

# Yamaha Extended Control API Specification (Basic)

Rev. 1.00

---

## Contents

1. Preface .....	4
2. Definitions .....	4
3. API Overview .....	4
4. System .....	6
4.1. getDeviceInfo .....	6
4.2. getFeatures .....	7
4.3. getNetworkStatus .....	13
4.4. getFuncStatus .....	14
4.5. setAutoPowerStandby .....	15
4.6. getLocationInfo .....	16
4.7. sendIrCode .....	16
5. Zone .....	17
5.1. getStatus .....	17
5.2. getSoundProgramList .....	19
5.3. setPower .....	20
5.4. setSleep .....	20
5.5. setVolume .....	21
5.6. setMute .....	21
5.7. setInput .....	22
5.8. setSoundProgram .....	22
5.9. prepareInputChange .....	23
6. Tuner .....	23
6.1. getPresetInfo .....	24
6.2. getPlayInfo .....	25
6.3. setFreq .....	27
6.4. recallPreset .....	28
6.5. switchPreset .....	28
6.6. storePreset .....	29
6.7. setDabService .....	29
7. Network/USB .....	30
7.1. getPresetInfo .....	30
7.2. getPlayInfo .....	31
7.3. setPlayback .....	33
7.4. toggleRepeat .....	34
7.5. toggleShuffle .....	34
7.6. getListInfo .....	35
7.7. setListControl .....	37
7.8. setSearchString .....	38
7.9. recallPreset .....	39
7.10. storePreset .....	39
7.11. getAccountStatus .....	40
7.12. switchAccount .....	41
7.13. getServiceInfo .....	42
8. CD .....	44
8.1. getPlayInfo .....	44
8.2. setPlayback .....	45
8.3. toggleTray .....	46

---

8.4.	toggleRepeat.....	46
8.5.	toggleShuffle .....	47
9.	Response Code List.....	47
10.	Events.....	48
10.1.	Outline.....	48
10.2.	Conditions .....	48
10.3.	Details of event data.....	48
11.	All ID List.....	52
12.	Application Notes .....	53
12.1.	Example of List Control .....	53
12.1.1.	Preparation before start browsing .....	53
12.1.2.	Retrieving list information (In case max line number is less than or equal to 8) ....	53
12.1.3.	Layer change.....	54
12.1.4.	Retrieving list information (In case max line number is more than 8) .....	55
12.1.5.	Playback a file.....	57
12.1.6.	Moving back one layer .....	57
12.2.	Device Search.....	58

## 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 basic specifications including the way to control power, change input, browse list, retrieve album art. Advanced function including MusicCast link function and others is described in Yamaha Extended Control API Specification (Advanced) will be released on Oct 2016.

## 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 10
- Link  
A function to share an audio stream among rooms
- Device  
Equal to MusicCast Device
- Room  
A minimum component MusicCast devices can be shown/associated within a MusicCast network

## 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. System

### 4.1. getDeviceInfo

For retrieving basic information of a Device

#### Request Parameters

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

#### Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to <a href="#">"Response Code List"</a> for details
model_name	string	Returns model name Ex. "RX-V479" / "WXC-50" / "UNKNOWN"
destination	string	Returns Device's destination region code. There are cases of multiple regions codes like "UC" "BG" depending on Devices Values: "J" / "U" / "C" / "R" / "BG" / "T" / "A" / "L" / "K"
device_id	string	Return Device's ID (12 digit ASCII).Device ID is unique ID to identify Device. Note: Available on and after API Version 1.17
system_version	floating point	Returns System Version
api_version	floating point	Returns API Version
netmodule_version	string	Returns Network Module version
netmodule_checksum	string	Return check-sum of Network Module in hex. "00000000" is returned until check-sum calculation is done
operation_mode	string	Reserved
update_error_code	string	Reserved
update_progress	object	Reserved

Example Response	<pre>{   "response_code":0,   "model_name":"RX-V679",   "destination":"UC",   "system_id":"ABADCAFE",   "system_version":1.10,   "api_version":1.00,   "netmodule_version":"0200",   "netmodule_checksum":"878059DD",   "operation_mode":"normal",   "update_error_code"."00000000" }</pre>
------------------	---

## 4.2. getFeatures

For retrieving feature information equipped with a Device

### Request Parameters

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

### Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to <a href="#">"Response Code List"</a> for details
system	object	Returns system's overall info
- func_list	string array	Returns valid functions Values: "wired_lan" / "wireless_lan" / "wireless_direct" / "network_standby" / "network_standby_auto" / "bluetooth_standby" / "bluetooth_tx_setting" / "auto_power_standby" / "ir_sensor" / "speaker_a" / "speaker_b" / "headphone" / "dimmer" / "zone_b_volume_sync" / "airplay" / "stereo_pair" / "speaker_settings"
- zone_num	integer	Returns Zone numbers. Zone B is treated as Zone2 in YXC so a Device with ZoneB returns 2. A Device without Zones returns 1
- input_list	array	Returns a list of inputs
- id	string	Returns Input ID Refer to <a href="#">"All ID List"</a> for details
- distribution_enable	boolean	Returns whether an input can be a source of Link distribution
- rename_enable	boolean	Returns whether an input can be renamed
- account_enable	boolean	Return whether an input comes with an account info
- play_info_type	string	Returns a type of playback info. Depending on this type, use specific API to retrieve appropriate playback info Values: "none" / "tuner" / "netusb" / "cd"
- range_step	array	Returns minimum/maximum/step values of a parameter
- id	string	Returns an ID Value: "dimmer"
- min	integer	Returns a minimum value of a parameter
- max	integer	Returns a maximum value of a parameter
- step	integer	Returns a step value of a parameter
- speaker_settings	object	Reserved
- ymap_list	string array	Reserved
- web_control_url	string	Reserved
zone	array	Returns Zone related information
- id	string	Returns Zone ID. Refer to <a href="#">"All ID List"</a> for details

- zone_b	boolean	Returns whether the target Zone is Zone B or not. Valid only when Zone ID is "zone2"
- func_list	string array	Returns a list of valid functions Values: "power" / "sleep" / "volume" / "mute" / "sound_program" / "direct" / "pure_direct" / "enhancer" / "tone_control" / "equalizer" / "balance" / "dialogue_level" / "dialogue_lift" / "clear_voice" / "subwoofer_volume" / "bass_extension" / "signal_info" / "prepare_input_change" / "link_control" / "link_audio_delay"
- input_list	string array	Returns a list of selectable Input IDs Refer to " <a href="#">All ID List</a> " for details
- sound_program_list	string array	Returns a list of selectable Sound Program IDs Refer to " <a href="#">All ID List</a> " for details
- tone_control_mode_list	string array	Returns selectable settings of Tone Control Mode. If there's no list of this, it's fixed to "manual" Values: "manual" / "auto" / "bypass"
- equalizer_mode_list	string array	Returns selectable settings of Equalizer Mode. If there's not list of this, it's fixed to "manual" Values: "manual" / "auto" / "bypass"
- link_control_list	string array	Returns selectable settings of Link Control Values: "standard" / "stability" / "speed"
- link_audio_delay_list	string array	Returns selectable settings of Link Audio Delay Values: "lip_sync"/"audio_sync" / "audio_sync_on" / "audio_sync_off"
- range_step	array	Returns ID, min, max, step values of a parameter
- id	string	Returns ID of a parameter Values: "volume" / "tone_control" / "equalizer" / "balance" / "dialogue_level" / "dialogue_lift" / "subwoofer_volume"
- min	integer	Returns minimum value of a parameter
- max	integer	Returns maximum value of a parameter
- step	integer	Returns step value of a parameter
tuner	object	Returns information related to Tuner function
- func_list	string array	Returns a list of available functions Values: "am" / "fm" / "rds" / "dab" / "hd_radio" / "fm_auto_preset" / "dab_initial_scan" / "dab_tune_aid"
- range_step	array	Returns object ID, min, max, step values of a parameter
- id	string	Returns object ID of a parameter Values: "am" / "fm"
- min	integer	Returns minimum value (in unit of kHz)
- max	integer	Returns maximum value (in unit of kHz)
- step	integer	Returns step value (in unit of kHz)
- preset	object	Returns information related to Preset
- type	string	Returns Preset type. The value of "common" is for each band. Values: "common" (band common) / "separate" (each band preset)
- num	integer	Returns preset capable number. If type is "separate", each band of AM/FM/DAB can have presets up to this number
netusb	object	Returns information related to Net/USB function
- func_list	string array	Returns a list of available functions Value: "recent_info"
- preset	object	Returns information related to Preset



- num	integer	Returns preset capable number. All sources of Net/USB Input share this presets
- recent_info	object	Returns information related to playback history
- num	integer	Returns number of memorable playback history. Playback history is shared among all Net/USB Input sources
- vtuner_fver	string	Reserved
distribution	object	Returns information related to distribution function
- server_zone_list	string array	Returns a list of Zone ID that can work as a master of distributing server. If a Device doesn't have anything of this parameter, "main" is the one of Zone ID to be able to work as a master Values: "main" / "zone2" / "zone3" / "zone4"
clock	object	Reserved

Example Response	<pre>{   "response_code":0,   "system":{     "func_list":[       "wired_lan",       "wireless_lan",       "wireless_direct",       "network_standby",       "bluetooth_standby",       "speaker_settings"     ],     "zone_num":3,     "input_list":[       {         "id":"hdmi1",         "distribution_enable":false,         "rename_enable":true,         "account_enable":false       },       {         "id":"hdmi2",         "distribution_enable":false,         "rename_enable":true,         "account_enable":false       },       ...       {         "id":"spotify",         "distribution_enable":true,         "rename_enable":false,         "account_enable":false       }     ],     "speaker_settings":{       "type":"beam",       "common":{</pre>
------------------	---

```

    "attribute":1,
    "range_step":[
      {
        "id":"target_angle",
        "min":-90,
        "max":90,
        "step":1
      }
    ]
  },
  "front_l":{
    "attribute":7,
    "range_step":[
      {
        "id":"level",
        "min":-10,
        "max":10,
        "step":1
      },
      {
        "id":"h_angle",
        "min":-90,
        "max":90,
        "step":1
      }
    ]
  },
  ...
  "swfr":{
    "attribute":1,
    "range_step":[
      {
        "id":"level",
        "min":-10,
        "max":10,
        "step":1
      }
    ]
  }
},
"zone":[
  {
    "id":"main",
    "func_list":[
      "power",
      "sleep",
      "volume",
      "mute",

```

```

        "sound_program",
        "pure_direct",
        "enhancer",
        "tone_control",
        "prepare_input_change",
        "link_control"
    ],
    "input_list":[
        "hdmi1",
        "hdmi2",
        ...
        "spotify"
    ],
    "sound_program_list":[
        "munich",
        "vienna",
        ...
        "straight"
    ],
    "link_control_list":[
        "standard",
        "stability"
    ],
    "range_step":[
        {
            "id":"volume",
            "min":0,
            "max":194,
            "step":1
        },
        {
            "id":"tone_control",
            "min":-12,
            "max":12,
            "step":1
        }
    ]
},
{
    "id":"zone2",
    "zone_b":false,
    "func_list":[
        "power",
        "sleep",
        "volume",
        "mute",
        "tone_control",
        "prepare_input_change"
    ],

```

```

    "input_list":[
        "hdmi1",
        "hdmi2",
        ...
        "spotify"
    ],
    "range_step":[
        {
            "id":"volume",
            "min":0,
            "max":194,
            "step":1
        },
        {
            "id":"tone_control",
            "min":-5,
            "max":5,
            "step":1
        }
    ]
},
"tuner":{
    "func_list":[
        "am",
        "fm",
        "rds",
        "dab"
    ],
    "range_step":[
        {
            "id":"am",
            "min":531,
            "max":1611,
            "step":9
        },
        {
            "id":"fm",
            "min":76000,
            "max":90000,
            "step":100
        }
    ],
    "preset":{
        "type":"common",
        "num":40
    }
},
"netusb":{

```

	<pre> "preset":{   "num":40 } "vtuner_fver":"A" } } </pre>
--	--

### 4.3. getNetworkStatus

For retrieving network related setup / information

#### Request Parameters

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

#### Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details
network_name	string	Returns Network Name (Friendly Name)
connection	string	Returns setup status of Connection Values: "wired_lan" / "wireless_lan" / "wireless_direct" / "extend_1" / "extend_2" / "extend_3" / "unknown"
dhcp	boolean	Returns DHCP setup
ip_address	string	Returns IP Address
subnet_mask	string	Returns Subnet Mask
default_gateway	string	Returns Default Gateway
dns_server_1	string	Returns DNS Server 1
dns_server_2	string	Returns DNS Server 2
wireless_lan	object	Returns information related to wireless network (Wi-Fi)
- ssid	string	Returns SSID
- type	string	Returns encryption type Values: "none" / "wep" / "wpa2-psk(aes)" / "mixed_mode"
- key	string	Reserved
- ch	integer	Returns a channel number of connected AP. Returns 0 when disconnected
- strength	integer	Returns wireless strength. Returns -1 when Link is down status Value Range: -1 (Link Down)、0 ~ 100
wireless_direct	object	Returns information related to wireless network (Wireless Direct)
- ssid	string	Returns SSID
- type	string	Returns encryption type Values: "none" / "wpa2-psk(aes)"
- key	string	Reserved
musiccast_network	object	Returns information related to MusicCast Network
- ready	boolean	Returns a status whether or not a Device joins to MusicCast Network
- device_type	string	Returns a type of MusicCast Network on a Device. Returns "unknown" If not joined to MusicCast Network yet

		Values: "root" / "node" / "leaf" / "standard" / "unknown"
- child_num	integer	Returns clients number connected to a Device
- ch	integer	Returns a channel number that a Device discloses. Returns 0 when device_type is other than root or node
- initial_join_running	boolean	Returns whether or not Initial Join is being executed
mac_address	object	Returns MAC Address (12 digit ASCII)
- wired_lan	string	Returns wired MAC Address
- wireless_lan	string	Returns wireless (Wi-Fi) MAC Address
- wireless_direct	string	Returns wireless (Wireless Direct) MAC Address
vtuner_id	string	Reserved
airplay_pin	string	Reserved

Example Response	<pre>{   "response_code":0,   "network_name":"Yamaha AVR",   "connection":"wireless_lan",   "dhcp":true,   "ip_address":"192.168.0.10",   "subnet_mask":"255.255.255.0",   "default_gateway":"192.168.0.1",   "dns_server_1":"192.168.0.1",   "dns_server_2":"192.168.0.2",   "wireless":{     "ssid":"AP1-2.4GHz",     "type":"wpa2-psk(aes)",     "key":"abadcafe",     "strength":88   },   "musiccast_network":{     "ready":true,     "device_type":"root",     "child_num":1,     "initial_join_running":false   },   "mac_address":{     "wired_lan":"00A0DE012345",     "wireless_lan":"78A501012345",     "wireless_direct":"78A501012346"   },   "vtuner_id":"00A0DE012345",   "airplay_pin":"abc123", }</pre>
------------------	--

#### 4.4. getFuncStatus

For retrieving setup/information of overall system function. Parameters are readable only when corresponding functions are available in “func\_list” of /system/getFeatures

##### Request Parameters

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

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details
auto_power_standby	boolean	Returns whether or not Auto Power Standby is enabled
ir_sensor	boolean	Returns whether or not IR sensor is available
speaker_a	boolean	Returns whether or not Speaker A is available
speaker_b	boolean	Returns whether or not Speaker B is available
headphone	boolean	Returns whether or not headphone is connected
dimmer	integer	Returns Dimmer setting number. Returns -1 if it's Auto Dimmer, and 0 or higher if its' manual setup
zone_b_volume_sync	boolean	Reserved

Example Response	{ "response_code":0, "auto_power_standby":false, "ir_sensor":false, "speaker_a":true, "speaker_b":false, "headphone":false, "dimmer":3, "zone_b_volume_sync":false }
------------------	---

**4.5. setAutoPowerStandby**

For setting Auto Power Standby status. Actual operations/reactions of enabling Auto Power Standby depend on each Device

**Request Parameters**

name	type	required	description
enable	boolean	yes	Specifies Auto Power Standby status

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

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

	}
--	---

## 4.6. getLocationInfo

For retrieving Location information

### Request Parameters

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

### Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details
id	string	Returns Location ID in 32-digit hex
name	string	Returns Location Name
zone_list	object	Returns Location setup status of each Zone. Retrievable only when each Zone is valid
- main	boolean	Returns whether or not Main Zone Location setup is valid
- zone2	boolean	Returns whether or not Zone2 Location setup is valid
- zone3	boolean	Returns whether or not Zone3 Location setup is valid
- zone4	boolean	Returns whether or not Zone4 Location setup is valid
stereo_pair_status	string	Reserved

Example Response	<pre>{   "response_code":0,   "id":"9A237BF5AB80ED3C7251DFF49825CA42",   "name":"Home",   "zone_list":{     "main":true,     "zone2":true,     "zone3":false   },   "stereo_pair_status":"none" }</pre>
------------------	---

## 4.7. sendIrCode

For sending specific remote IR code. A Device is operated same as remote IR code reception. But continuous IR code cannot be used in this command. Refer to each Device's IR code list for details

### Request Parameters

name	type	required	description
code	string	yes	Specifies IR code in 8-digit hex

URI	<BaseURL>/v1/system/sendIrCode?code=<code>
Method	GET



Example Request	http://{host}/YamahaExtendedControl/v1/system/sendIrCode?code=7F016C13
-----------------	--

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

## 5. Zone

### 5.1. getStatus

For retrieving basic information of each Zone like power, volume, input and so on

**Request Parameters**

name	type	required	description
zone	string	yes	Specifies target Zone Values: "main" / "zone2" / "zone3" / "zone4"

URI	<BaseURL>/v1/<zone>/getStatus
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/main/getStatus

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details
power	string	Returns power status Values: "on" / "standby"
sleep	integer	Returns Sleep Timer setup value (unit in minutes) Values: 0 / 30 / 60 / 90 / 120
volume	integer	Returns volume value Value Range: calculated by minimum/maximum/step values gotten via /system/getFeatures
mute	boolean	Returns mute status
max_volume	integer	Returns Max Volume setup Value Range: calculated by minimum/maximum/step values gotten via /system/getFeatures
input	string	Returns selected Input ID Values: Input IDs gotten via /system/getFeatures
distribution_enable	boolean	Returns whether or not current Input is distributable status
sound_program	string	Returns selected Sound Program ID Values: Sound Program IDs gotten via /system/getFeatures
direct	boolean	Returns Direct status

pure_direct	boolean	Returns Pure Direct status
enhancer	boolean	Returns Enhancer status
tone_control	object	Returns Tone Control status
- mode	string	Returns Mode type. Available on API version XXX Values: what is gotten via /system/getFeatures
- bass	integer	Returns Bass status Value Range: calculated by minimum/maximum/step values gotten via /system/getFeatures
- treble	integer	Returns Treble status Value Range: calculated by minimum/maximum/step values gotten via /system/getFeatures
equalizer	object	Returns Equalizer status
- mode	string	Returns Mode type. Available on API version XXX Values: what is gotten via /system/getFeatures
- low	integer	Returns Low status Value Range: calculated by minimum/maximum/step values gotten via /system/getFeatures
- mid	integer	Returns Mid status Value Range: calculated by minimum/maximum/step values gotten via /system/getFeatures
- high	integer	Returns High status Value Range: calculated by minimum/maximum/step values gotten via /system/getFeatures
balance	integer	Returns Speaker L/R Balance status. Negative values are for left side, positive values are for right side balance Value Range: calculated by minimum/maximum/step values gotten via /system/getFeatures
dialogue_level	integer	Returns Dialogue Level status Values: Value range calculated by minimum/maximum/step values gotten via /system/getFeatures
dialogue_lift	integer	Returns Dialogue Lift status Value Range: calculated by minimum/maximum/step values gotten via /system/getFeatures
clear_voice	boolean	Returns Clear Voice status
subwoofer_volume	integer	Returns Subwoofer Volume status Value Range: calculated by minimum/maximum/step values gotten via /system/getFeatures
bass_extension	boolean	Returns Bass Extension status
link_control	string	Returns Link Control setup status Values: what is gotten via /system/getFeatures
link_audio_delay	string	Returns Audio Delay setup status Values: what is gotten via /system/getFeatures
disable_flags	integer	Returns flags of functions that are not operable/available currently. Disable status is expressed as OR of bit field shown below, i.e. functions that listed in zone - func_list of /system/getFeatures, and also that have no flags of below are only valid/available/operable b[0] Volume b[1] Mute b[2] Link Audio Delay

Example Response	<pre>{   "response_code":0,   "power":"on",   "sleep":0,   "volume":30,   "mute":false,   "max_volume":100,   "input":"pandora",   "distribution_enable":true,   "sound_program":"chamber",   "pure_direct":false,   "enhancer":true,   "tone_control":{     "bass":0,     "treble":2   },   "dialogue_level":0,   "dialogue_lift":1,   "disable_flags":0 }</pre>
------------------	---

## 5.2. getSoundProgramList

For retrieving a list of Sound Program available in each Zone. It is possible for the list contents to be dynamically changed

### Request Parameters

name	type	required	description
zone	string	yes	Specifies target Zone Values: "main" / "zone2" / "zone3" / "zone4"

URI	<BaseURL>/v1/<zone>/getSoundProgramList
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/main/getSoundProgramList

### Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details
sound_program_list	string array	Returns a list of selectable sound program

Example Response	<pre>{   "response_code":0,   "sound_program_list":[     "munich",     "vienna",     ...     "straight"   ] }</pre>
------------------	---

	]
	}

### 5.3. setPower

For setting power status of 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"
power	string	yes	Specifies power status Values: "on" / "standby" / "toggle"

URI	<BaseURL>/v1/<zone>/setPower?power=<power>
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/zone3/setPower?power=on

#### Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

### 5.4. setSleep

For setting Sleep Timer for each Zone

With Zone B enabled Devices, target Zone is described as Master Power, but Main Zone is used to set it up via YXC

#### Request Parameters

name	type	required	description
zone	string	yes	Specifies target zone. Available for zones with this function Values: "main" / "zone2" / "zone3" / "zone4"
sleep	integer	yes	Specifies Sleep Time (unit in minutes) Values: 0 / 30 / 60 / 90 / 120

URI	<BaseURL>/v1/<zone>/setSleep?sleep=<sleep>
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/main/setSleep?sleep=120

#### Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

## 5.5. setVolume

For setting volume in each Zone. Values of specifying range and steps are different. There are some Devices that cannot allow this value to be up to Device's maximum volume.

### Request Parameters

name	type	required	description
zone	string	yes	Specifies target zone. Available for zones with this function Values: "main" / "zone2" / "zone3" / "zone4"
volume	integer	yes	Specifies volume value Value Range: calculated by minimum/maximum/step values gotten via /system/getFeatures (Available on and after API Version 1.17) "up" / "down"
step	integer	no	Specifies volume step value if the volume is "up" or "down". If nothing specified, minimum step value is used implicitly. (Available on and after API Version 1.17) Values: Value range calculated by minimum/maximum/step values gotten via /system/getFeatures

URI	<BaseURL>/v1/<zone>/setVolume?volume=<volume>&step=<step>
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/main/setVolume?volume=50

### Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

## 5.6. setMute

For setting mute status 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"
enable	boolean	yes	Specifying mute status

URI	<BaseURL>/v1/<zone>/setMute?enable=<enable>
-----	---

Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/main/setMute?enable=false

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

**5.7. setInput**

For selecting each Zone input

**Request Parameters**

name	type	required	description
zone	string	yes	Specifies target zone Values: "main" / "zone2" / "zone3" / "zone4"
input	string	yes	Specifies Input ID Values: Input IDs gotten via /system/getFeatures
mode	string	no	Specifies select mode. If no parameter is specified, actions of input change depend on a Device's specification Value: "autoplay_disabled" (Restricts Auto Play of Net/USB related Inputs). Available on and after API Version 1.12

URI	<BaseURL>/v1/<zone>/setInput?input=<input>&mode=<mode>
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/main/setInput?input=airplay&mode=autoplay_disabled

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

**5.8. setSoundProgram**

For selecting Sound Programs

**Request Parameters**

name	type	required	description
zone	string	yes	Specifies target zone. Available for zones with this function

			Values: "main" / "zone2" / "zone3" / "zone4"
program	string	yes	Specifies Sound Program ID Values: Sound Program IDs gotten via /system/getFeatures

URI	<BaseURL>/v1/<zone>/setSoundProgram?program=<program>		
Method	GET		
Example Request	http://{host}/YamahaExtendedControl/v1/main/setSoundProgram?program=vienna		

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

**5.9. prepareInputChange**

Let a Device do necessary process before changing input in a specific zone. This is valid only when "prepare\_input\_change" exists in "func\_list" found in /system/getFuncStatus.

MusicCast CONTROLLER executes this API when an input icon is selected in a Room, right before sending various APIs (of retrieving list information etc.) regarding selecting input

**Request Parameters**

name	type	required	description
zone	string	yes	Specifies target zone Values: "main" / "zone2" / "zone3" / "zone4"
input	string	yes	Specifies Input ID Values: Input IDs gotten via /system/getFeatures

URI	<BaseURL>/v1/<zone>/prepareInputChange?input=<input>		
Method	GET		
Example Request	http://{host}/YamahaExtendedControl/v1/main/prepareInputChange?input=usb		

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

**6. Tuner**

## APIs in regard to Tuner setting and getting information

Target inputs: AM / FM / DAB

## 6.1. getPresetInfo

For retrieving Tuner preset information

**Request Parameters**

name	type	required	description
band	string	yes	Specifying a band. Values depend on Preset Type gotten via /system/getFeatures Values: "common" (common), "am" / "fm" / "dab" (separate)

URI	<BaseURL>/v1/tuner/getPresetInfo?band=<band>
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/tuner/getPresetInfo?band=fm

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details
preset_info	array	Returns preset information. A number of array elements is to be retrieved via /system/getFeature
- band	string	Returns band information. "unknown" when no presets Values: "am" / "fm" / "dab" / "unknown"
- number	integer	Returns; frequency (unit in kHz) (band = AM or FM) Station ID (band = DAB) 0 when there's no presets
- hd_program	integer	Reserved
func_list	string array	Returns a list of valid functions for Preset. (Recall/Store functions are always valid without specifically listed here) Values: "clear" / "move"

Example Response	<pre>{   "response_code":0,   "preset_info":[     {       "band":"am",       "number":530       "hd_program":0     },     {       "band":"fm",       "number":87500,       "hd_program":0     },     ...   ] }</pre>
------------------	--



	<pre> {     "band": "unknown",     "number": 0,     "hd_program": 0 } ], "func_list": [     "clear",     "move" ] } </pre>
--	--

## 6.2. getPlayInfo

For retrieving playback information of Tuner

### Request Parameters

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

### Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to <a href="#">"Response Code List"</a> for details
band	string	Returns current Band Values: "am" / "fm" / "dab"
auto_scan	boolean	Returns Auto Scan (Up or Down) status
auto_preset	boolean	Returns Auto Preset execution status
am	object	Returns AM related information
- preset	integer	Returns current preset number. 0 when there's no presets Values: 0 (no presets), or one in the range gotten via /system/getFeatures
- freq	integer	Returns frequency (unit in kHz)
- tuned	boolean	Returns Tuned status
fm	object	Returns FM related information
- preset	integer	Returns current preset number. 0 when there's no presets Values: 0 (no presets), or one in the range gotten via /system/getFeatures
- freq	integer	Returns frequency (unit in kHz). In case of DAB, returns Station ID
- tuned	boolean	Returns Tuned status
- audio_mode	string	Returns Audio Mode Values: "mono" / "stereo"
rds	object	Returns RDS information. Available only when RDS is valid
- program_type	string	Returns Program Type
- program_service	string	Returns Program Service
- radio_text_a	string	Returns Radio Text A
- radio_text_b	string	Returns Radio Text B
dab	object	Returns DAB related information. Available only when DAB is valid
- preset	integer	Returns current preset number. 0 when current station is not in presets

		Values: 0 (no presets), or one in the range gotten via /system/getFeatures
- id	integer	Returns Station ID
- status	string	Returns DAB status. When it's in Tune Aid, valid parameters are "tune_aid" and "CH Label" only Values: "not_ready" / "initial_scan" / "tune_aid" / "ready"
- freq	integer	Returns DAB frequency (unit in kHz) Value Range: 174000 - 240000
- category	string	Returns Category Values: "primary" / "secondary"
- audio_mode	string	Returns Audio Mode Values: "mono" / "stereo"
- bit_rate	integer	Returns audio bitrate (unit in kbps) Value Range: 32 ~ 256
- quality	integer	Returns signal quality level Value Range: 0 - 100
- tune_aid	integer	Returns signal strength level Value Range: 0 - 100
- off_air	boolean	Returns Off Air status
- dab_plus	boolean	Returns DAB+ status
- program_type	string	Returns Program Type
- ch_label	string	Returns CH Label
- service_label	string	Returns Service Label
- dls	string	Returns DLS
- ensemble_label	string	Returns Ensemble Label
- initial_scan_progress	integer	Returns Initial Scan progress status. Available only when "dab_initial_scan" exists in tuner - func_list under /system/getFeatures Value Range: 0 - 100
- total_station_num	integer	Returns station numbers detected by Initial Scan. Available only when "dab_initial_scan" exists in tuner - func_list under /system/getFeatures 0 if Initial Scan hasn't executed or nothing found Value Range: 0 - 255
hd_radio	object	Reserved

Example Response	<pre>{   "response_code":0,   "band":"fm",   "auto_scan":false,   "am":{     "preset":8,     "freq":531,     "tuned":true   },   "fm":{     "preset":0,     "freq":87500,     "tuned":true   },   "rds":{     "program_type":"Affairs",</pre>
------------------	---

```

"program_service":"RDS PS",
"radio_text_a":"RDS Radio Text A",
"radio_text_b":"RDS Radio Text B"
},
"dab":{
  "preset":10,
  "id":10000,
  "status":"ready",
  "freq":22564,
  "category":"primary",
  "audio_mode":"stereo",
  "bit_rate":128,
  "quality":90,
  "tune_aid":80,
  "off_air":false,
  "dab_plus":false,
  "program_type":"Pop Music",
  "ch_label":"12D",
  "service_label":"BBC Radio 1",
  "dls":"The Chart Game: www.bbc.co.uk/radio2",
  "ensemble_label":"BBC Network DAB"
},
"hd_radio":{
  "program":1,
  "program_list":[
    1,
    2,
    3
  ],
  "station":"WXYZ-FM",
  "category":"Top 40",
  "title":"Beat It",
  "artist":"Michael Jackson",
  "album":"Thriller"
}
}

```

### 6.3. setFreq

Fr setting Tuner frequency

#### Request Parameters

name	type	required	description
band	string	yes	Specifies Band Values : "am" / "fm"
tuning	string	yes	Specifies a tuning method. Use "tp_up" and "tp_down" only when Band is RDS Values: "up" / "down" / "cancel" / "auto_up" / "auto_down" / "tp_up" / "tp_down" / "direct"

num	integer	no	Specifies frequency (unit in kHz). Valid only when tuning is "direct"
-----	---------	----	---

URI	<BaseURL>/v1/tuner/setFreq?band=<band>&tuning=<tuning>&num=<num>		
Method	GET		
Example Request	http://{host}/YamahaExtendedControl/v1/tuner/setFreq?band=fm&tuning=direct&num=87500		

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

**6.4. recallPreset**

For recalling a Tuner preset

**Request Parameters**

name	type	required	description
zone	string	yes	Specifies station recalling zone. This causes input change in specified zone Values: "main" / "zone2" / "zone3" / "zone4"
band	string	yes	Specifies Band type. Depending on Preset Type gotten via /system/getFeatures, specifying value is different Values: "common" (band common) / "separate" (each band preset)
num	integer	yes	Specifies Preset number Value: one in the range gotten via /system/getFeatures

URI	<BaseURL>/v1/tuner/recallPreset?zone=<zone>&band=<band>&num=<num>		
Method	GET		
Example Request	http://{host}/YamahaExtendedControl/v1/tuner/recallPreset?zone=main&band=fm&num=13		

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

**6.5. switchPreset**

For selecting Tuner preset.

Call this API after change the target zone's input to Tuner. It is possible to change Band in case of preset type is "common". In case of preset type is "separate", need to change the target Band before calling this API.

This API is available on and after API Version 1.17.

### Request Parameters

name	type	required	description
dir	string	yes	Specifies change direction of preset Values: "next" / "previous"

URI	<BaseURL>/v1/tuner/switchPreset?dir=<dir>		
Method	GET		
Example Request	http://{host}/YamahaExtendedControl/v1/tuner/switchPreset?dir=next		

### Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }	
------------------	-----------------------------	--

## 6.6. storePreset

For registering current station to a preset

### Request Parameters

name	type	required	description
num	integer	yes	Specifying a preset number Value: one in the range gotten via /system/getFeatures

URI	<BaseURL>/v1/tuner/storePreset?num=<num>		
Method	GET		
Example Request	http://{host}/YamahaExtendedControl/v1/tuner/storePreset?num=10		

### Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }	
------------------	-----------------------------	--

## 6.7. setDabService

For selecting DAB Service. Available only when DAB is valid to use

**Request Parameters**

name	type	required	description
dir	string	yes	Specifies change direction of services Values: "next" / "previous"

URI	<BaseURL>/v1/tuner/setDabService?dir=<dir>		
Method	GET		
Example Request	http://{host}/YamahaExtendedControl/v1/tuner/setDabService?dir=next		

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }	
------------------	-----------------------------	--

## 7. Network/USB

APIs in regard to Network/USB related setting and getting information

Target Inputs: USB / Network related ones (Server / Net Radio / Pandora / Spotify / AirPlay etc.)

### 7.1. getPresetInfo

For retrieving preset information. Presets are common use among Net/USB related input sources

**Request Parameters**

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

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details
preset_info	array	Returns preset info. Element number of an array can be gotten via system/getFeatures
- input	string	Returns Input ID. Returns "unknown" if no presets available
- text	string	Returns text info. Returns "" (empty text) if no presets available
func_list	string array	Returns a list of valid functions for Preset. (Recall/Store functions are always valid without specifically listed here) Values: "clear" / "move"

Example Response	{
------------------	---

	<pre> "response_code":0, "preset_info":[   {     "input":"net_radio",     "text":"CBC Radio Jazz"   },   {     "input":"pandora",     "text":"Queen Radio"   },   ...   {     "input":"unknown",     "text":""   } ], "func_list":[   "clear",   "move" ] } </pre>
--	--

## 7.2. getPlayInfo

For retrieving playback information

### Request Parameters

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

### Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to <a href="#">"Response Code List"</a> for details
input	string	Returns current Net/USB related Input ID Refer to <a href="#">"All ID List"</a> for details
playback	string	Returns playback status Values: "play" / "stop" / "pause" / "fast_reverse" / "fast_forward"
repeat	string	Returns repeat setting status Value: "off" / "one" / "all"
shuffle	string	Returns shuffle setting status Values: "off" / "on" / "songs" / "albums"
play_time	integer	Returns current playback time (unit in second). Returns -60000 if playback time is invalid Value Range: -60000 (invalid) / -59999 ~ 59999 (valid)
total_time	integer	Returns total playback time (unit in second). Returns 0 if total time is not available or invalid Value Range: 0 ~ 59999

artist	string	<p>Returns artist name.</p> <p>Returns station name if the input is Net Radio / Pandora / radiko.</p> <p>Returns station name/artist name if the input is Rhapsody (Radio).</p> <p>Returns ad name if Pandora playbacks ad contents.</p> <p>If input is MC Link, returns master's internal content info or Room Name if the master input is one of external sources</p> <p>Note: Rhapsody service name will be changed to Napster.</p>
album	string	<p>Returns album name.</p> <p>Returns channel name if the input is SiriusXM.</p> <p>Returns subtitle name if the input is radiko.</p> <p>Returns company name if Pandora playbacks an ad.</p> <p>If input is MC Link, returns master's internal content info or Input Name if the master input is one of external sources</p>
track	string	<p>Returns track name.</p> <p>Returns song name if the input is Rhapsody / SiriusXM / Pandora.</p> <p>Returns title name if the input is radiko.</p> <p>If input is MC Link, returns master's internal content info or empty text if the master input is one of external sources</p> <p>Note: Rhapsody service name will be changed to Napster.</p>
albumart_url	string	<p>Returns a URL to retrieve album art data. Data is in jpg/png/bmp/ymf format. The path is given as relative address. If "xxx/yyy/zzz.jpg" is returned, the absolute path is expressed as http://{host}/xxx/yyy/zzz.jpg</p> <p>Note: ymf is original format encrypted by Yamaha AV encryption method.</p> <p>Note: Return jpg/png/bmp format data on and after API Version 1.17.</p>
albumart_id	integer	<p>Returns ID to identify album art. If ID got changed, retry to get album art data via albumart_url</p> <p>Value Range: 0 ~ 255</p>
usb_devicetype	string	<p>Returns USB device type. Returns "unknown" if no USB is connected</p> <p>Values: "msc" / "ipod" / "unknown"</p>
auto_stopped	boolean	<p>Returns whether or not auto top has initiated. If it is true, display appropriate messages to the external application user interface depending on which input current one is. This flag is cleared (set back to false) with these conditions as follows;</p> <ul style="list-style-type: none"> <li>- Playback is initiated properly</li> <li>- /netusb/setPlayback is executed</li> <li>- type = play found in /netusb/setListControl is executed</li> </ul> <p>Target Input : Pandora / SiriusXM</p> <p>A MusicCast Device that detects non-operation time (by key operation on the Device or by remote control) will always return false flag in this data</p>
attribute	integer	<p>Returns playback attribute info. Attributes are expressed as OR of bit field as shown below;</p> <p>b[0] Playable</p> <p>b[1] Capable of Stop</p> <p>b[2] Capable of Pause</p>



		<p> b[3] Capable of Prev Skip  b[4] Capable of Next Skip  b[5] Capable of Fast Reverse  b[6] Capable of Fast Forward  b[7] Capable of Repeat  b[8] Capable of Shuffle  b[9] Feedback Available (Pandora)  b[10] Thumbs-Up (Pandora)  b[11] Thumbs-Down (Pandora)  b[12] Video (USB)  b[13] Capable of Bookmark (Net Radio)  b[14] DMR Playback (Server)  b[15] Station Playback (Rhapsody / Napster)  b[16] AD Playback (Pandora)  b[17] Shared Station (Pandora)  b[18] Capable of Add Track (Rhapsody/Napster/Pandora/JUKE/Qobuz)  b[19] Capable of Add Album (Rhapsody / Napster / JUKE)  b[20] Shuffle Station (Pandora)  b[21] Capable of Add Channel (Pandora)  b[22] Sample Playback (JUKE)  b[23] MusicPlay Playback (Server)  b[24] Capable of Link Distribution  b[25] Capable of Add Playlist (Qobuz) </p> <p> With Pandora, b[9] = 1 validates "thumbs_up" / "thumbs_down" / "mark_tired" of managePlay and "why_this_song" of getPlayDescription.  b[21] = 1 validates "add_channel_track" / "add_channel_artist" </p> <p>Note: Rhapsody service name will be changed to Napster.</p>
--	--	--

Example Response	<pre> {   "response_code":0,   "input":"usb",   "playback":"stop",   "repeat":"off",   "shuffle":"on",   "play_time":200,   "total_time":314,   "artist":"尾崎豊",   "album":"壊れた扉から",   "track":"Forget-me-not",   "albumart_url":"/YamahaRemoteControl/AlbumART/AlbumART.jpg",   "albumart_id":18,   "usb_devicetype":"msc",   "attribute":511 } </pre>
------------------	--

### 7.3. setPlayback

For controlling playback status

### Request Parameters

name	type	required	description
playback	string	yes	Specifies playback status Values: "play" / "stop" / "pause" / "play_pause" / "previous" / "next" / "fast_reverse_start" / "fast_reverse_end" / "fast_forward_start" / "fast_forward_end"

URI	<BaseURL>/v1/netusb/setPlayback?playback=<playback>
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/netusb/setPlayback?playback=stop

### Response Parameters

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

## 7.4. toggleRepeat

For toggling repeat setting. No direct / discrete setting commands available

### Request Parameters

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

### Response Parameters

name	type	Description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

## 7.5. toggleShuffle

For toggling shuffle setting. No direct / discrete setting commands available

### Request Parameters

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

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

**7.6. getListInfo**

For retrieving list information. Basically this info is available to all relevant inputs, not limited to or independent from current input

**Request Parameters**

name	type	required	description
list_id	string	no	Specifies list ID. If nothing specified, "main" is chosen implicitly Values: "main" (common for all Net/USB sources) "auto_complete" (Pandora) "search_artist" (Pandora) "search_track" (Pandora)
input	string	yes	Specifies target Input ID. Controls for setListControl are to work with the input specified here Values: Input IDs for Net/USB related sources
index	integer	no	Specifies the reference index (offset from the beginning of the list). Note that this index must be in multiple of 8. If nothing was specified, the reference index previously specified would be used Values: 0, 8, 16, 24, ..., 64984, 64992
size	integer	yes	Specifies max list size retrieved at a time Value Range: 1 - 8
lang	string	no	Specifies list language. But menu names or text info are not always necessarily pulled in a language specified here. If nothing specified, English ("en") is used implicitly Values: "en" (English)/ "ja" (Japanese)/ "fr" (French)/ "de" (German)/ "es" (Spanish)/ "ru" (Russian)/ "it" (Italy)/ "zh" (Chinese)

URI	<BaseURL>/v1/netusb/getListInfo?input=<input>&index=<index>&size=<size>&lang=<lang>
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/netusb/getListInfo?input=usb&index=32&size=8&lang=en

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details  If codes shown below is returned, display appropriate messages to the

		external application user interface depending on which input current one is. List info is initialized once an error happens. 100 : Access Error (common for all Net/USB sources) 112 : Access Denied (Server)
menu_layer	integer	Returns menu layer Value Range: 0 - 15
max_line	integer	Returns max row number of the list Value Range: 0 - 65000
index	integer	Returns the reference index of the list
playing_index	integer	Returns the index number in a list that is currently in playback. If no list element has playback status, or playback status wasn't found, returns -1
menu_name	string	Returns menu name
list_info	array	Returns list info with the row number size specified by Request Parameter. Returns empty data if index or size wasn't specified. If the max row number is reached in the middle of this list info, the list ends at the max row.
- text	string	Returns text info to display
- thumbnail	string	Return a URL to retrieve thumbnail data. Returns "" (empty text) if no sum nail is available
- attribute	integer	<p>Returns attribute info of list elements. Attributes are expressed as OR of bit field as shown below;</p> <p>b[0] Name exceeds max byte limit (common for all Net/USB sources)</p> <p>b[1] Capable of Select (common for all Net/USB sources)</p> <p>b[2] Capable of Play (common for all Net/USB sources)</p> <p>b[3] Capable of Search (Rhapsody / Napster / JUKE)</p> <p>b[4] Album Art available (common for all Net/USB sources)</p> <p>b[5] Now Playing (Pandora)</p> <p>b[6] Capable of Add Bookmark (Net Radio)</p> <p>b[7] Capable of Add Track (Rhapsody / Napster / JUKE / Qobuz)</p> <p>b[8] Capable of Add Album (Rhapsody / Napster / JUKE / Qobuz)</p> <p>b[9] Capable of Add Channel (Rhapsody / Napster / Pandora)</p> <p>b[10] Capable of Remove Bookmark (Net Radio)</p> <p>b[11] Capable of Remove Track (Rhapsody / Napster / JUKE / Qobuz)</p> <p>b[12] Capable of Remove Album (Rhapsody / Napster / JUKE / Qobuz)</p> <p>b[13] Capable of Remove Channel (Rhapsody / Napster / Pandora)</p> <p>b[14] Capable of Remove Playlist (Rhapsody / Qobuz)</p> <p>b[15] Playlist (JUKE / Qobuz)</p> <p>b[16] Radio (JUKE)</p> <p>b[17] Shuffle (Pandora)</p> <p>b[18] Shared Station (Pandora)</p> <p>b[19] Premium Item (radiko)</p> <p>b[20] Capable of Add Artist (Qobuz)</p> <p>b[21] Capable of Remove Artist (Qobuz)</p> <p>b[22] Capable of Add Playlist (Qobuz)</p> <p>Regarding an element with its attributes of b[1] and b[2] both set as 1, it is valid to do both layer movement/shift and start playback by setListControl. If an element got neither b[1] = 0 nor b[2] = 0, it is unavailable content so apply appropriate UI reaction like making it</p>

		gray-out etc.
		Note: Rhapsody service name will be changed to Napster.

Example Response	<pre>{   "response_code":0,   "menu_layer":3,   "max_line":40,   "index":32,   "playing_index":-1,   "menu_name":"The Beatles",   "list_info":[     {       "text":"Please Please Me",       "attribute":2     },     {       "text":"With The Beatles",       "attribute":2     },     ...     {       "text":"Hey Jude"       "attribute":20     },     {       "text":"Let It Be",       "attribute":20     }   ] }</pre>
------------------	--

## 7.7. setListControl

For control a list. Controllable list info is not limited to or independent from current input

### Request Parameters

name	type	required	description
list_id	string	no	Specifies list ID. If nothing specified, "main" is chosen implicitly Values: "main" (common for all Net/USB sources) "auto_complete" (Pandora) "search_artist" (Pandora) "search_track" (Pandora)
type	string	yes	Specifies list transition type. "select" is to enter and get into one deeper layer than the current layer where the element specified by the index belongs to. "play" is to start playback current index element, "return" is to go back one upper layer than current. "select" and "play" needs to specify an index at the same time. In case to "select" an element with its attribute being "Capable of

			Search", specify search text using setSearchString in advance. (Or it is possible to specify search text and move layers at the same time by specifying an index in setSearchString) Values: "select" / "play" / "return"
index	integer	no	Specifies an element position in the list being selected (offset from the beginning of the list). This is mandatory to specify if the type is "select" or "play" Value Range: 0 - 64999
zone	string	no	Specifies target zone to playback. In the specified zone, input change occurs at the same time of playback. This parameter is valid only when type "play" is specified. If nothing is specified, "main" is chosen implicitly 値 : "main" / "zone2" / "zone3" / "zone4"

URI	<BaseURL>/v1/netusb/setListControl?list_id=<list_id>&type=<type>&index=<index>&zone=<zone>		
Method	GET		
Example Request	http://{host}/YamahaExtendedControl/v1/netusb/setListControl?list_id=main&type=play&index=50&zone=main		

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

**7.8. setSearchString**

For setting search text. Specifies string executing this API before select an element with its attribute being "Capable of Search" or retrieve info about searching list(Pandora).

**Request Parameters**

name	type	required	description
list_id	string	no	Specifies list ID. If nothing specified, "main" is chosen implicitly Values : "main" (All) "auto_complete" (Pandora) "search_artist" (Pandora) "search_track" (Pandora)
string	string	yes	Setting search text
index	integer	no	Specifies an element position in the list being selected (offset from the beginning of the list).Valid only when the list_id is "main" Specifies index an element with its attribute being "Capable of Search" Controls same as setListControl "select" are to work with the index an element specified. If no index is specified, non-actions of select Values : 0 ~ 64999

URI	<BaseURL>/v1/netusb/setSearchString
Method	POST
Example Request (URI)	http://{host}/YamahaExtendedControl/v1/netusb/setSearchString
Example Request (json)	{ "list_id":"auto_complete", "string":"michael" }

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

**7.9. recallPreset**

For recalling a content preset

**Request Parameters**

name	type	required	description
zone	string	yes	Specifies station recalling zone. This causes input change in specified zone Values: "main" / "zone2" / "zone3" / "zone4"
num	integer	yes	Specifies Preset number Value: one in the range gotten via /system/getFeatures

URI	<BaseURL>/v1/netusb/recallPreset?zone=<zone>&num=<num>
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/netusb/recallPreset?zone=main&num=13

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

**7.10. storePreset**

For registering current content to a preset. Presets are common use among Net/USB related input sources.

**Request Parameters**

name	type	required	description
num	integer	yes	Specifying a preset number Value: one in the range gotten via /system/getFeatures

URI	<BaseURL>/v1/netusb/storePreset?num=<num>		
Method	GET		
Example Request	http://{host}/YamahaExtendedControl/v1/netusb/storePreset?num=10		

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }	
------------------	-----------------------------	--

**7.11. getAccountStatus**

For retrieving account information registered on Device

**Request Parameters**

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

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details
service_list	array	Returns a list of Streaming Service
- id	string	Returns current Input ID Refer to " <a href="#">All ID List</a> " for details
- registered	boolean	Returns whether account registered on device correctly. If input is Pandora, returns "false" if no registered account or not select to use registered account.
- login_status	string	Returns login status Value : "logged_in" (All) "logged_out" (All) "access_error" (All) "password_changed" (Rhapsody / Napster / SiriusXM) "account_expired" (Rhapsody / Napster / JUKE) "invalid_account" (Rhapsody / Napster / Spotify) "suspended_account" (Pandora / Spotify)  Note: Rhapsody service name will be changed to Napster.



- username	string	Reserved
- type	string	Returns account type Value: "formal" (official account) / "trial" (trial account) / "unpaid" (unpaid account) / "expired" (expired account)
- trial_time_left	integer	Returns remaining trial days only account type is "trial". If this value is 0, trial account is set to expire within 24 hours. Return -1 if trial account out of date. Return -2 if not retrieving information from server.

Example Response	<pre>{   "response_code":0,   "service_list":[     {       "id":"rhapsody",       "registered":true,       "login_status":"logged_in",       "username":"user1",       "type":"formal",       "trial_time_left":0     },     {       "id":"siriusxm",       "registered":true,       "login_status":"logged_in",       "username":"",       "type":"trial",       "trial_time_left":7     },     ...     {       "id":"qobuz",       "registered":false,       "login_status":"logged_out",       "username":"",       "type":"formal",       "trial_time_left":0     }   ] }</pre>
------------------	---

## 7.12. switchAccount

For switching account for service corresponding multi account

### Request Parameters

name	type	required	description
input	string	yes	Specifies target Input ID. Value: "pandora"
index	integer	yes	Specifies switch account index Value : 0 ~ 7 (Pandora)

timeout	integer	yes	Specifies timeout duration(ms) for this API process. If specifies 0, treat as maximum value. Value: 0 ~ 60000
---------	---------	-----	--

URI	<BaseURL>/v1/netusb/switchAccount?input=<input>&index=<index>&timeout=<timeout>		
Method	GET		
Example Request	http://{host}/YamahaExtendedControl/v1/netusb/switchAccount?input=pandora&index=3&timeout=10000		

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details  If codes shown below is returned, display appropriate messages to the external application user interface depending on which input current one is. 100 : Access error (All) 101 : Other error (All) 105 : suspend account(Pandora)

Example Response	{ "response_code":0 }
------------------	-----------------------------

**7.13. getServiceInfo**

For retrieving information of various Streaming Service. The combination of Input/Type is available as follows;

Input	Type
Pandora	Account List (account_list) : retrieving list of account registered on Device Licensing (licensing) : checking license Activation Code (activation_code) : retrieving Activation Code  ※Disable to check Rhapsody license by referring the value of this APIs response_code. a Device issues events of netusb – account_updated by condition, retrieve the info excute /netusb/getAccountStatus. (Sometimes Device not issue events)  ※Before retrieve Activation Code, retrieve Account List and check not to reach Max about the num of registration. Note: Rhapsody service name will be changed to Napster.

**Request Parameters**

name	type	required	description
input	string	yes	Specifies target Input ID. Value: "pandora" / "rhapsody" / "napster"
type	string	yes	Specifies type of retrieving info

			Value: "account_list" (Pandora) "licensing" (Rhapsody / Napster / Pandora) "activation_code" (Pandora)
timeout	integer	yes	Specifies timeout duration(ms) for this API process. If specifies 0, treat as maximum value. Value: 0 ~ 60000

URI	<BaseURL>/v1/netusb/getServiceInfo?input=<input>&type=<type>&timeout=<timeout>
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/netusb/getServiceInfo?input=pandora&type=account_list&timeout=10000

### Response Parameters (Pandora – Account List)

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details
account_list	array	Returns list of account
- index	integer	Returns index to dissem account Value: 0 ~ 7
- username	string	Reserved-
- selected	boolean	Returns whether this account is selected

Example Response	<pre>{   "response_code":0,   "account_list":[     {       "index":0,       "username":"",       "selected":true     },     ...     {       "index":7,       "username":"",       "selected":false     }   ] }</pre>
------------------	--

### Response Parameters (Pandora – Licensing)

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details  If codes shown below is returned, display appropriate messages to the external application user interface. 100 : Access error 109 : License error

Example Response	{ "response_code":0 }
------------------	-----------------------------

**Response Parameters (Rhapsody/Napster – Licensing)**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details  Note that disable to check license by referring the value of response_code.

Example Response	{ "response_code":0 }
------------------	-----------------------------

**Response Parameters (Pandora – Activation Code)**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details  If codes shown below is returned, display appropriate messages to the external application user interface. 100 : Access error 106 : reach account num ceiling 110 : Read Only Mode
url	string	Returns Activation URL
code	string	Returns Activation Code
index	integer	Returns index to discern account. Execute <a href="#">switchAccount</a> using this index value if login soon after complete Activation on Web site. Value: 0 ~ 7

Example Response	{ "response_code":0, "url":"http://www.pandora.com/yamaha", "code":"00000000", "index":0 }
------------------	---

## 8. CD

APIs in regard to CD setting and getting information

### 8.1. getPlayInfo

For retrieving playback information of CD

**Request Parameters**

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

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to “ <a href="#">Response Code List</a> ” for details
device_status	string	Returns current CD device status Values: "open" / "close" / "ready" / "not_ready"
playback	string	Returns playback status Values: "play" / "stop" / "pause" / "fast_reverse" / "fast_forward"
repeat	string	Returns repeat setting status Value: "off" / "one" / "all" / "folder" / "a-b"
shuffle	string	Returns shuffle setting status Values: "off" / "on" / "folder" / "program"
play_time	integer	Returns current playback time (unit in second). Value Range: -59999 ~ 59999
total_time	integer	Returns current track total playback time (unit in second). Value Range: 0 ~ 59999
disc_time	integer	Returns CD total playback time (unit in second). Value Range: 0 ~ 59999
track_number	integer	Returns track number that is currently in playback. If no track, or playback status is complete stop, returns -1. Values: 0 ~ 512
total_tracks	integer	Returns total track number Values: 0 ~ 512
artist	string	Returns artist name.
album	string	Returns album name.
track	string	Returns track name.

Example Response	<pre>{   "response_code":0,   "device_status":"ready",   "playback":"play",   "repeat":"all",   "shuffle":"on",   "play_time":100,   "total_time":300,   "track_number":5,   "total_tracks":13,   "artist":"Mr.Children",   "album":"SUPERMARKET FANTASY",   "track":"GIFT" }</pre>
------------------	---

**8.2. setPlayback**

For controlling playback status

### **Request Parameters**

name	type	required	description
playback	string	yes	Specifies playback status Values: "play" / "stop" / "pause" / "previous" / "next" / "fast_reverse_start" / "fast_reverse_end" / "fast_forward_start" / "fast_forward_end" / "track_select"
num	integer	no	Specifies target track number to playback. This parameter is valid only when playback "track_select" is specified. Values: 1 ~ 512

URI	<BaseURL>/v1/cd/setPlayback?playback=<playback>&num=<num>
Method	GET
Example Request	http://{host}/YamahaExtendedControl/v1/cd/setPlayback?playback=stop

### **Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

## 8.3. toggleTray

For toggling CD tray Open/Close setting

### **Request Parameters**

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

### **Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to " <a href="#">Response Code List</a> " for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

## 8.4. toggleRepeat

For toggling repeat setting. No direct / discrete setting commands available

### **Request Parameters**

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

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to “ <a href="#">Response Code List</a> ” for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

## 8.5. toggleShuffle

For toggling shuffle setting. No direct / discrete setting commands available

**Request Parameters**

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

**Response Parameters**

name	type	description
response_code	integer	0: Successful request / Others: Error Refer to “ <a href="#">Response Code List</a> ” for details

Example Response	{ "response_code":0 }
------------------	-----------------------------

## 9. 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

## 10. Events

### 10.1. Outline

Events are to notify Devices' status/setup changes immediately to external applications. Events are spread out as UDP unicast.

### 10.2. Conditions

When a request header from an external application contains fields shown below, a Device issues events of status changes. If no event is required, do not include the specified fields:

X-AppName:MusicCast/XXX

X-AppPort:YYY

\*Here XXX stands for an application version, YYY shows event receiving port.

Following is an example of MusicCast CONTROLLER:

X-AppName:MusicCast/1.40(iOS)

X-AppPort:41100

Event notification timeouts in 10 minutes if no further event request is sent from an IP address set as event receiving device. If another request is made within 10 minutes of previous request, the timeout duration is reset and extended.

Event receiving port will be overwritten if a different port number is sent as a request by the registered device using X-AppPort.

### 10.3. Details of event data

Following data sets only which had status/setting change and which are valid on each Device and each zone are to be notified by events.

#### **Event Parameters**

name	type	Description
system	object	Returns System related information
· bluetooth_info_updated	boolean	Returns whether or not Bluetooth info has changed. If so, pull



		renewed info using /system/getBluetoothInfo
· func_status_updated	boolean	Returns whether or not System overall info has changed. If so, pull renewed info using /system/getFuncStatus
· speaker_settings_updated	boolean	Reserved
· name_text_updated	boolean	Reserved
· tag_updated	boolean	Reserved
· location_info_updated	boolean	Returns whether or not Location info has changed. If so, pull renewed info using /system/getLocationInfo
· stereo_pair_info_updated	boolean	Reserved
main	object	Returns Main Zone related information
· power	string	Returns power status Values: "on" / "standby"
· input	string	Returns current Input ID Values: Input IDs gotten via /system/getFeature
· volume	integer	Returns volume value Values: Value range calculated by minimum/maximum/step values gotten via /system/getFeatures
· mute	boolean	Returns mute status
· status_updated	boolean	Returns whether or not other info has changed than main zone power/input/volume/mute status. If so, pull renewed info using /main/getStatus
· signal_info_updated	boolean	Reserved
zone2	object	Reserved
zone3	object	Reserved
zone4	object	Reserved
tuner	object	Returns Tuner related information
· play_info_updated	boolean	Returns whether or not playback info has changed. If so, pull renewed info using /tuner/getPlayInfo
· preset_info_updated	boolean	Returns whether or not preset info has changed. If so, pull renewed info using /tuner/getPresetInfo
netusb	object	Returns Net/USB related information
· play_error	integer	Returns error codes happened during playback for displaying appropriate messages to the external application user interface. If multiple errors happen at the same time, refer to the value of multiple_play_errors sent together for proper messaging Values: 0: No Error 1: Access Error (common for all Net/USB sources) 2: Playback Unavailable (common for all Net/USB sources) 3: Skip Limit Reached (Rhapsody / Napster / Pandora) 4: Invalid Session (Rhapsody / Napster / SiriusXM) 5: High-Resolution File Not Playable at MusicCast Leaf (Server) 6: User Unauthenticated (Qobuz) 7: Track Restricted by Right Holders (Qobuz) 8: Sample Restricted by Right Holders (Qobuz) 9: Genre Restricted by Streaming Credentials (Qobuz) 10: Application Restricted by Streaming Credentials (Qobuz) 11: Intent Restricted by Streaming Credentials (Qobuz)

		100: Multiple Errors (common for all Net/USB sources)  Note: Rhapsody service name will be changed to Napster.
- multiple_play_errors	integer	Returns bit field flags of multiple playback errors. Flags are expressed as OR of bit field. play_error code x is stored as a flag in b[x] shown below. x=0 is reserved for it is for No Error, and x=100 is ignored here b[0] reserved (for it's No Error) b[1] Access Error (common for all Net/USB sources) ... b[11] Intent Restricted by Streaming Credentials (Qobuz)
- play_message	string	Returns playback related message. Max size is 256 bytes
- account_updated	boolean	Returns whether or not account info has changed. If so, pull renewed info using /netusb/getAccountStatus
- play_time	integer	Returns current playback time (unit in second) Value: ranges -59999 - 59999 (TBD)
- preset_info_updated	boolean	Returns whether or not preset info has changed. If so, pull renewed info using netusb/getPresetInfo
- recent_info_updated	boolean	Reserved
- preset_control	object	Returns results of Preset operations
- type	string	Returns a type of Preset operations Values: "store" / "clear" / "recall"
- num	integer	Returns a Preset number being operated Value: one in the range gotten via /system/getFeatures
- result	string	Returns the result of operation Values: "success" (for all types) / "error" (for all types) / "empty" (only for recall) / "not_found" (only for recall)
- trial_status	object	Returns trial status of a Device
- input	string	Returns Input IDs related to Net/USB
- enable	boolean	Returns whether or not trial can be initiated. If false, new trial cannot get started due to a Device in trial status
- trial_time_left	object	Returns remaining time of a trial
- input	string	Returns Net/USB related Input IDs
- time	integer	Returns remaining days of trial. 0 means it expires within 24 hours. -1 means it has expired, -2 means no info is retrieved yet from the server
- play_info_updated	boolean	Returns whether or not playback info has changed. If so, pull renewed info using /netusb/getPlayInfo
- list_info_updated	boolean	Returns whether or not list info has changed. If so, pull renewed info using /netusb/getListInfo
cd	object	Returns CD related information
- device_status	string	Returns CD device status Values: "open" / "close" / "ready" / "not_ready"
- play_time	integer	Returns current playback time (unit in second). Value Range: -59999 ~ 59999
- play_info_updated	boolean	Returns whether or not playback info has changed. If so, pull renewed info using /cd/getPlayInfo
dist	object	Returns Link distribution related information

- dist_info_updated	boolean	Returns whether or not Link distribution Device info has changed. If so, pull renewed info using /dist/getDistributionInfo
clock	object	Reserved
device_id	string	Return s Device ID.ID is same value using /system/getDeviceInfo Note: Available on and after API Version 1.17

Event Example	<pre> {   "system":{     "name_text_updated":true   },   "main":{     "power":"on",     "input":"siriusxm",     "volume":30,     "mute":false,     "status_updated":true   },   "zone2":{     "power":"on",     "input":"cd",     "volume":50,     "mute":false     "enhancer":false   },   "tuner":{     "play_info_updated":false   },   "netusb":{     "play_error ":0,     "account_updated":true,     "play_time":50,     "trial_status":{       "input":"siriusxm",       "enable":false     },     "trial_time_left":{       "input":"siriusxm",       "time":5     },     "play_info_updated":false,     "list_info_updated":false   },   "cd":{     "tray_status":"ready",     "play_time":100,     "play_info_updated":false   } } </pre>
---------------	---

## 11. All ID List

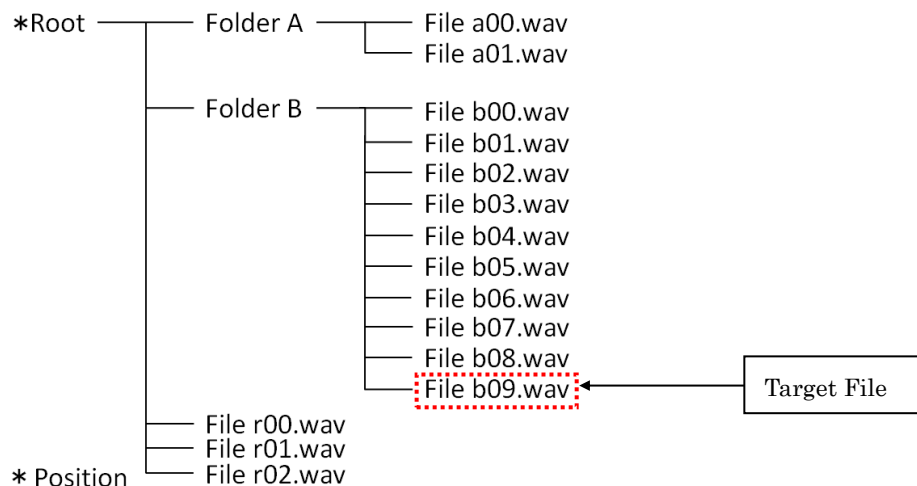
Type	ID
Zone	main / zone2 / zone3 / zone4  ※Zone A is handles as "main", Zone B is handles as "zone2"
Input	cd / tuner / multi_ch / phono / hdmi1 / hdmi2 / hdmi3 / hdmi4 / hdmi5 / hdmi6 / hdmi7 / hdmi8 / hdmi / av1 / av2 / av3 / av4 / av5 / av6 / av7 / v_aux / aux1 / aux2 / aux / audio1 / audio2 / audio3 / audio4 / 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 / rhapsody / napster / pandora / siriusxm / spotify / juke / airplay / radiko / qobuz / 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

Note: Rhapsody service name will be changed to Napster. Accordingly, rhapsody input ID will be changed to napster input ID.

## 12. Application Notes

### 12.1. Example of List Control

This shows an example when you change a Device's input to USB, browse a USB memory which has folders/files as shown below and playback the target file "File b09.wav" in the Folder B



#### 12.1.1. Preparation before start browsing

Send "prepare\_input\_change" to a Device in order to let it do necessary process before the input change

##### Request

`http://{host}/YamahaExtendedControl/v1/main/prepareInputChange?input=usb`

##### Response

```
{
  "response_code":0
}
```

If input change is expressly necessary, send "setInput" to do so.

##### Request

`http://{host}/YamahaExtendedControl/v1/main/setInput?input=usb&mode=autoplay_disabled`

##### Response

```
{
  "response_code":0
}
```

#### 12.1.2. Retrieving list information (In case max line number is less than or equal to 8)

Use "getListInfo" command to get list information

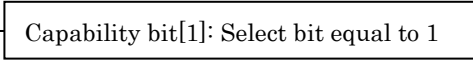
Here shows how to get it at the root layer

##### Request

`http://{host}/YamahaExtendedControl/v1/netusb/getListInfo?input=usb&index=0&size=8&lang=en`

Response

```
{
  "response_code":0,
  "menu_layer":0,
  "max_line":5,
  "index":0,
  "playing_index":-1,
  "menu_name":"USB",
  "list_info":[
    {
      "text":"FolderA",
      "thumbnail":"",
      "attribute":2
    },
    {
      "text":"FolderB",
      "thumbnail":"",
      "attribute":2
    },
    {
      "text":"File r00",
      "thumbnail":"",
      "attribute":4
    },
    {
      "text":"File r01",
      "thumbnail":"",
      "attribute":4
    },
    {
      "text":"File r02",
      "thumbnail":"",
      "attribute":4
    }
  ]
}
```



## 12.1.3. Layer change

Use “setListControl” to move to Folder B layer. Confirm if the attribute of Response Parameter is selectable as described before.

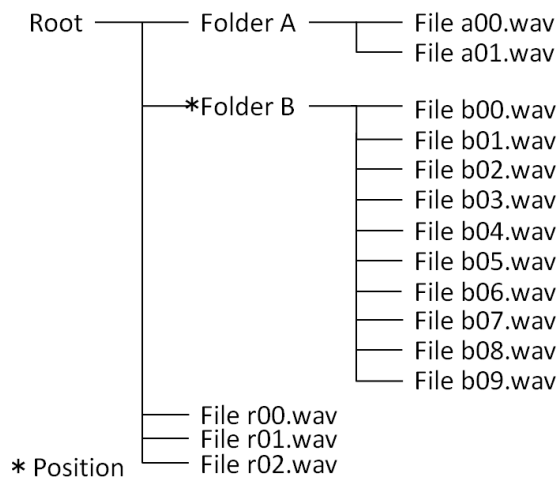
Request

`http://{host}/YamahaExtendedControl/v1/netusb/setListControl?list_id=main&type=select&index=1`

Response

```
{
  "response_code":0
}
```

Now the current position is moved to Folder B.



#### 12.1.4. Retrieving list information (In case max line number is more than 8)

Use “getListInfo” to get list information.

Here shows how to get it of Folder B.

##### Request

`http://{host}/YamahaExtendedControl/v1/netusb/getListInfo?input=usb&index=0&size=8&lang=en`

##### Response

```

{
  "response_code":0,
  "menu_layer":1,
  "max_line": 10 ← Max line number = 10
  "index":0,
  "playing_index":-1,
  "menu_name":"Folder B",
  "list_info":[
    {
      "text":"File b00",
      "thumbnail":"",
      "attribute":4
    },
    {
      "text":"File b01",
      "thumbnail":"",
      "attribute":4
    },
    {
      "text":"File b02",
      "thumbnail":"",
      "attribute":4
    }
  ]
}

```

```
        "text": "File b03",
        "thumbnail": "",
        "attribute": 4
    }
    {
        "text": "File b04",
        "thumbnail": "",
        "attribute": 4
    }
    {
        "text": "File b05",
        "thumbnail": "",
        "attribute": 4
    }
    {
        "text": "File b06",
        "thumbnail": "",
        "attribute": 4
    }
    {
        "text": "File b07",
        "thumbnail": "",
        "attribute": 4
    }
    ]
}
```

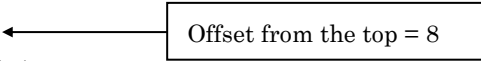
With this example, max\_line is 10 so the remaining information more than line 8 is needed.  
To do so, set “index = 8” in the Request Parameter.

#### Request

<http://{host}/YamahaExtendedControl/v1/netusb/getListInfo?input=usb&index=8&size=8&lang=en>

#### Response

```
{
    "response_code": 0,
    "menu_layer": 1,
    "max_line": 10
    "index": 8,
    "playing_index": -1,
    "menu_name": "Folder B",
    "list_info": [
        {
            "text": "File b08",
            "thumbnail": "",
            "attribute": 4
        },
        {
            "text": "File b09",
```





```

        "thumbnail":"","
        "attribute":4
    },
]
}

```

← Capability bit[2]: Play bit equal to 1

#### 12.1.5. Playback a file

Here shows how to specify and playback File b09. Confirm if the attribute of Response Parameter is playable. And the index of Request Parameter is 9.

##### Request

`http://{host}/YamahaExtendedControl/v1/netusb/setListControl?list_id=main&type=play&index=9`

##### Response

```

{
    "response_code":0
}

```

The File b09 in Folder B is now playback.

#### 12.1.6. Moving back one layer

Here shows an example of how to move back to one layer.

##### Request

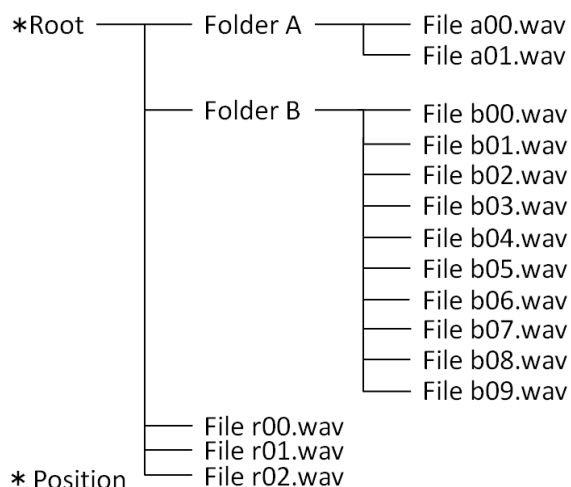
`http://{host}/YamahaExtendedControl/v1/netusb/setListControl  
?list_id=main&type=return`

##### Response

```

{
    "response_code":0
}

```



##### \*Note

getListInfo is the only command that blocks other commands execution. It may take up to 30 seconds to get the whole list information, and all other commands are not accepted all the while

## 12.2. Device Search

Here explains procedures of how to search MusicCast Devices

1. Specify MediaRenderer and issue M-Search.
2. Access to the URL of device description read from the Location header in the response.
3. Confirm tags shown below.

**<manufacturer>Yamaha Corporation</manufacturer>**

See if manufacturer name is "Yamaha Corporation"

**<yamaha:X\_device>**

See if there is this Yamaha tag

**<yamaha:X\_URLBase>http://192.168.10.103:80/</yamaha:X\_URLBase>**

Confirm device's IP address

**<yamaha:X\_yxcControlURL>/YamahaExtendedControl/v1/</yamaha:X\_yxcControlURL>**

Check the URL of YXC control

[Example of device description of a MusicCast Device]

```
<?xml version="1.0" encoding="utf-8" ?>
<root xmlns="urn:schemas-upnp-org:device-1-0"
xmlns:yamaha="urn:schemas-yamaha-com:device-1-0">
<specVersion>
  <major>1</major>
  <minor>0</minor>
</specVersion>
<device>
  <dlna:X_DLNAOC
xmlns:dlna="urn:schemas-dlna-org:device-1-0">DMR-1.50</dlna:X_DLNAOC>
  <deviceType>urn:schemas-upnp-org:device:MediaRenderer:1</deviceType>
  <friendlyName>Room A</friendlyName>
  <manufacturer>Yamaha Corporation</manufacturer>
  <manufacturerURL>http://www.yamaha.com/</manufacturerURL>
  <modelDescription>MusicCast</modelDescription>
  <modelName>WXC-50</modelName>
  <modelNumber>50</modelNumber>
  <modelURL>http://www.yamaha.com/</modelURL>
  <serialNumber>0DA27313</serialNumber>
  <UDN>uuid:9ab0c000-f668-11de-9976-00a0ded26c17</UDN>
  <iconList>
    <icon>
      <mimetype>image/jpeg</mimetype>
      <width>48</width>
      <height>48</height>
      <depth>24</depth>
      <url>/Icons/48x48.jpg</url>
```

```
</icon>
<icon>
  <mimetype>image/jpeg</mimetype>
  <width>120</width>
  <height>120</height>
  <depth>24</depth>
  <url>/Icons/120x120.jpg</url>
</icon>
<icon>
  <mimetype>image/png</mimetype>
  <width>48</width>
  <height>48</height>
  <depth>24</depth>
  <url>/Icons/48x48.png</url>
</icon>
<icon>
  <mimetype>image/png</mimetype>
  <width>120</width>
  <height>120</height>
  <depth>24</depth>
  <url>/Icons/120x120.png</url>
</icon>
</iconList>
<serviceList>
<service>
  <serviceType>urn:schemas-upnp-org:service:AVTransport:1</serviceType>
  <serviceId>urn:upnp-org:serviceId:AVTransport</serviceId>
  <SCPDURL>/AVTransport/desc.xml</SCPDURL>
  <controlURL>/AVTransport/ctrl</controlURL>
  <eventSubURL>/AVTransport/event</eventSubURL>
</service>
<service>
  <serviceType>urn:schemas-upnp-org:service:RenderingControl:1</serviceType>
  <serviceId>urn:upnp-org:serviceId:RenderingControl</serviceId>
  <SCPDURL>/RenderingControl/desc.xml</SCPDURL>
  <controlURL>/RenderingControl/ctrl</controlURL>
  <eventSubURL>/RenderingControl/event</eventSubURL>
</service>
<service>
  <serviceType>urn:schemas-upnp-org:service:ConnectionManager:1</serviceType>
  <serviceId>urn:upnp-org:serviceId:ConnectionManager</serviceId>
  <SCPDURL>/ConnectionManager/desc.xml</SCPDURL>
  <controlURL>/ConnectionManager/ctrl</controlURL>
  <eventSubURL>/ConnectionManager/event</eventSubURL>
</service>
</serviceList>
<presentationURL>http://192.168.10.103/</presentationURL>
</device>
<yamaha:X_device>
```

```
<yamaha:X_URLBase>http://192.168.10.103:80/</yamaha:X_URLBase>
<yamaha:X_serviceList>
<yamaha:X_service>

<yamaha:X_specType>urn:schemas-yamaha-com:service:X_YamahaRemoteControl:1</yamaha:X
_specType>
  <yamaha:X_controlURL>/YamahaRemoteControl/ctrl</yamaha:X_controlURL>
  <yamaha:X_unitDescURL>/YamahaRemoteControl/desc.xml</yamaha:X_unitDescURL>
  </yamaha:X_service>
  <yamaha:X_service>

<yamaha:X_specType>urn:schemas-yamaha-com:service:X_YamahaExtendedControl:1</yamaha
:X_specType>
  <yamaha:X_yxcControlURL>/YamahaExtendedControl/v1/</yamaha:X_yxcControlURL>
  <yamaha:X_yxcVersion>0801</yamaha:X_yxcVersion>
  </yamaha:X_service>
</yamaha:X_serviceList>
</yamaha:X_device>
</root>
```

---

Revision History

Rev.	Date	Contents	Author
1.00	7/7/2016	First edition	Eto