
}
```
Example: heos://browse/remove_media?sid=3&mid=271

4.4.18 Add Variety - FEATURE NOT IMPLEMENTED

Command: heos://browse/add_variety?sid=source_id&mid=media_id

Attribute	Description	Enumeration
sid	Source id returned by 'get_music_sources' command	N/A
mid	Media id returned by 'browse'or 'search' command	N/A

Note: The mid for this command must be an 'artist' or 'track' media type.
Response:
```
{
"heos": {
"command": "browse/add_variety",
"result": "success",
"message": "sid='source_id'&mid='media_id'"
}
}
```
Example: heos://browse/add_variety?sid=4&mid=856

4.4.19 Bookmark Artist / Track - FEATURE NOT IMPLEMENTED

Command: heos://browse/bookmark?sid=source_id&mid=media_id

Attribute	Description	Enumeration
sid	Source id returned by 'get_music_sources' command	N/A
mid	Media id returned by 'browse'or 'search' command	N/A

Note: The mid for this command must be an 'artist' or 'track' media type.
Response:
```
{
"heos": {
"command": " browse/bookmark",
"result": "success",
"message": "sid='source_id'&mid='media_id'"
}
}
```
Example: heos://browse/bookmark?sid=4&mid=21

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 Queue Changed

Response:

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

5.9 Player Volume Changed

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

5.10 Player Mute Changed

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

5.11 Player Repeat Mode Changed

Response:

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

5.12 Player Shuffle Mode Changed

Response:

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

5.13 Group Status Changed

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

5.14 Group Volume Changed

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

5.15 Group Mute Changed

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

5.16 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

Respone:
```
{
    "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
Wroing 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