Yamaha Extended Control API Specification (Advanced)

Rev. 2.10

Contents

1. Preface

This document describes specifications of API of Yamaha Extended Control (YXC, hereafter) for Yamaha A/V products. YXC is Yamaha's new communication protocol sent over Ethernet and Wi-Fi to control MusicCast enabled devices. This document describes only advanced function (Link etc.). Basic function is described in Yamaha Extended Control API Specification(Basic).

This specification (Rev.2. 1 0) applies only to MusicCast devices that have been upgraded to a product firmware that supports after API version 2.00.

2. Definitions

- MusicCast
  A function to share music easily among audio devices in multiple rooms at homes
- MusicCast Device(s)
  All products/devices that are equipped with MusicCast functions
- MusicCast Network
  A communication network on which MusicCast Devices join.
  - ➢ By multiple MusicCast Devices communicating each other on the same network (=MusicCast Network), all MusicCast devices can be operated using MusicCast App/YXC
- Location
  A unit of MusicCast Network being comprised. Location is expressed as Location ID made up by 32 characters
  - ➢ Max number of MusicCast Devices within 1 Location is up to 32 (available on or after API version 2.00: the max number before version 2.00 was 10).
- Link
  A function to share an audio stream among rooms
- Link Master Room
  A master room an audio distribution comes from
- Link Client Room(s)
  Client room(s) a music track to be distributed from Link Master Room
- Linked Rooms
  All rooms being linked including master and clients, or a Group
- Linked Room
  A room of linked client room(s)
- Group
  All lined rooms including master and clients, or Linked Rooms
- Device
  Equal to MusicCast Device
- Room
  A minimum component MusicCast devices can be shown/associated within a MusicCast network
- Main Zone Sync
  A function to input the same source as Main Zone to Zone 2/3.

3. API Overview

- Request Parameters
  Parameter information to inquire. Parameters stated as "required = yes" have always to be specified as one of inquiring parameters, while parameters marked as "required = no" can be omitted or need to be used in special cases
- Response Parameters
  Parameter information to be returned in the body of response (in JSON object format). Not all parameters listed here are always to be returned in a response, depending on each device specs/functions/setup situations. "response_code" is always returned
  - Note: JSON format data expressions in this document get well formatted for readability, but actual data don't include spaces, intends and carriage return/line feeds
- Base URL
  "<BaseURL>" shown in URI is http://{host}/YamahaExtendedControl
  - {host} is an IP address of a Device to get controlled
- API version
  Values like "v1" or "v2" specified in URI shows a corresponding API version number. Backward compatibility of APIs always being assured unless specified, all APIs that have equal to or smaller API version number(s) that can be read via "getDeviceInfo" are supported on a Device
- Data size of a sting
  In case a string is specified with its data size, it is a size before JSON escape process

4. Zone

4.1. setLinkControl

For setting Link Control in each Zone.

Request Parameters

name	type	required	description
zone	string	yes	Specifies target zone. Available for zones with this function Values: "main" / "zone2" / "zone3" / "zone4"
control	string	yes	Specifies Link Control setting Values: Values gotten via /system/getFeatures

URI	<BaseURL>/v1/<zone>/setLinkControl?control=<control>
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/main/setLinkControl?control=normal

Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to "Response Code List" for details

Example Response	{
	   "response_code":0
	}

4.2. setLinkAudioDelay

For setting Link Audio Delay in each Zone. This setting is invalid when Link Control setting is " Stability Boost ".

Request Parameters

name	type	required	description
zone	string	yes	Specifies target zone. Available for zones with this function Values: "main" / "zone2" / "zone3" / "zone4"
delay	string	yes	Specifies Link Audio Delay setting Values: Values gotten via /system/getFeatures

URI	<BaseURL>/v1/<zone>/setLinkAudioDelay?delay=<delay>
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/main/setLinkAudioDelay?delay=lip_sync

Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to "Response Code List" for details

Example Response	{
	
	   "response_code":0
	
	}

4.3. setLinkAudioQuality

For setting Link Audio Quality in each Zone.

Request Parameters

name	type	required	description
zone	string	yes	Specifies target zone. Available for zones with this function Values: "main" / "zone2" / "zone3" / "zone4"
mode	string	yes	Specifies Link Audio Quality setting Values: Values gotten via /system/getFeatures

URI	<BaseURL>/v1/<zone>/setLinkAudioQuality?mode=<mode>
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/main/setLinkAudioQuality?mode=compressed

Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to "Response Code List" for details

Example Response	{
	
	   "response_code":0
	
	}

5. Distribution (Link)

APIs in regard to Link distribution related setting and getting information

5.1. getDistributionInfo

For retrieving a Device information related to Link distribution

Request Parameters

URI	<BaseURL>/v1/dist/getDistributionInfo
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/dist/getDistributionInfo

Response Parameters

name	type	Description
response_code	integer	0: Successful request / Others: Error Refer to "Response Code List" for details
group_id	string	Returns Group ID in 32-digit hex
group_name	string	Returns Group Name
role	string	Returns a role of Link distribution Values: "server" / "client" / "none"
status	string	Returns a construction state of distribution system Valid only when role is Server. (Valid for API Version 2.00 or later) Values: "building" / "working" / "deleting"
server_zone	string	Returns target Zone ID that can work as a client of distributing server. If there's no parameter of this, "main" is implicitly chosen. Values: "main" / "zone2" / "zone3" / "zone4"
client_list	array	Returns registered client IP address list Valid only when the role is "server"
|- ip_address	string	Returns clients' IP Address
|- data_type	string	Returns a type of distributed data Values: "base" / "ext"
build_disable	array	Returns information on whether distribution construction is prohibited or not If there is no parameter, please handle as it is not in the construction prohibited state
|- role	string	Returns role of distribution that construction is prohibited Values: "server" / "client"
|- reasons	string array	Returns informations of reason when construction of distribution is prohibited Values: "unknown" / "not_implemented"
audio_dropout	boolean	Returns whether sound interruption was detected during distribution This value is cleared when CONNECT is executed or Network Module is reset Even when distribution is canceled, this value is not cleared
mc_surround	object	Returns informations of MusicCast Surround
|- id	Integer	Returns ID of MusicCast Surround

		When ID is 0, it indicates a non-built state Values : 0 - 4294967295
|- role	string	Returns role of MusicCast Surround Values : "none" / "master_multi_main" / "master_stereo_lr" / "master_stereopair_l" / "master_stereopair_r" / "slave_surround_l" / "slave_surround_r" / "slave_surround_lr" / "slave_stereopair_l" / "slave_stereopair_r" / "slave_subwoofer"
|- status	string	Returns build state of MusicCast Surround Valid only when role is Master Values : "none" / "building" / "working" / "deleting" / "ch_overlap" (the work channels of wlan 0 and wlan 1 overlap)
|- slave_info	array	Returns information of Slave device It can be get only when the role is Master
|- role	string	Returns role of Slave device Values : "slave_surround_l" / "slave_surround_r" / "slave_surround_lr" / "slave_stereopair_l" / "slave_stereopair_r" / "slave_subwoofer"
|- status	string	Returns build state of Slave device Values : "none" / "building" / "working" / "deleting" / "losing" (lost from the master device) / "build_pending" (during standby) / "ch_overlap" (the work channels of wlan 0 and wlan 1 overlap)
|- ip_address	string	Returns IP address of Slave device
|- mac_address	string	Returns MAC Address (12 digit ASCII) of Slave device
|- master_info	object	Returns information of Master device It can be get only when the role is Slave
|- role	string	Returns role of Master device Values : "master_multi_main" / "master_stereo_lr" / "master_stereopair_l" / "master_stereopair_r"
|- ip_address	string	Returns IP address of Master device
|- mac_address	string	Returns MAC Address (12 digit ASCII) of Master device
|- build_disable	array	Returns information on whether MusicCast Surround build is prohibited or not If this parameter does not exist, please be treated as especially to build not a prohibited state
|- role	string	Returns roles of MusicCast Surround for which build is prohibited Values : "master" / "slave"
|- reasons	string array	When the build of MusicCast Surround is prohibited, returns information (plural) indicating the reason Values : "unknown"

Example Response	{
	    "response_code":0,
	    "group_id":"9A237BF5AB80ED3C7251DFF49825CA42",
	    "group_name":"[Link] Living Room",
	    "role":"server",
	    "status":" working ",
	    "server_zone":"main",
	    "client_list":[

```
        {
          "ip_address":"192.168.0.5",
          "data_type":"base"
        },
        {
          "ip_address":"192.168.0.11",
          "data_type":"base"
        },
        {
          "ip_address":"192.168.0.22,
          "data_type":"ext"
        }
      ]
      "build_disable":[
        {
          "role":"client",
          "reasons":[
            "unknown"
          ]
        }
      ]
      "audio_dropout":false,
      "mc_surround":{
        "id":123456789,
        "role":"master_multi_main",
        "status":"working":
        "slave_info":[
          {
            "role":"slave_surround_l",
            "ip_address":"192.168.0.11",
            "mac_address":"00A0DE000001"
          },
          {
            "role":"slave_surround_r",
            "ip_address":"192.168.0.12",
            "mac_address":"00A0DE000002"
          }
        ],
        "build_disable":[
          {
            "role":"slave",
            "reasons":[
              "unknown"
            ]
          }
        ]
      }
    }
```

5.2. setServerInfo

For setting a Link distribution server (Link master)

Request Parameters

Name	type	required	description
group_id	string	yes	Specify Group ID in 32-digit hex. Specify "" (empty text) here to cancel a Device being the Link distribution server. Group ID will be initialized ("000...") after the cancel operation
Zone	string	no	Specifies which target Zone ID to be the Link distribution server. If nothing is specified, current setting is kept. Zone ID to be the Link distribution server is confirmable using system/getFeatures server_zone_list. Values: "main" / "zone2" / "zone3" / "zone4"
Type	string	no	Specifies a type of adding or removing clients. Not necessary to specify when canceling the Link master status, Values: "add" / "remove"
client_list	string array	no	Specifies IP addresses of adding/removing clients. Specifiable up to 9 clients

URI	\<BaseURL\>/v1/dist/setServerInfo
Method	POST
Example Request (URI)	http://{host}/YamahaExtendedControl/v1/dist/setServerInfo

Example Request (json)	{
	  "group_id":"9A237BF5AB80ED3C7251DFF49825CA42",
	  "zone":"main",
	  "type":"add",
	  "client_list":[
	    "192.168.0.5",
	    "192.168.0.11"
	  ]
	}

Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to "Response Code List" for details

Example Response	{
	  "response_code":0
	}

5.3. setClientInfo

For setting Link distributed clients. If a Device is already setup as Link distribution server, this client setup is denied by that Device: use this API after canceling a Device's Link distribution server setup using setServerInfo, then confirming that the target Device's role is changed to other

values than "server" using getDistributionInfo.

<u>Request Parameters</u>

name	type	required	description
group_id	string	yes	Specifies Group ID in 32-digit hex. Specify "" (empty text) here to cancel a Device being a Link distributed client. Group ID will be initialized ("000...") after the cancel operation
zone	string array	no	Specifies which target Zone ID to be a Link distributed client. Not necessary to specify when cancelling a client status. Values: "main" / "zone2" / "zone3" / "zone4"
server_ip_address	string	no	Specifies the IP Address of the Link distribution server

URI	<BaseURL>/v1/dist/setClientInfo
Method	POST
Example Request (URI)	http://{host}/YamahaExtendedControl/v1/dist/setClientInfo

Example Request (json)	{
	  "group_id":"9A237BF5AB80ED3C7251DFF49825CA42",
	  "zone":[
	    "main",
	    "zone2"
	  ]
	}

<u>Response Parameters</u>

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to "[Response Code List](#)" for details

Example Response	{
	  "response_code":0
	}

5.4. startDistribution

For initiating Link distribution. This is valid to a Device that is setup as Link distribution server.

<u>Request Parameters</u>

name	type	required	description
num	integer	yes	Specifies Link distribution number on current MusicCast Network

URI	<BaseURL>/v1/dist/startDistribution?num=<num>
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/dist/startDistribution?num=2

<u>Response Parameters</u>

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to "Response Code List" for details

Example Response	{
	   "response_code":0
	}

5.5. stopDistribution

For quitting Link distribution. This is valid to a Device that is setup as Link distribution server.

Request Parameters

URI	<BaseURL>/v1/dist/stopDistribution
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/dist/stopDistribution

Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to "Response Code List" for details

Example Response	{
	   "response_code":0
	}

5.6. setGroupName

For setting up Group Name. Note that Group Name is reserved in volatile memory

Request Parameters

name	type	required	description
name	string	yes	Specifies Group Name. Use UTF-8 within 128 bytes. Default name would be used if it's not setup or "" (empty text) is specified

URI	<BaseURL>/v1/dist/setGroupName
Method	POST
Example Request (URI)	http://{host}/YamahaExtendedControl/v1/dist/setGroupName

Example Request (json)	{
	   "name":"[Link] Living Room"
	}

Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to "Response Code List" for details

Example Response	{
	
	  "response_code":0
	
	}

5.7. setMcSurround

Set MusicCast Surround. It is effective only for devices that can become Master as a role of MusicCast Surround.

<u>**Request Parameters**</u>

name	type	required	description
id	integer	no	Specify the ID of MusicCast Surround. Please specify a value other than 0 when building. When adding a device, specifying a value is unnecessary. When 0 is specified, it releases MusicCast Surround. Values : 0 ～ 4294967295
role	string	no	Specify the role of MusicCast Surround of the target device. When "none" is specified, all the builded MusicCast Surround will be canceled. Since you can not overwrite the role of MusicCast Surround, you need to cancel MusicCast Surround once when you change it. Values : "none" / "master_multi_main" / "master_stereo_lr" / "master_stereopair_l" / "master_stereopair_r"
slave_info	array	no	Specify the information of the device to be added to the Slave of MusicCast Surround.
| - role	string	no	Specify the role of MusicCast Surround of the target device. Since you can not overwrite the role of MusicCast Surround, you need to cancel MusicCast Surround once when you change it. Values : "slave_surround_l" / "slave_surround_r" / "slave_surround_lr" / "slave_stereopair_l" / "slave_stereopair_r" / "slave_subwoofer"
| - ip_address	string	no	Specify the IP adress of the target device.

URI	<BaseURL>/v1/dist/setMcSurround
Method	POST
Example Request (URI)	http://{host}/YamahaExtendedControl/v1/dist/setMcSurround

Example Request (json)	{
	  " id":123456789,
	  "role":"master_multi_main",
	  "slave_info":[
	    {
	      "role":"slave_surround_l",
	      "ip_address":"192.168.0.11"
	    },
	    {
	      "role":"slave_surround_r",
	      "ip_address":"192.168.0.12"
	    }

	]
	}

Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error
		Refer to "Response Code List" for details

Example Response	{
	"response_code":0
	}

5.8. resetMcSurround

Reset MusicCast Surround.
The role of the device is valid in either Master / Slave. When Reset is sent to the master device, even if MusicCast Surround is built, it is not requested to release the Slave device.
If the Master device falls into an uncontrollable state and you can no longer cancel the Slave device from MusicCast Surround via the Master device, please reset MusicCast Surround information with this command.
Please note that MusicCast Surround information will be overwritten again by a reconnection request from the master device when resetting the Slave device with the Master device in MusicCast Surround.

Request Parameters

URI	<BaseURL>/v1/dist/resetMcSurround
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/dist/resetMcSurround

Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error
		Refer to "Response Code List" for details

Example Response	{
	"response_code":0
	}

5.9. confirmMcSurround

Make a request to check the role of MusicCast Surround on the display unit of the device itself.
It can also use it as a way to check the role after building MusicCast Surround and to identify the target device before construction.
Refer to the specifications of the device body for the display specification for each role.

Request Parameters

name	type	required	description
type	string	yes	Set the type of request.
			While displaying the main unit indicator, you can extend the

			display time by specifying "Start" as the type and sending the command.Values : "start" / "stop"
role	string	no	Specify the role. Please specify "Role" which can operate for each device or "none" from "/system/getFeatures". It is valid only when type is "start", and when "role" is not specified, it is treated as "none" is specified. Designation is unnecessary when type is "stop".Values : "none" / "master_multi_main" / "master_stereo_lr" / "master_stereopair_l" / "master_stereopair_r" / "slave_surround_l" / "slave_surround_r" / "slave_surround_lr" / "slave_stereopair_l" / "slave_stereopair_r" / "slave_subwoofer"
duration	integer	no	Specify the display time of the indicator in seconds. It is valid only when type is "start", and if duration is not specified, it is treated as if 255 was specified. Designation is unnecessary when type is "stop". Values : 1 - 255

URI	<BaseURL>/v1/dist/confirmMcSurround?type=<type>&role=<role>&duration=<duration>
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/dist/confirmMcSurround?type=start&role=master_multi_main&duration=10

Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to "Response Code List" for details

Example Response	{
	   "response_code":0
	}

5.10. getMcSpeakerSettings

Get settings and information on the MusicCast Surround speaker.

Request Parameters

name	type	required	description
channel	string	no	Specify the channel. By specifying it individually, you can only get information on that channel. If you specify a "common" to get the information of the common functions. If you specify "all" will find all the information. If they are not specified it will be the "all" is specified. Values : "front_l" / "front_r" / "front" / "center" / "surr_l" / "surr_r" / "surr" / "swfr" / "common" / "all"

URI	<BaseURL>/v1/dist/getMcSpeakerSettings?channel=<channel>

Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/dist/getMcSpeakerSettings?channel=all

Response Parameters

name	type	Description
response_code	integer	0: Successful request / Others: Error Refer to "Response Code List" for details
common	object	Returns information on common functions not dependent on Channel
|- attribute	integer	Returns attribute information of common function Attributes are represented by the OR of the following bit fields. It becomes 0 if the target function is not currently available or does not exist. b[0]　　Test Tone configurable
|- test_tone	boolean	Returns output state of Test Tone
front_l	object	Returns information on the Front L channel Only parameters whose function is enabled by "/system/getFeatures" can be acquired. *Information on each channel exists only when the corresponding channel or function exists. Also, the format of the stored data is common to all channels.
|- attribute	integer	Returns channel attribute information Attributes are represented by the OR of the following bit fields It becomes 0 if the target function is not currently available or does not exist. b[0]　　Channel Level configurable b[1]　　reserved b[2]　　reserved b[3]　　reserved b[4]　　Distance configurable b[5]　　Phase configurable
|- level	integer	Returns channel Level Values: Values calculated from the minimum value / maximum value / step value that can be obtained with "/system/getFeatures".
|- distance	integer	Returns distance (unit : "cm") Values: Values calculated from the minimum value / maximum value / step value that can be obtained with "/system/getFeatures".
|- phase	string	Returns phase setting Values : "normal" / "reverse"
front_r	object	Returns information on Front R channel
front_l	object	Returns information on Front L channel
center	object	Returns information on Center channel
surr_l	object	Returns information on Surround L channel
surr_r	object	Returns information on Surround R channel
surr	object	Returns information on Surround channel
swfr	object	Returns information on Subwoofer channel

Example Response	{

	"response_code":0,
	"front_l":{
	  "attribute":19,
	  "level":3,
	  "test_tone":false,
	  "distance":300
	},
	…
	"swfr":{
	  "attribute":1,
	  "level"1
	}
	}

5.11. setMcSpeakerTestTone

Set the Test Tone output of the MusicCast Surround speaker.

Request Parameters

name	type	required	Description
enable	boolean	yes	Set the output of Test Tone.

URI	<BaseURL>/v1/dist/setMcSpeakerTestTone?enable=<enable>
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/dist/setMcSpeakerTestTone?enable=true

Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to "Response Code List" for details

Example Response	{
	  "response_code":0
	}

5.12. setMcChannelSettings

MusicCast Sets each channel of the surround speaker.
You can specify only one channel at a time, but you can set multiple parameters for that channel at the same time.

Request Parameters

name	type	required	description
channel	string	yes	Specify channel. Values : "front_l" / "front_r" / "front" / "center" / "surr_l" / "surr_r" / "surr" / "swfr"
level	integer	no	Specify channel level. Values: Values calculated from the minimum value / maximum value / step value that can be obtained with

			"/system/getFeatures".
distance	integer	no	Specify distance. Values: Values calculated from the minimum value / maximum value / step value that can be obtained with "/system/getFeatures".
phase	string	no	Specify phase setting. Values : "normal" / "reverse"
URI	<BaseURL>/v1/dist/setMcChannelSettings?channel=<channel>&level=<level>&test_tone=<test_tone>&distance=<distance>		
Method	GET		
Example Request	http://{host}/YamahaExtendedControl/v1/dist/setMcChannelSettings?channel=front_l&level=3&distance=150		

Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to "Response Code List" for details

Example Response	{
	   "response_code":0
	}

6. Response Code List

In a response contains one of Response Codes listed below. No other data is included to a response if its response code was other than '0' (Successful request)

code	description
0	Successful request
1	Initializing
2	Internal Error
3	Invalid Request (A method did not exist, a method wasn't appropriate etc.)
4	Invalid Parameter (Out of range, invalid characters etc.)
5	Guarded (Unable to setup in current status etc.)
6	Time Out
99	Firmware Updating
	(100s are Streaming Service related errors)
100	Access Error
101	Other Errors
102	Wrong User Name
103	Wrong Password
104	Account Expired
105	Account Disconnected/Gone Off/Shut Down
106	Account Number Reached to the Limit
107	Server Maintenance
108	Invalid Account
109	License Error
110	Read Only Mode
111	Max Stations
112	Access Denied
113	There is a need to specify the additional destination Playlist
114	There is a need to create a new Playlist
115	Simultaneous logins has reached the upper limit
	(200s are distribution related errors)
200	Linking in progress
201	Unlinking in progress

7. All ID List

Type	ID
Zone	main / zone2 / zone3 / zone4 ※Zone A is handles as "main", Zone B is handles as "zone2"
Input	cd / tuner / multi_ch / phono / hdmi1 / hdmi2 / hdmi3 / hdmi4 / hdmi5 / hdmi6 / hdmi7 / hdmi8 / hdmi / av1 / av2 / av3 / av4 / av5 / av6 / av7 / v_aux / aux1 / aux2 / aux / audio1 / audio2 / audio3 / audio4 / audio5 / audio_cd / audio / optical1 / optical2 / optical / coaxial1 / coaxial2 / coaxial / digital1 / digital2 / digital / line1 / line2 / line3 / line_cd / analog / tv / bd_dvd / usb_dac / usb / bluetooth / server / net_radio / napster / pandora / siriusxm / spotify / juke / airplay / radiko / qobuz / tidal / deezer / mc_link / main_sync / none
Sound Program	munich_a / munich_b / munich / frankfurt / stuttgart / vienna / amsterdam / usa_a / usa_b / tokyo / freiburg / royaumont / chamber / concert / village_gate / village_vanguard / warehouse_loft / cellar_club / jazz_club / roxy_theatre / bottom_line / arena / sports / action_game / roleplaying_game / game / music_video / music / recital_opera / pavilion / disco / standard / spectacle / sci-fi / adventure / drama / talk_show / tv_program / mono_movie / movie / enhanced / 2ch_stereo / 5ch_stereo / 7ch_stereo / 9ch_stereo / 11ch_stereo / stereo / surr_decoder / my_surround / target / straight / off

8. MusicCast Network Initial Setup

Initial Setup means to let a MusicCast Device join to MusicCast Network. It can be done only though MusicCast CONTROLLER app. To do so, download and install the free app of MusicCast CONTROLLER to mobile devices (smart phones, tablets)

➢ AppStore
(US)
https://itunes.apple.com/us/app/musiccast-controller-us/id1002730190
(Outside US)
https://itunes.apple.com/jp/app/musiccast-controller/id1012248381

GooglePlay
https://play.google.com/store/apps/details?id=com.yamaha.av.musiccastcontroller

1. Launch the app
   Choose "Setup" when it's launched. When you setup 2nd or more MusicCast devices, choose "Add New Device" in the setup menu of the app.
2. Turn on the power of MusicCast Device being setup.
3. Press and hold "CONNECT" button on a MusicCast Device for more than 5 seconds
4. Choose or type in Location Name
5. Choose or type in Room Name
6. Choose a Room Image
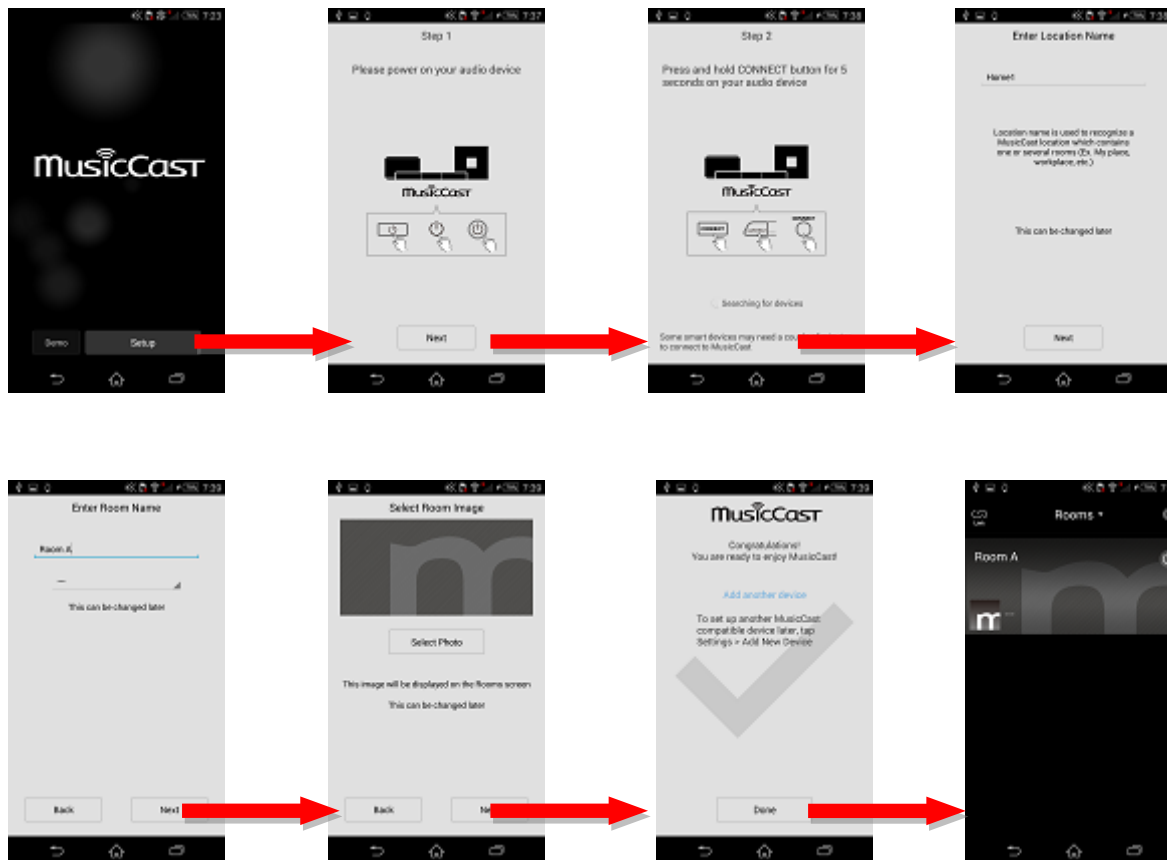
9. Application Notes

9.1. Link Function

Here explains how to use MusicCast Link function with YXC commands.

Assume that MusicCast Network initial setup is already done and MusicCast Network is established as following.
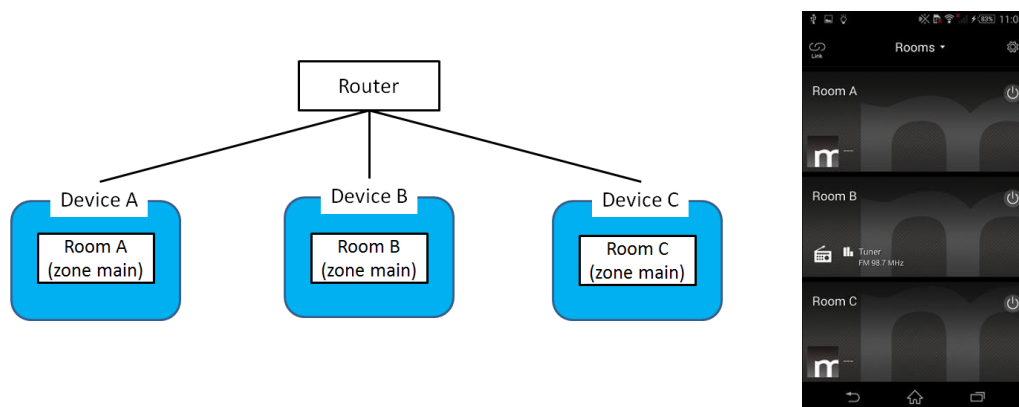
9.1.1. Compatibility check before creating group

Before creating group, check whether devices are compatible with each other.

1. Compare master's "compatible_client" with "version(major)" of client from distribution's data of getFeatures.

Master
Request (URI)
http://{Device B}/ YamahaExtendedControl/v1/system/getFeatures

Response
```
{
    :
  "distribution":{
    "version":3.10,
    "compatible_client":[
      2,     ← list of client version (major) compatible with master
      3
    ],
    :
  }
}
```

Client
Request (URI)
http://{Device A}/ YamahaExtendedControl/v1/system/getFeatures
http://{Device C}/ YamahaExtendedControl/v1/system/getFeatures

Response
```
{
     :
   "distribution":{
       "version":3.10,                    ← client version
       "compatible_client":[
         2,
         3
       ],
        :
   }
}
```

2. If the master's "compatible client" list contains "version" of the client(s), they are compatible and the distribution can be done. If not, they are incompatible and the distribution is not possible. Please update Device to the latest firmware.

3. If the "version" or the "compatible_client" of getFeatures > distribution's data does not exist in both the master and the client(s), it is possible to create a Group because the "version" is 1.xx thus they are compatible for the distribution.

9.1.2. Making a Group

Here shows how to make a Group. This is trying to establish a distribution network with Device B being a master, and with Device A and Device C being clients.

1. Create a random Group ID (16 byte).
2. Setup all clients with the Group ID and target Zone being used by "setClientInfo".

Request (URI)
http://{Device A}/ YamahaExtendedControl/v1/dist/setClientInfo

Request (json)
```
{
        "group_id":"9A237BF5AB80ED3C7251DFF49825CA42",
        "zone":[
            "main"
        ]
}
```
Response
```
{
        "response_code":0
}
```

Request (URI)
http://{Device C}/ YamahaExtendedControl/v1/dist/setClientInfo

Request (json)
```
{
        "group_id":"9A237BF5AB80ED3C7251DFF49825CA42",
```

```
        "zone":[
            "main"
        ]
}
```
Response
```
{
        "response_code":0
}
```

3. Setup the master with GroupID and client list by "setServerInfo (add)".
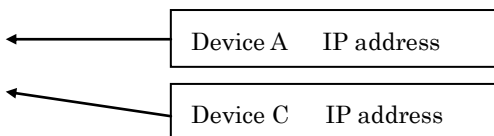
Request (URI)
http://{Device B}/ YamahaExtendedControl/v1/dist/setServerInfo

Request (json)
```
{
        "group_id":"9A237BF5AB80ED3C7251DFF49825CA42",
        "zone":"main",
        "type":"add",
        "client_list":[
            "192.168.0.5",          ← Device A    IP address
            "192.168.0.6"           ← Device C    IP address
        ]
}
```
Response
```
{
        "response_code":0
}
```

4. Start distribution with setting current all system distribution number to the master by "startDistribution".
   ＊When group creation is completed, the status of getDistrbutionInfo changes to "working".

Request (URI)
http://{Device B}/ YamahaExtendedControl/v1/dist/startDistribution?num=0

Response
```
{
        "response_code":0
}
```
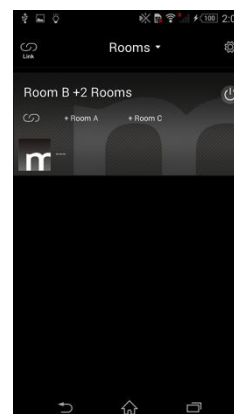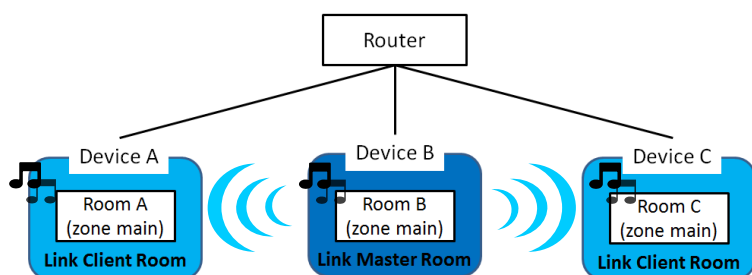
9.1.3. Removing a client from the Group
Here shows how to remove DeviceC from the Group

1. Clear GourpID and target Zone of the removing client(s) by "setClientInfo".
   *Set Group ID with empty text "".
   <u>Request (URI)</u>
   http://{Device C}/ YamahaExtendedControl/v1/dist/setClientInfo

   <u>Request (json)</u>
   ```
   {
           "group_id":"",
           "zone":[
               "main"
           ]
   }
   ```
   <u>Response</u>
   ```
   {
           "response_code":0
   }
   ```

2. Setup the master with GroupID and removing client list by "setServerInfo (remove)".
   *If all clients are to be remove, set empty text "" to GroupID in setServerInfo
   <u>Request (URI)</u>
   http://{Device B}/ YamahaExtendedControl/v1/dist/setServerInfo

   <u>Request (json)</u>
   ```
   {
           "group_id":"9A237BF5AB80ED3C7251DFF49825CA42",
           "zone":"main",
           "type":"remove",
           "client_list":[
               "192.168.0.6"          ◄──── Device C    IP address
           ]
   }
   ```
   <u>Response</u>
   ```
   {
           "response_code":0
   }
   ```

3. Reflect current all system distribution number to the master by "startDistribution".
   <u>Request (URI)</u>
   http://{Device B}/ YamahaExtendedControl/v1/dist/startDistribution?num=2

   <u>Response</u>
   ```
   {
            "response_code":0
   }
   ```
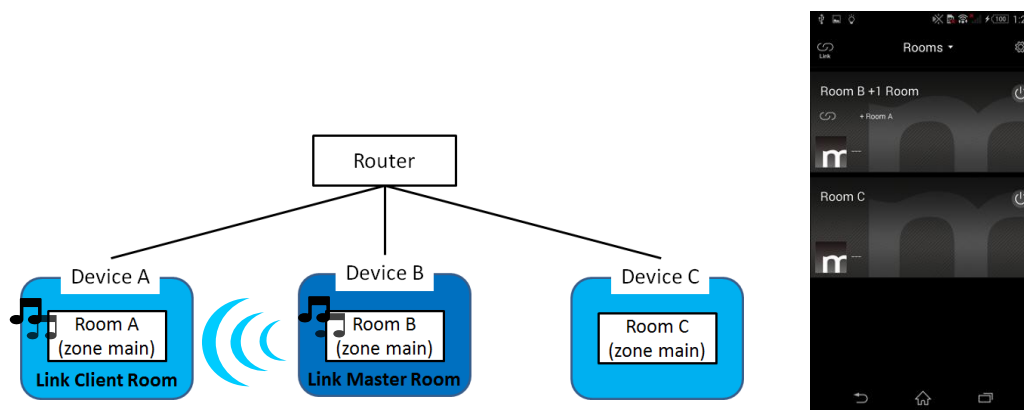
9.1.4. Adding a client to the Group

Here shows how to add DeviceC to the Group.

1. Setup GourpID and target Zone of the adding client(s) by "setClientInfo".
   <u>Request (URI)</u>
   http://{Device C}/ YamahaExtendedControl/v1/dist/setClientInfo

   <u>Request (json)</u>
   ```
   {
           "group_id":"9A237BF5AB80ED3C7251DFF49825CA42",
           "zone":[
               "main"
           ]
   }
   ```
   <u>Response</u>
   ```
   {
           "response_code":0
   }
   ```

2. Setup the master with GroupID and adding client list by "setServerInfo (add)".
   <u>Request (URI)</u>
   http://{Device B}/ YamahaExtendedControl/v1/dist/setServerInfo

   <u>Request (json)</u>
   ```
   {
           "group_id":"9A237BF5AB80ED3C7251DFF49825CA42",
           "zone":"main",
           "type":"add",
           "client_list":[
               "192.168.0.6"          ◄———— Device C    IP address
           ]
   }
   ```
   <u>Response</u>
   ```
   {
           "response_code":0
   }
   ```

3. Reflect all system distribution number to the master by "startDistribution".
   Request (URI)
   http://{Device B}/ YamahaExtendedControl/v1/dist/startDistribution?num=1
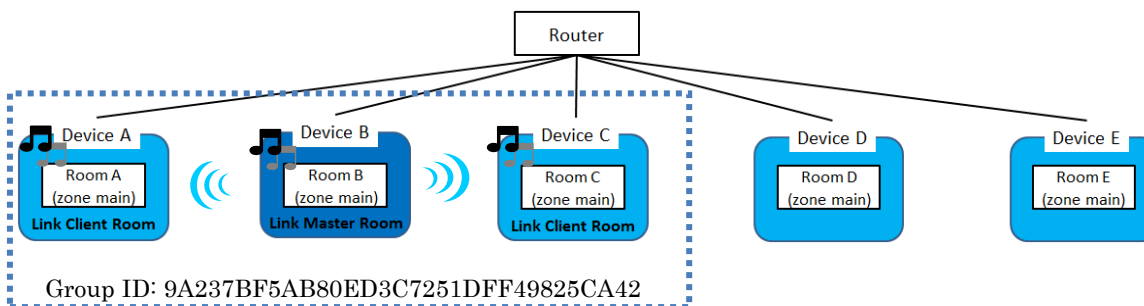
   Response
   {
           "response_code":0
   }
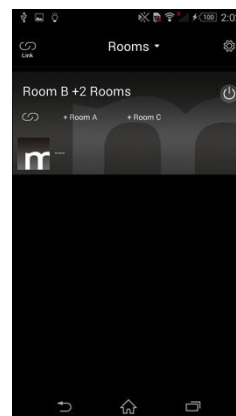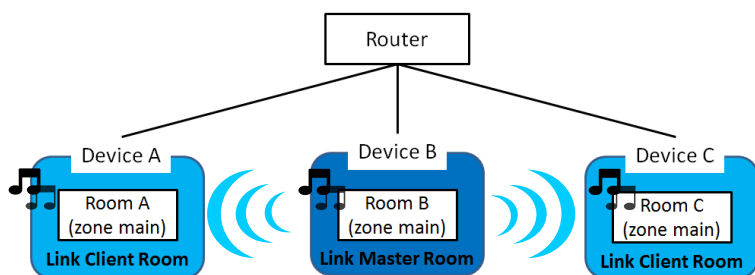
9.1.5. Making multi Groups

Here shows how to make multi Groups.
Assume that MusicCast Network initial setup is already done and MusicCast Network is established as following.
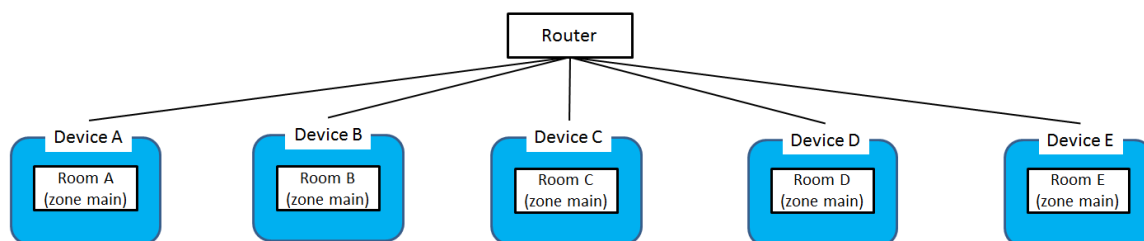
Follow 13.2.1(Making a Group), establish a distribution network with Device B being a master, and with Device A and Device C being clients.

Next is trying to establish a distribution network with Device D being a master, and with Device E being clients.

1. Create a random Group ID (16 byte).
   ＊Note not set the same Group ID

2. Setup all clients with the Group ID and target Zone being used by "setClientInfo".

   Request (URI)
   http://{Device E}/ YamahaExtendedControl/v1/dist/setClientInfo

   Request (json)
   ```
   {
           "group_id":"7B335AE4C12345677251DAA466669B40",
           "zone":[
               "main"
           ]
   }
   ```
   Response
   ```
   {
           "response_code":0
   }
   ```

3. Setup the master with GroupID and client list by "setServerInfo (add)".

   Request (URI)
   http://{Device D}/ YamahaExtendedControl/v1/dist/setServerInfo

   Request (json)
   ```
   {
           "group_id":"7B335AE4C12345677251DAA466669B40",
           "zone":"main",
           "type":"add",
           "client_list":[
               "192.168.0.8"    ←——————— Device E    IP address
           ]
   }
   ```
   Response
   ```
   {
           "response_code":0
   }
   ```

4. Start distribution with setting current all system distribution number to the master by "startDistribution".Already establish a distribution network with Device B being a master, and with Device A and Device C being clients, need to set num = 2.

   Request (URI)
   http://{Device B}/ YamahaExtendedControl/v1/dist/startDistribution?num=2

<u>Response</u>
```
{
        "response_code":0
}
```
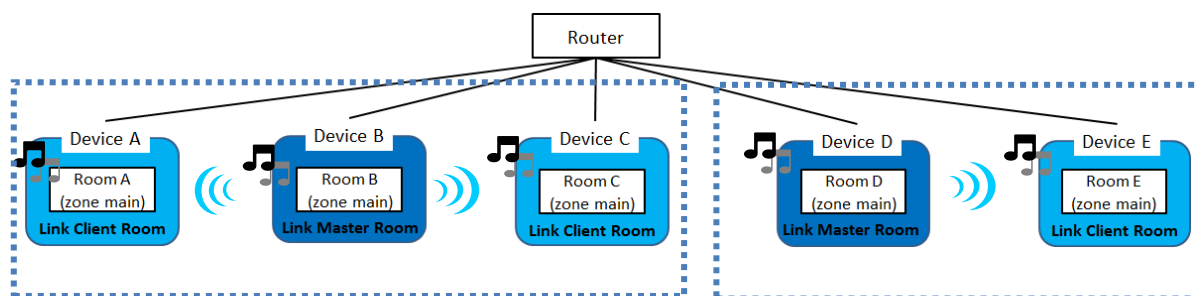

9.1.6. External causes/interruptions making a break of a Group

If things shown below happen, it is necessary for a Group to be properly handled as mentioned below (mostly by removing client(s) from the master. Refer to 13.2.2, procedure #2);

1. In case a Device's input is selected to others than MusicCast Link
   - If a Device is the master, nothing matters
   - If a Deice is a client, it needs to be removed from the Group, meaning:
     ・Remove the client from the master
     ・Clear the Group ID in the client
2. In case a Device is assigned to the other Group for some reason
   - If a Device is the master, all client(s) assigned to it need to be removed from the Group
   - If a Device is a client, it needs to be removed from the master
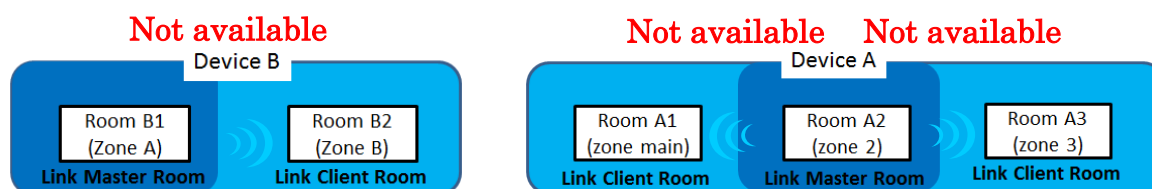
9.1.7. Note to making Group
   1. In case of Multi Zone Receiver, each Zone is unable to join to another group discretely.
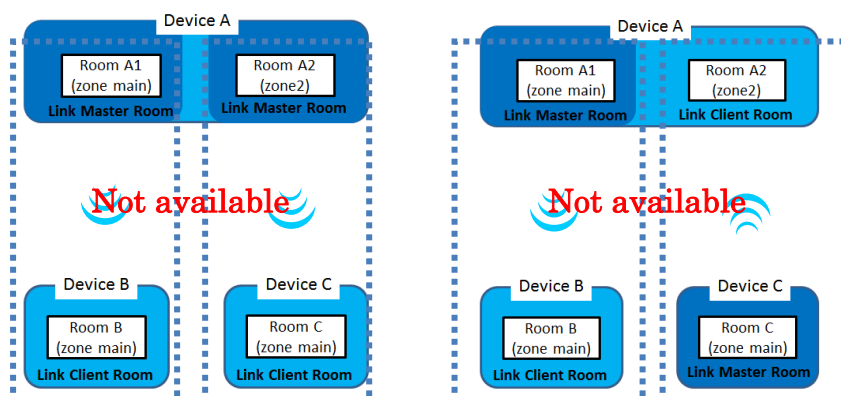
2. Zone A or B independently cannot join as a client (always Zone A AND B should join as clients at the same time).
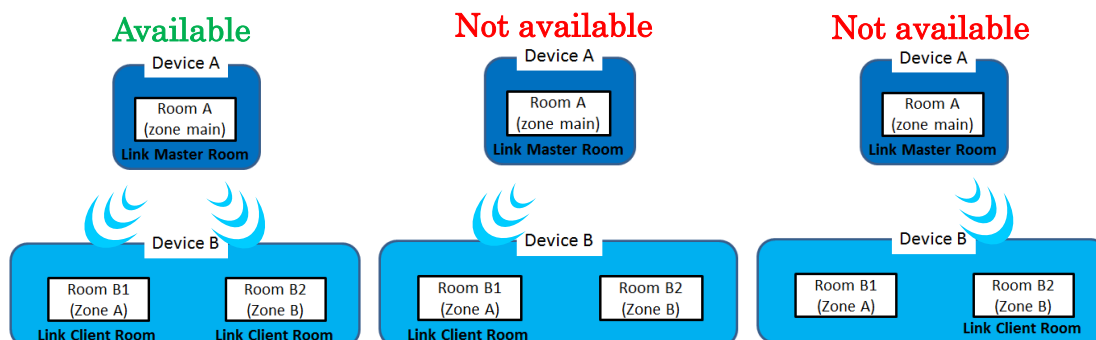
3. Each Zone is unable to be set as a Master or a Client in the same group.

Available to make pseudo-Group that Main Zone is Master and Zone2 or Zone3 equipped Main Zone Sync feature is Client. In this case, no network distribution among Master and Client.

4. Group can be made between devices that have different Location IDs. The max number of MusicCast Devices within one Group is up to the number specified by get_max of getFeatures +1. (this is valid with API Version 2.00 or later. The maximum number before Version 2.00 was limited up to 10).

5. Devices not included in the group may respond "Client". Please refer to the information below and control.

- If the device is not included in the group, if any Input of the Zone is MusicCast Link, the information of "Role" acquired by getDistributionInfo is "Client".

- If the device is not included in the group, the Group ID is "empty text". Please use this information to determine whether it is included in the group or not included in the group.

It is basically a specification that can create multiple Groups within the same Location. When creating a group, you can not display accurate information unless you use both the Role and Group ID information.

9.1.8. Cautions when creating Group on MusicCast devices before 2017 and after 2018
  1. There are combinations that can not create a group depending on device generation and firmware version.

- Devices before 2017 must be updated to the latest firmware that is compatible with the devices after 2018, or you cannot create Groups among devices after 2018 and before 2017.

- Even among devices before 2017 only, you cannot create a group with devices that mix up with the latest firmware and the old firmware.

		Client		
		2015-2017 devices (old firmware)	2015-2017 devices (latest firmware)	2018- devices
Master	2015-2017 devices (old firmware)	✓	N/A	N/A
	2015-2017 devices (latest firmware)	N/A	✓	✓
	2018- devices	N/A	✓	✓

A list of group creation permission / rejection based on combination of devices by generation

Supplement (distinction)
- 2015-2017 devices : getDeviceInfo > netmodule_generation ＝ 1, or no information
- 2018- devices : getDeviceInfo > netmodule_generation ＝ 2
- old firmware : getFeatures > distribution - version ＝ no information
- latest firmware : getFeatures > distribution - version ＝ after 2.xx

2. To determine the device before 2017 and the device after 2018, please refer to the netmodule_generation of getDeviceInfo. However, if the device before 2017 is not the latest firmware, it has no information of netmodule_generation.

- Compare the data of "netmodule_generation" of getDeviceInfo of master and client respectively.

Master
Request (URI)
http://{Device B}/ YamahaExtendedControl/v1/system/getDeviceInfo

Response
{
  :
  "netmodule_generation":**2**,   ←——————— generation of master
  :
}

]

Client
Request (URI)
http://{Device A}/ YamahaExtendedControl/v1/system/getDeviceInfo
http://{Device C}/ YamahaExtendedControl/v1/system/getDeviceInfo

Response
{
  :
  "netmodule_generation":**1**,   ←   [generation of client]
  :
}

- If the "netmodule_generation" of the master and the client(s) all on 2 or more, you can create a Group. If not, check "9.1.1. Compatibility check before creating group" above for compatibility between the devices.

- If there is no "netmodule_generation" information of master and client, or if there is no compatibility among devices, please update the devices to the latest firmware.

3. If you set the device of after 2018 to the master of distribution, it may take some time for group creation to complete by settings. Inconsistency will occur if you do other operations while creating Group, please refer to the following and make appropriate control.

- It takes time to create a group if the setting of "Link control" in getStatus of Zone is "standard" or "stability".

  ・ Time varies depending on the number of units and settings. Maximum 2 minutes to 3 minutes.
  ・ When you can disable operation with Controller's UI until Group creation is completed, please prohibit other operations by graying out the button until status of getDistributionInfo becomes "working".
  ・ When you can not disable operation with Controller UI until Group creation is completed, we recommend changing the Link control setting to "speed". However, "speed" assumes that the network setting of the device is used for wired connection.
  ・ You cannot cancel creating a Group while it is being processed.

Request (URI)
http://{Device A}/ YamahaExtendedControl/v1/dist/getDistributionInfo

Response
{
  :
  "status":" working ",   ←   [Status of creatuing group]
  :
}

9.1.9. Group Volume

How to control the group volume when linking with MusicCast-Link.

1. When the group volume operation is started, temporarily store the volume value of all the rooms.

2. When changing the volume of "LinkServer" device, change the volume of "LinkClient" at the same rate as the change ratio .

3. If the change amount of the volume set for LinkClient is less than 1, setClientVolume is used to set the change amount to less than 1.
   (Since the volume length is different for each device, it may happen if the LinkMaster range is large and the LinkClient range is small.)

9.2. Event Process

Events are spread out as UDP unicast. Retrieve necessary information by polling because of UDP packet loss.
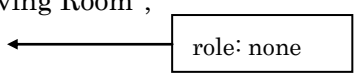(Ex) getDistributionInfo

After send getDistributionInfo, infrequently receive below response.

<u>Request (URI)</u>
http://{Device A}/ YamahaExtendedControl/v1/dist/getDistributionInfo

<u>Response</u>
```
{
    "response_code":0,
    "group_id":"9A237BF5AB80ED3C7251DFF49825CA42",
    "group_name":"[Link] Living Room",
    "role":"none",                         ← role: none
    "status":" working ",
    "server_zone":"main",
    "client_list":[
      {
        "ip_address":"192.168.0.5",
        "data_type":"base"
      },
      {
        "ip_address":"192.168.0.11",
        "data_type":"base"
      },
      {
        "ip_address":"192.168.0.22,
        "data_type":"ext"
      }
    ]
    "build_disable":[
      {
        "role":"client",
```

```
        "reasons":[
            "unknown"
        ]
    }
]
    "audio_dropout":false,
}
```

When group_id isn't 00000000000000000000000000000000 and client_list isn't empty, process "role" is "server" , not "none".

9.3. Stereo Pair

Establish Stereo Pair setting by operating Devices. Stereo Pair setting is established as following.

Comfirm Stereo Pair status.

<u>Request</u>

http://{DeviceA}/ YamahaExtendedControl/v1/system/getStereoPairInfo

<u>Response</u>
```
{
        "response_code":0,
        "status":"master_left",
        "pair_info":{
            "alive":true,
            "ip_address":"192.168.0.11",
            "mac_address":"00A0DE012345"
        }
}
```

<u>Request</u>

http://{DeviceB}/ YamahaExtendedControl/v1/system/getStereoPairInfo
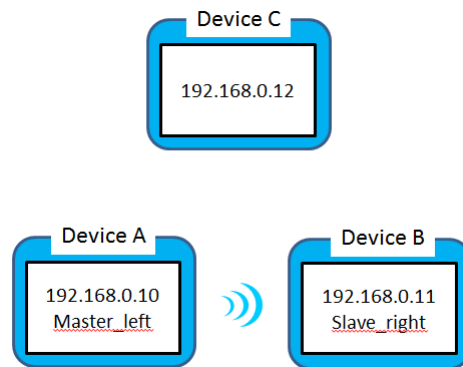
<u>Response</u>
```
{
        "response_code":0,
        "status":"slave_right",
        "pair_info":{
            "alive":true,
            "ip_address":"192.168.0.10",
            "mac_address":"00A0DE012346"
        }
}
```

This is trying to establish a distribution network with Device C being a server, and with Stereo Pair devices being clients.

1. Setup client with the Group ID and target Zone being used by "setClientInfo".

   <u>Request (URI)</u>
   http://{Device A}/ YamahaExtendedControl/v1/dist/setClientInfo
   <u>Request (json)</u>
   ```
   {
           "group_id":"9A237BF5AB80ED3C7251DFF49825CA42",
           "zone":[
               "main"
           ]
           " server_ip_address ":"192.168.0.12"          ← Need server IP address
   }
   ```
   <u>Response</u>
   ```
   {
           "response_code":0
   }
   ```

2. Setup the server with GroupID and client list by "setServerInfo (add)".

   <u>Request (URI)</u>
   http://{Device C}/ YamahaExtendedControl/v1/dist/setServerInfo

   <u>Request (json)</u>
   ```
   {
           "group_id":"9A237BF5AB80ED3C7251DFF49825CA42",
           "zone":"main",
           "type":"add",
           "client_list":[
               "192.168.0.10"
           ]
   }
   ```
   <u>Response</u>
   ```
   {
           "response_code":0
   }
   ```
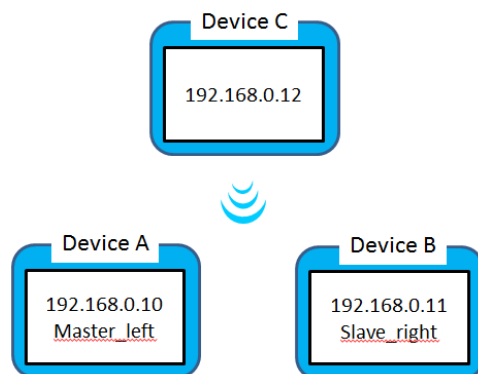
3. Start distribution with num(more than 3) to the server by "startDistribution"

Request (URI)
http://{Device C}/ YamahaExtendedControl/v1/dist/startDistribution?num=4
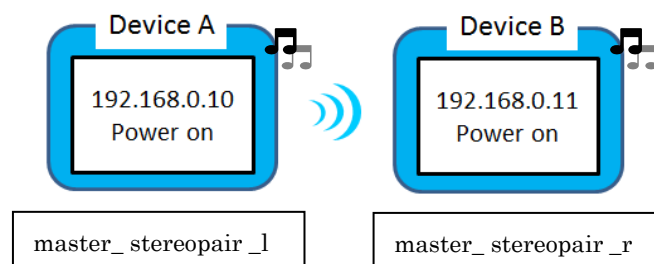
Response
{
    "response_code":0
}

9.4. MusicCast Surround/Stereo

Please build "MusicCast surround / stereo" by operating the special application "MusicCast CONTROLLER".

1. On restrictions after "MusicCast surround / stereo" construction
   In Slave's device, Input becomes MC link fixed, and each function (Favorite, alarm, etc.) can not be used.
   The setting of Link Control (Speed Boost etc.) is prohibited.
2. About how to count MusicCast surround / stereo device at linking.
   When making the MusicCast surround/stereo device the master of the link, reduce the number of Slave of MusicCast surround/stereo device from the maximum number.
   You do not need to do anything when you make a MusicCast surround/stereo device a client of a link. Because MusicCast surround/stereo device Master redistributes to Slave.
3. About troubleshooting
   Please send resetMcSurround to each Master / Slave when an error occurs.

9.4.1. MusicCast Stereo Pair

Suppose the following MusicCast Stereo Pair state is configured.

Check the Stereo Pair status of the device.

<u>Request</u>
http://{DeviceA}/ YamahaExtendedControl/v1/dist/getDistributionInfo

<u>Response</u>
```
{
        "mc_surround":{
        "id":123456789,
        "role":" master_stereopair_l ",
        "status":"working":
        "slave_info":[
              {
                 "role":"slave_ stereopair _r",
                 "ip_address":"192.168.0.11",
                 "mac_address":"00A0DE000002"
              },
           ],
        }
}
```

<u>Request</u>
http://{DeviceB}/ YamahaExtendedControl/v1/dist/getDistributionInfo

<u>Response</u>
```
{
        "mc_surround":{
        "id":123456789,
        "role":" slave_stereopair_r ",
        "status":"working":
        " master_info":[
              {
                 "role":"master_ stereopair _l",
                 "ip_address":"192.168.0.10",
                 "mac_address":"00A0DE000001"
              },
           ],
        }
}
```

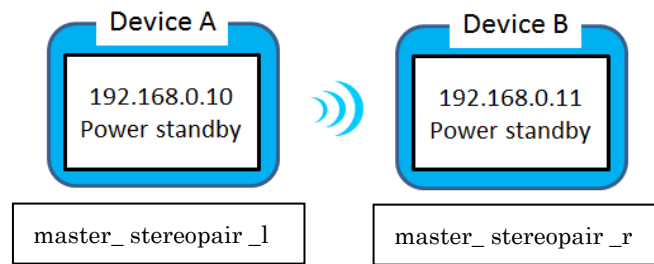Please control power supply, Volume etc only for master device. The slave device is linked to the master device.

<u>Request</u>
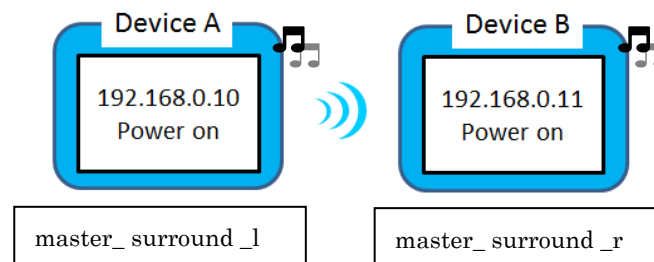http://{DeviceA}/ YamahaExtendedControl/v1/main/setPower?power=standby

<u>Response</u>
```
{
        "response_code":0,
```

}

9.4.2. MusicCast Surround Pair

Suppose the following MusicCast Surround Pair state is configured.

Check the Surround Pair status of the device.

<u>Request</u>
http://{DeviceA}/ YamahaExtendedControl/v1/dist/getDistributionInfo

<u>Response</u>
```
{
        "mc_surround":{
        "id":123456789,
        "role":" master_multi_main ",
        "status":"working":
        "slave_info":[
            {
               "role":" slave_surround_lr ",
               "ip_address":"192.168.0.11",
               "mac_address":"00A0DE000002"
            },
          ],
        }
}
```

<u>Request</u>
http://{DeviceB}/ YamahaExtendedControl/v1/dist/getDistributionInfo

<u>Response</u>
```
{
        "mc_surround":{
        "id":123456789,
```

```
      "role":" slave_surround_lr ",
      "status":"working":
      " master_info":[
            {
                "role":" master_multi_main ",
                "ip_address":"192.168.0.10",
                "mac_address":"00A0DE000001"
            },
         ],
      }
}
```

Please control power supply, Volume etc only for master device. The slave device is linked to the master device.
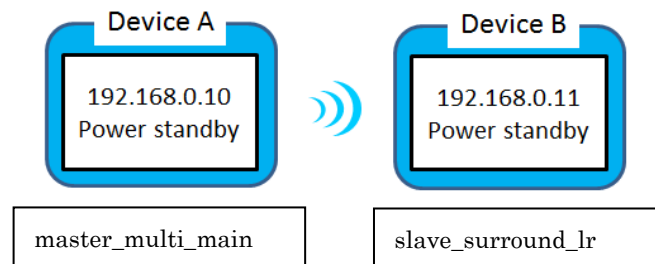
Request
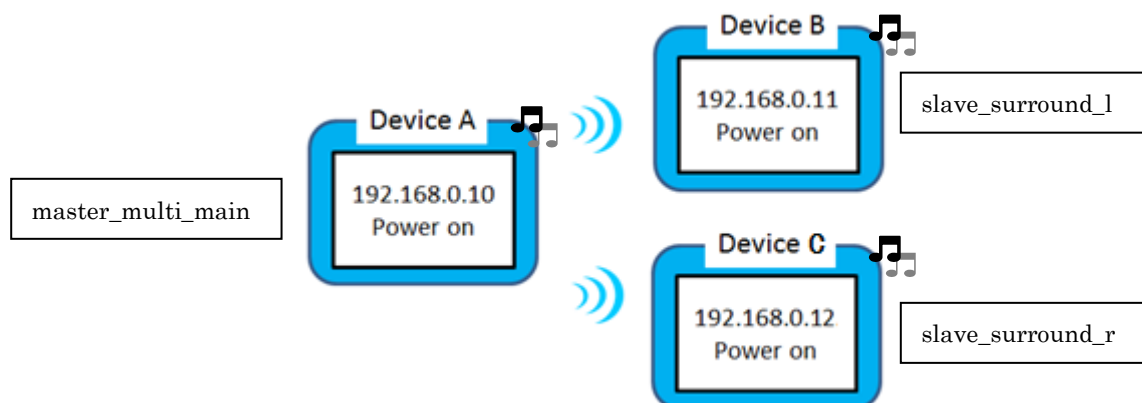http://{DeviceA}/ YamahaExtendedControl/v1/main/setPower?power=standby

Response
```
{
      "response_code":0,
}
```


Suppose the following MusicCast Surround Pair state is configured.

Check the Surround Pair status of the device.

Request
http://{DeviceA}/ YamahaExtendedControl/v1/dist/getDistributionInfo

Response
```
{
        "mc_surround":{
        "id":123456789,
        "role":" master_multi_main ",
        "status":"working":
        "slave_info":[
             {
                "role":" slave_surround_l ",
                "ip_address":"192.168.0.11",
                "mac_address":"00A0DE000002"
             },
             {
                "role":" slave_surround_r ",
                "ip_address":"192.168.0.12",
                "mac_address":"00A0DE000003"
             },
          ],
        }
}
```

Request
http://{DeviceB}/ YamahaExtendedControl/v1/dist/getDistributionInfo

Response
```
{
        "mc_surround":{
        "id":123456789,
        "role":" slave_surround_l ",
        "status":"working":
        " master_info":[
             {
                "role":" master_multi_main ",
                "ip_address":"192.168.0.10",
                "mac_address":"00A0DE000001"
             },
          ],
        }
}
```

Request
http://{DeviceC}/ YamahaExtendedControl/v1/dist/getDistributionInfo

Response
```
{
        "mc_surround":{
        "id":123456789,
        "role":" slave_surround_r ",
        "status":"working":
        " master_info":[
            {
              "role":" master_multi_main ",
              "ip_address":"192.168.0.10",
              "mac_address":"00A0DE000001"
            },
          ],
        }
}
```
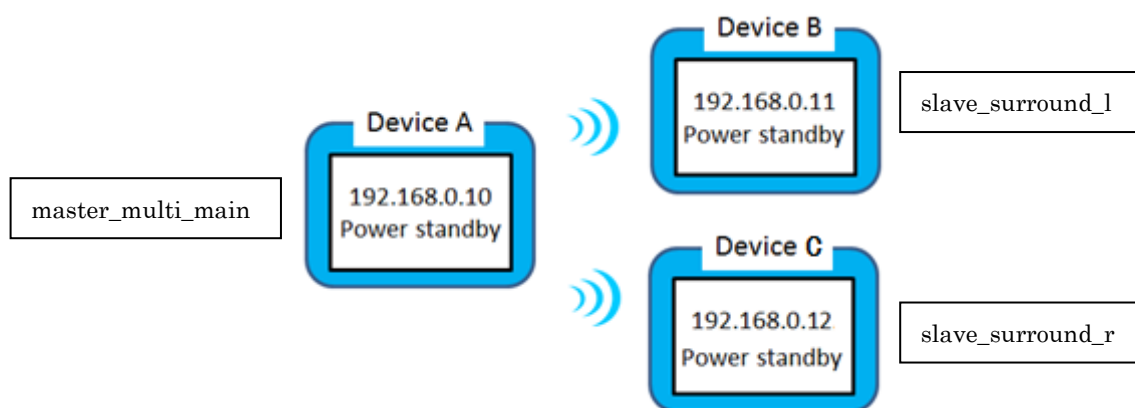
Please control power supply, Volume etc only for master device. The slave device is linked to the master device.

Request
http://{DeviceA}/ YamahaExtendedControl/v1/main/setPower?power=standby
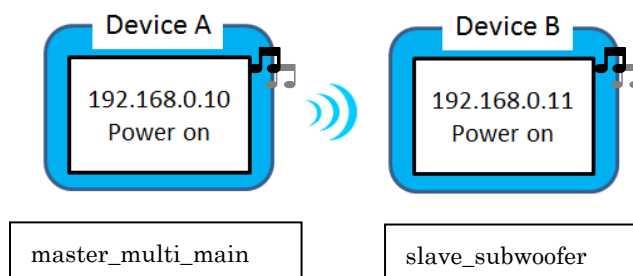
Response
```
{
        "response_code":0,
}
```


9.4.3. MusicCast Subwoofer Pair

Suppose the following MusicCast Subwoofer Pair state is configured.

Check the Subwoofer Pair status of the device.

<u>Request</u>
http://{DeviceA}/ YamahaExtendedControl/v1/dist/getDistributionInfo

<u>Response</u>
```
{
        "mc_surround":{
        "id":123456789,
        "role":" master_multi_main ",
        "status":"working":
        "slave_info":[
             {
                "role":" slave_subwoofer ",
                "ip_address":"192.168.0.11",
                "mac_address":"00A0DE000002"
             },
           ],
        }
}
```

<u>Request</u>
http://{DeviceB}/ YamahaExtendedControl/v1/dist/getDistributionInfo

<u>Response</u>
```
{
        "mc_surround":{
        "id":123456789,
        "role":" slave_subwoofer ",
        "status":"working":
        " master_info":[
             {
                "role":" master_multi_main ",
                "ip_address":"192.168.0.10",
                "mac_address":"00A0DE000001"
             },
           ],
        }
}
```
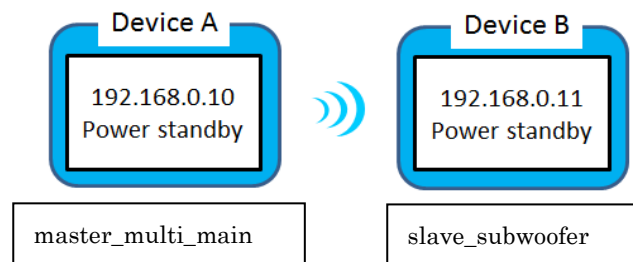
Please control power supply, Volume etc only for master device. The slave device is linked to the master device.

<u>Request</u>
http://{DeviceA}/ YamahaExtendedControl/v1/main/setPower?power=standby

Response
```
{
        "response_code":0,
}
```

Revision History

Rev.	Date	Contents	Author
1.00	9/28/2016	First edition	Eto
1.10	1/26/2017	Add note: stereo pair, event process	Eto
1.20	11/7/2017	Add note: group volume, note to making group	Fukuyama
2.00	5/11/2018	Expand control function. (Refer Revision_Modification.pdf for more detail)	Fukuyama
2.10	8/9/2018	Expand control function. (Refer Revision_Modification.pdf for more detail)	Fukuyama