

**qsonix**  
TM

## Qsonix Integration Guide

*Version 2.9*

*Updated 9/1/2010*

## Table of Contents

Table of Contents .....	2
Revision History .....	4
Introduction .....	6
Serial Communications .....	6
Ethernet Communications .....	6
Command Structure .....	6
Playback Control Commands .....	7
Play [Zone] .....	7
PlaySeq [Zone][SequenceNumber] .....	7
Stop [Zone] .....	7
PauseSet [Zone #][Setting] .....	7
PauseToggle [Zone #] .....	7
Next [Zone #]{TrackCount} .....	8
Previous [Zone #]{TrackCount} .....	8
Cue [Zone #][CueLocation] .....	8
Volume [Zone #][VolumePercentage] .....	8
VolumeUp [Zone #][Amount] .....	8
VolumeDown [Zone #][Amount] .....	9
MuteToggle [Zone #] .....	9
MuteSet [Zone #][MuteValue] .....	9
RepeatToggle [Zone #] .....	9
RepeatSet [Zone #][RepeatValue] .....	10
Shuffle [Zone #] .....	10
NowPlaying [Zone #][Action] .....	10
Preview [Zone #][TrackID] .....	10
QueueItem [Zone #][Location][ObjectType][ObjectID] .....	10
QueueItemPlaylist [PlaylistID][Location][ObjectType][ObjectID] .....	11
MoveItemPlaylist [PlaylistID][FromSequenceNumber][ToSequenceNumber] .....	11
RemoveItemPlaylist [PlaylistID][SequenceNumber] .....	12
Shutdown .....	12
Restart .....	12
Two-way Commands .....	13
Get Zonename [Zone] {UserData} .....	15
Get SystemStatus {UserData} .....	15
Get ZoneVisibility .....	16

Get_Albums [PageNumber][PageSize]{UserData}.....	16
Get_Artists [PageNumber][PageSize]{UserData}.....	17
Get_Genres [PageNumber][PageSize]{UserData}.....	17
Get_Periods [PageNumber][PageSize]{UserData}.....	18
Get_Playlists [PageNumber][PageSize]{UserData}.....	19
GetFor_NowPlaying [Zone][TrackCount]{UserData}.....	19
GetFor_Search [SearchTerm][PageNumber][PageSize]{UserData}.....	20
GetFor_StylesForGenre [GenreID][PageNumber][PageSize]{UserData}.....	21
GetFor_AlbumsForArtist [ArtistID][PageNumber][PageSize]{UserData}.....	22
GetFor_AlbumsForGenre [GenreID][PageNumber][PageSize]{UserData}.....	22
GetFor_AlbumsForStyle [GenreID][PageNumber][PageSize]{UserData}.....	23
GetFor_AlbumsForPeriod [PeriodID][PageNumber][PageSize]{UserData}.....	24
GetFor_TracksForAlbum [AlbumID][PageNumber][PageSize]{UserData}.....	25
GetFor_TracksForPlaylist [PlaylistID][PageNumber][PageSize]{UserData}.....	25
GetDetails_Album [AlbumID]{UserData}.....	26
GetDetails_Artist [ArtistID]{UserData}.....	27
System Feedback .....	29
Feedback TrackChange [On/Off].....	29
Feedback TrackPosition [On/Off].....	29
Feedback RepeatChange [On/Off].....	30
Feedback PlaylistChange [On/Off].....	30
Feedback PlaystateChange [On/Off].....	31
Feedback Volume [On/Off].....	31
Feedback All [On/Off].....	31
Feedback GetValue .....	31
Cover Art Image Support .....	33
Deprecated commands .....	34
Queue [Zone#][Location].....	34

## Revision History

10/5/2005

- Response to GET NOWPLAYING command results changed. Added *Sequence* as a new returned field to the NOWPLAYING response. Also, *IsCurrent* field now returns either a 1 or 0 value instead of only returning if the value is 1.
- FEEDBACK CURRENTTRACK and FEEDBACK NEXTTRACK responses now include the *Sequence* field.
- Added new command “GET SYSTEMSTATUS” that can be used to check the availability of the system as well as retrieve the current system software version.

10/19/2005

- Added QUEUEITEM command to support directly queuing objects by ID number.

06/1/2007

- Added QUEUEITEMPLAYLIST command to support directly queuing objects by ID number to.
- Added another Feedback parameter (volume).

10/1/2007

- Removed old protocol no longer support in current firmware
- Support for Q110 (4-Zone) unit added

5/15/2008

- Added documentation for SetFeedback command.
- Removed SELECTPLAYLIST command – no longer implemented
- Removed SORTRESULTS command – no longer implemented
- Added full library browsing functions including “GET\_xxx”, “GETFOR\_xxx”, “GETDETAILS\_xxx” commands.

8/18/2008

- Updated document format
- Added instructions for TCP/IP support
- Added Cover Art Documentation

11/9/2008

- Added return fields to GET\_PERIODS: TotalTracks, TotalDuration, TotalDurationAsString
- Added new command: GETFILTERED\_ALBUMS
- Added AvailableAlpha to the response for GET\_ARTISTS and GETFILTERED\_ALBUMS

12/2/2008

- Added the REMOVEITEMPLAYLIST command.

5/13/2009

- Added the MOVEITEMPLAYLIST command.
- Added the PREVIEW command.
- Added the CUE command.

10/23/2009

- Updated the QUEUEITEM command to include a new object type of COMPOSER.
- Updated the QUEUEITEMPLAYLIST command to include a new object type of COMPOSER.
- Added the GETFOR\_ALBUMSFORCOMPOSER command.
- Added the GET\_COMPOSERS command.

8/30/2010

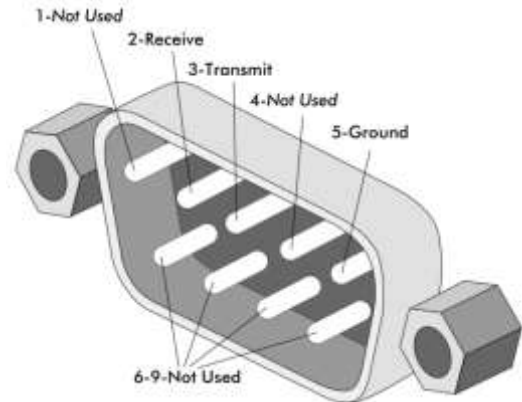
- Updated the GET\_ALBUMS command to return an additional field (AlbumTitleSort).
- Updated the GET\_ARTISTS command to return an additional field (ArtistNameSort).
- Updated the GET\_PERIODS command to return several additional fields (AlbumCount, TrackCount, Playlist Duration)
- Added versioning capability to allow access to newer versions of the protocol while maintaining backwards compatibility for existing applications. See the SET\_VERSION command for details.
- Updated several commands to return the AvailableAlpha field in the header to aid in creating applications that need to display a list of available letters in the resulting data set for indexing purposes. The following commands now include this field in the returned header: GET\_ARTISTS, GET\_ALBUMS, GET\_GENRES.

## Introduction

This document describes how to communicate with Qsonix Digital Media Management Systems using either RS-232 Serial or TCP/IP communications.

## Serial Communications

The rear panel of the Qsonix systems unit has a standard DB-9 serial port labeled “Serial”. Typical connection to this port requires use of a standard RS-232 null modem cable. Qsonix systems support communications between 9600 to 38400 baud, 8 data bits, 1 stop bit, no parity, no flow control. Serial configuration can be set by accessing the Integration tab under System Settings in the Qsonix software.



## Ethernet Communications

The Qsonix system also supports access via a TCP connection using the same communication protocol found in this document. The default port number for TCP communications is 5031. This port number can be configured by accessing the Integration tab under the System Settings in the Qsonix software.

## Command Structure

Commands sent to Qsonix systems will always end in a carriage return (ASCII 13). If a sequence of characters is sent to the unit without ending in a carriage return (an incomplete command), it will be ignored. The unit will wait up to 5 seconds for the command to be completed (by sending additional characters including a carriage return) before clearing the input buffer and ignoring the incomplete command.

A typical command will include a verb and one or more parameters. Parameters are always separated from each other and from the verb by a space (ASCII 32). For example, the command to begin playback for **Zone 1** is “**Play 1**” and the command to set the volume for **Zone 2** to 50% is “**Volume 2 50**”. Command verbs are not case sensitive but are typically specified in proper case for ease of readability. Some commands support optional parameters. Optional parameters are always last in the list of parameters and they can be provided to give a more specific command. For example, the Next command will cause the playback to skip to the next track in the now playing list for the specified zone. The Next command can also skip forward by more than one track with a single command (example: “**Next 2 5**” will skip playback of **Zone 2** forward by 5 tracks). When optional parameters are omitted, a default value that is specified in the command description will be used.

Some parameters will be full strings (such as a track title) that may include spaces within the parameter. These parameters will be enclosed in double-quotes. For example, the return status message indicating the currently playing track will return the track title currently playing: Feedback CurrentTrack 1 “*Here Comes the Sun*” indicates that **Zone 1** is currently playing the track *Here Comes the Sun*.

Most commands include a parameter that specifies on which zone the command should operate. The Zone parameter must be a numeric value equal corresponding to the zone the command should operate on. This must be less than the number of zones supported by the system (typically 1 to 4). A value of 0 can be used to indicate that the command should operate on the “current zone”. The current zone is the one that is current displayed in the lower right corner of the screen in the “Now Playing” area. The current zone in this area is also indicated by the color coding in the user interface.

The command reference below will show verbs in boldface and parameters will be shown by name in italics surrounded by square brackets (ex: [Zone #] or [Volume Percentage]). Optional parameters are shown in italics and surrounded by braces (ex: {# of tracks}).

## Playback Control Commands

### Play [Zone]

DESCRIPTION

Plays the currently selected track for the specified [Zone].

PARAMETERS

Zone # Specifies which zone should be acted upon. Must be 0 or a valid zone number.

### PlaySeq [Zone] [SequenceNumber]

DESCRIPTION

Skips playback to the specified sequence number in the play queue.

PARAMETERS

Zone # Specifies which zone should be acted upon. Must be 0 or a valid zone number.

SequenceNumber Playback will jump to this sequence number in the playback queue for the zone specified. Tracks in the playback queue are numbered sequentially starting at 0 (the first track is sequence number 0).

### Stop [Zone]

DESCRIPTION

Stops playback for the specified [Zone].

PARAMETERS

Zone # Specifies which zone should be acted upon. Must be 0 or a valid zone number.

### PauseSet [Zone #] [Setting]

DESCRIPTION

Sets the pause state of playback for the specified Zone.

PARAMETERS

Zone # Specifies which zone should be acted upon. Must be 0 or a valid zone number.

Setting Must be ON or OFF to indicate the new state of pause

### PauseToggle [Zone #]

DESCRIPTION

Toggles between paused and un-paused state for playback for the specified [Zone].

PARAMETERS

Zone # Specifies which zone should be acted upon. Must be 0 or a valid zone number.

## Next [Zone #] {TrackCount}

### DESCRIPTION

Skips playback forward by {TrackCount} tracks for the specified [Zone].

### PARAMETERS

- Zone # Specifies which zone should be acted upon. Must be 0 or a valid zone number.
- TrackCount Specifies the number of tracks to skip forward. This parameter is optional and the default value is 1.

## Previous [Zone #] {TrackCount}

### DESCRIPTION

Skips playback backward by {TrackCount} tracks for the specified [Zone].

### PARAMETERS

- Zone # Specifies which zone should be acted upon. Must be 0 or a valid zone number.
- TrackCount Specifies the number of tracks to skip forward. This parameter is optional and the default value is 1.

## Cue [Zone #] [CueLocation]

### DESCRIPTION

Cues into the current track to the number of seconds specified by [CueLocation].

### PARAMETERS

- Zone # Specifies which zone should be acted upon. Must be 0 or a valid zone number.
- CueLocation Specifies where the current cue point of the current track should be set. This should be an integer value specifying the number of seconds from the start of the track.

## Volume [Zone #] [VolumePercentage]

### DESCRIPTION

Sets the volume for the specified [Zone] to the [VolumePercentage] specified. To turn the volume up or down (not set a discreet value, see the VolumeUp or VolumeDown command instead).

### PARAMETERS

- Zone # Specifies which zone should be acted upon. Must be 0 or a valid zone number.
- VolumePercentage Specifies the new volume setting desired in terms of a percent of full volume. The value must be an integer value between 0 and 100.

## VolumeUp [Zone #] [Amount]

### DESCRIPTION

Increases the volume for the specified [Zone] by the [Amount] specified.

The VolumeUp command is used to set the volume to a value relative to the current setting, so the Amount value must be between 1 and 100. Note that if the value specified would cause the volume

setting to be greater than 100, the volume will be set to 100 instead. For instance, if the volume is currently set to 50% and the VolumeUp command is issued with an *Amount* parameter of 70, the volume will be set to 100% (instead of 120%).

#### PARAMETERS

Zone # Specifies which zone should be acted upon. Must be 0 or a valid zone number.

Amount Specifies the amount of volume change desired. The playback volume change amount is specified in terms of percent (0 to 100%).

### VolumeDown [*Zone #*] [*Amount*]

#### DESCRIPTION

Decreases the volume for the specified [*Zone*] by the [*Amount*] specified.

The VolumeDown command is used to set the volume to a value relative to the current setting, so the *Amount* value must be between 1 and 100. Note that if the value specified would cause the volume setting to be greater than 100, the volume will be set to 100 instead. For instance, if the volume is currently set to 50% and the VolumeDown command is issued with an *Amount* parameter of 70, the volume will be set to 100% (instead of 120%).

#### PARAMETERS

Zone # Specifies which zone should be acted upon. Must be 0 or a valid zone number.

Amount Specifies the amount of volume change desired. The playback volume change amount is specified in terms of percent (0 to 100%).

### MuteToggle [*Zone #*]

#### DESCRIPTION

Toggles the mute setting for the specified [*Zone*]. For a discreet version of this command, see MuteSet.

#### PARAMETERS

Zone # Specifies which zone should be acted upon. Must be 0 or a valid zone number.

### MuteSet [*Zone #*] [*MuteValue*]

#### DESCRIPTION

Sets the mute setting for the specified [*Zone*] to a discreet value specified by the [*MuteValue*] parameter.

#### PARAMETERS

Zone # Specifies which zone should be acted upon. Must be 0 or a valid zone number.

MuteValue Specifies the desired new state of mute. The value must be either *On* or *Off*.

### RepeatToggle [*Zone #*]

#### DESCRIPTION

Toggles the repeat setting for the specified [*Zone*]. For a discreet version of this command, see RepeatSet.

#### PARAMETERS

Zone # Specifies which zone should be acted upon. Must be 0 or a valid zone number.

## RepeatSet [Zone #] [RepeatValue]

### DESCRIPTION

Sets the repeat setting for the specified [Zone] to a discreet value specified by the [RepeatValue] parameter.

### PARAMETERS

Zone # Specifies which zone should be acted upon. Must be 0 or a valid zone number.  
RepeatValue Specifies the desired new state of repeat. The value must be either *On* or *Off*.

## Shuffle [Zone #]

### DESCRIPTION

Shuffles the now playing list of tracks for the specified [Zone]. The shuffle operation will randomly resort all tracks in the now playing list. If the zone is currently playing a track, that track will be moved to the top of the list and will continue playing while all other tracks below it will be shuffled randomly.

### PARAMETERS

Zone # Specifies which zone should be acted upon. Must be 0 or a valid zone number.

## NowPlaying [Zone #] [Action]

### DESCRIPTION

The NowPlaying command is used for actions that affect the Now Playing playlist for the specified [Zone].

### PARAMETERS

Zone # Specifies which zone should be acted upon. Must be 0 or a valid zone number.  
Action Specifies what action should be taken on the specified zone's now playing playlist. Action must be one of the following values:  
*ClearAll* – Clears all tracks from the playlist  
*ClearPlayed* – Clears all tracks that have completed playing from the playlist

## Preview [Zone #] [TrackID]

### DESCRIPTION

Play a short preview of the track specified by [TrackID] in the zone specified by [Zone]. When the preview is completed, playback will return to its previous state.

### PARAMETERS

Zone # Specifies which zone should be acted upon. Must be 0 or a valid zone number.  
TrackID Specifies the track that should be previewed.

## QueueItem [Zone #] [Location] [ObjectType] [ObjectID]

### DESCRIPTION

Allows adding of items to the Now Playing playlist for the specified [Zone]. The location in the current playlist that the item will be inserted is determined by the value of the [Location] parameter.

PARAMETERS

Zone #	Specifies which zone should be acted upon. Must be 0 or a valid zone number.
Location	Specifies where the item specified should be added to the now playing queue. <i>Location</i> must be one of the following values:  <i>End</i> – Add selected item to the end of the playlist <i>Next</i> – Add selected item to the playlist immediately after the current track <i>Now</i> – Add selected item to the playlist immediately after the current track and then immediately call track next so that the queued track will start playing immediately. <i>Clear</i> – Clear the playlist and then add the selected item. # - If a number is specified as the location parameter, then the item(s) will be inserted into the now playing playlist at the index specified. The index is 0-based.
ObjectType	Selection of exactly what item is to be insert into the playlist is further determined by the <i>ObjectType</i> and <i>ObjectID</i> parameters. The <i>ObjectType</i> parameter must be one of the following:  <i>Album, Artist, Genre, Track, Playlist, Composer</i>
ObjectID	The <i>ObjectID</i> parameter is then used to indicate which item of the specified <i>ObjectType</i> will be inserted into the playlist.

### QueueItemPlaylist [PlaylistID] [Location] [ObjectType] [ObjectID]

DESCRIPTION

Allows adding of items to the Now Playing playlist for the specified [Zone]. The location in the current playlist that the item will be inserted is determined by the value of the [Location] parameter.

PARAMETERS

PlaylistID	Specifies which playlist should have the items added.
Location	Specifies where the item specified should be added to the specified playlist. <i>Location</i> must be one of the following values:  <i>End</i> – Add selected item to the end of the playlist <i>Clear</i> – Clear the playlist and then add the selected item.
ObjectType	Selection of exactly what item is to be insert into the playlist is further determined by the <i>ObjectType</i> and <i>ObjectID</i> parameters. The <i>ObjectType</i> parameter must be one of the following:  <i>Album, Artist, Genre, Track, Playlist, Composer</i>
ObjectID	The <i>ObjectID</i> parameter is then used to indicate which item of the specified <i>ObjectType</i> will be inserted into the playlist.

### MoveItemPlaylist [PlaylistID] [FromSequenceNumber] [ToSequenceNumber]

DESCRIPTION

Moves the track with the specified [FromSequenceNumber] from the playlist specified by [PlaylistID] to a new location specified by [ToSequenceNumber].

PARAMETERS

PlaylistID	Specifies which playlist should have the items added. This can also be provided as a string value formatted as “NPx” where NP represents “Now Playing” and ‘x’ is the zone number. For example, to remove an item from the now playing playlist for zone 3, the PlaylistID parameter
------------	--

can be supplied as NP3.

FromSequenceNumber The sequence number of the item to be moved within the playlist.

ToSequenceNumber The sequence number within the playlist where the item should be moved to.

## RemoveItemPlaylist [*PlaylistID*] [*SequenceNumber*]

### DESCRIPTION

Removes the track with the specified [*SequenceNumber*] from the playlist specified by [*PlaylistID*].

### PARAMETERS

PlaylistID Specifies which playlist should have the items added. This can also be provided as a string value formatted as “NPx” where NP represents “Now Playing” and ‘x’ is the zone number. For example, to remove an item from the now playing playlist for zone 3, the PlaylistID parameter can be supplied as NP3.

SequenceNumber The sequence number of the item to be removed from the playlist.

## Shutdown

### DESCRIPTION

Remotely turn off the power to the unit.

## Restart

### DESCRIPTION

Remotely restart the system.

## Set\_Version [*VersionNumber*] {*UserData*}

### DESCRIPTION

Configures the system to communicate using the specified version number of the communications protocol. If this command is never called by the client, the system will default to communicating in version 2.8. Using newer versions of the protocol will provide additional functionality and information.

### PARAMETERS

VersionNumber Specifies what version of the protocol should be used for IP and Serial communications. *Int32*

## Two-way Commands

Various commands are provided for the purposes of retrieving data from the server or to receive feedback of current server status. All commands that are requests of the server to return data will conform to a standard response format. The response will always include a header that can be used to identify what type of data is being returned to the client. Data returned will use standard delimiters to aide in parsing the resulting information. A group delimiter (ASCII 29 – shown in this document as the character `*`) will be used to denote the end of the header. A row delimiter (ASCII 30 – shown in this document as the character `↵`) will be used to divide distinct rows of results (as in the case of a multi-row response). A field delimiter (ASCII 31 - shown in this document as the character `♦`) will be used to delimit multiple values in a response or within a row of a response. In some special cases, a single field in a response can contain multiple values (such as the `OutputsPlayingThisTrack` field returned from the `GETFOR_TRACKSFORPLAYLIST` command). The multi-value delimiter (ASCII 3) will be used to divide individual values within a multi-valued field in the response.

Commands that return a multi-row response also support paging operations. This feature allows the client to request data in pages of a specified number of rows and the server will manage the current index location and support next/back functions from the client. This is designed to support clients that are only able to display a limited number of items on the screen at a time and will provide the user with page up / page down buttons to navigate the list.

For example, the command “GET PLAYLISTS” can be used to return a list of playlists from the server. The response for this command will include a header that will include the data type being returned “PLAYLISTS”, followed by the current page number and then by the total page count. These values will be separated by the field delimiter and terminated by the group delimiter. For example, the header portion of this response would be formatted as follows: `PLAYLISTS♦3♦12*`

Following the header of this response, the actual list of playlists for the specified page would appear as a list of rows. Each row will contain multiple values (fields) containing the details of the playlist (including PlaylistID, Playlist Name, Number of Tracks in Playlist, Total Duration of Playlist (in seconds), Total Duration of Playlists (as a time-formatted string). For example, each row of the data portion of the response would be formatted as follows: `327♦Jazz / Blues Mix♦115♦31050♦8:37:30↵`

Note that the entire response packet will then be delimited by a carriage return (ASCII 13).

The typical format of these data retrieval commands will be “GET [*datatype*] [*paging command*]”. Currently, only two types of *datatypes* are supported, PLAYLISTS and NOWPLAYING. The *paging command* can be optionally used to retrieve additional pages and supported commands include: NEXT, PREVIOUS, FIRST, LAST, REFRESH. In addition, the paging command CURRENT can be used in conjunction with the NOWPLAYING *datatype* to retrieve the page that includes the current track in the now playing list. All variations of the supported commands are listed below.

The GET\_xxx commands can be used to retrieve a variety of metadata and allows browsing of the full media library. Each of these commands accept a PageNumber and PageSize parameter. If a PageNumber is passed that is greater than the maximum number of pages, the last page will be returned. The PageNumber parameter can also be passed as a single letter, causing the results to return the first page that contains results starting with that letter. For example, the command “GET\_ALBUMS G 30” will return the first page of 30 albums that has album titles starting with the letter G. Note that the items starting with G will not necessarily appear as the first item on the page, but the page will contain the start of the items that begin with the letter G.

### Data Types

The data type of each field in responses is noted in the description of the field. Possible values are listed below:

- String(x) – A string value with a maximum length of x
- Int32 – a 32-bit integer value (maximum value of 2,147,483,647)
- Int16 – a 16-bit integer value (maximum value of 32,767)
- uint8 – an unsigned 8-bit integer value (maximum value of 256)

### *User Data*

Each of the commands in this section support an optional UserData parameter. This is always the last parameter in the command and must be a string. The UserData parameter cannot contain spaces. The data supplied in the UserData parameter will be returned in the response header as the last field in the header. If the UserData is not supplied in the request command, an empty field will be returned in the response header.

## Get Zonename [Zone] {UserData}

### DESCRIPTION

Used to retrieve the user-specified name assigned to each of the zones in the system.

### PARAMETERS

Zone Specifies the zone to retrieve the name for. Must be a valid zone number.  
*uInt8*

### RESPONSE DEFINITION

ZONENAME♦[UserData]\*[ZoneNumber]♦[ZoneName]↻

ZoneNumber The number of the zone. *Int16*

ZoneName The user-specified name of the zone. *String(50)*

## Get SystemStatus {UserData}

### DESCRIPTION

Used to check the availability of the server.

### PARAMETERS

Zone Specifies the zone to retrieve the name for. Must be a valid zone number.  
*uInt8*

### RESPONSE DEFINITION

SYSTEMSTATUS♦[UserData]\*[Status]♦[SoftwareVersion]♦[BuildNumber]♦[DeviceID]♦[SerialNumber]♦[IPAddress]♦[MACAddress]♦[SubnetMask]♦[DefaultGateway]♦[PrimaryDNS]♦[SecondaryDNS]♦[AvailableSpace]♦[TrackCount]♦[AlbumCount]♦[ArtistCount]♦[GenreCount]♦[StylesCount]♦[PlaylistCount]♦[FriendlyName]↻

Status Currently supports only “OK” indicating that the system is available.  
*String(2)*

SoftwareVersion The current software version running on the system *String(20)*

BuildNumber The current software build number running on the system.  
*String(20)*

DeviceID The system Device Identifier. *String(36)*

SerialNumber The serial number of the system. *Int32*

IPAddress The current IP address of the system. *String(15)*

MACAddress The current Media Access Control address (MAC) of the system.  
*String(17)*

SubnetMask The current subnet mask of the system. *String(15)*

DefaultGateway The current default gateway of the system. *String(15)*

PrimaryDNS The current primary DNS entry in the system. *String(15)*

SecondaryDNS The current secondary DNS entry in the system. *String(15)*

AvailableSpace Amount of storage space remaining on the system, formatted as a readable string (ex: 321MB). *String(10)*

TrackCount Total number of tracks currently stored on the system. *Int32*

AlbumCount Total number of albums currently stored on the system. *Int32*

ArtistCount Total number of artists currently stored on the system. *Int32*

GenreCount	Total number of genres currently stored on the system. <i>Int32</i>
StylesCount	Total number of styles currently stored on the system. <i>Int32</i>
PlaylistCount	Total number of playlists currently stored on the system. <i>Int32</i>
FriendlyName	The friendly name of the machine set by the user in the configuration area. <i>String(50)</i>

## Get ZoneVisibility

### DESCRIPTION

Used to check the availability of a zone of the server.

### PARAMETERS

Zone	Specifies the zone to retrieve the name for. Must be a valid zone number. <i>uInt8</i>
------	--

### RESPONSE DEFINITION

ZONESVISIBILITY\**[VisibilityStatus]*↻

VisibilityStatus	A string containing a one or zero for each zone (1=visible) Example: 0011 would indicate that Zones 3 and 4 should be visible / active. <i>String(4)</i>
------------------	--

## Get\_Albums [PageNumber] [PageSize] {UserData}

### DESCRIPTION

Retrieves a list of all albums in the library.

### PARAMETERS

PageNumber	Specifies the page number to be retrieved. <i>Int32</i>
PageSize	Specifies the number of rows in each page. <i>Int32</i>

### RESPONSE DEFINITION

GET\_ALBUMS\**[PageNumber]*♦*[TotalPages]*♦*[TotalAlbums]*♦*[UserData]*♦*[AvailableAlpha]*\**[AlbumID]*♦*[AlbumTitle]* ♦*[AlbumArtistID]*♦*[AlbumArtistName]*♦*[Year]*♦*[TrackCount]*♦*[AlbumTitleSort]*↻

PageNumber	The page number being returned. <i>Int32</i>
TotalPages	Indicates how many pages are available at the page size specified. <i>Int32</i>
TotalAlbums	Count of the number of albums in the library. <i>Int32</i>
AlbumID	Unique album identifier for the album being returned. <i>Int32</i>
AlbumTitle	Title of the album. <i>String(100)</i>
AlbumArtistID	Unique artist identifier for the artist that is the primary artist on the album. <i>Int32</i>
AlbumArtistName	Name of the primary artist for the album. <i>String(255)</i>
Year	Year the album was released. <i>Int16</i>
TrackCount	Number of tracks associated with the album. <i>Int32</i>
AlbumTitleSort	Sortable title of the album. <i>String(100)</i>
AvailableAlpha	A list of letters indicating what initials are contained within the reply. For example, if there is at least one value in the response that

AlbumTitleSort begins with the letter “A”, it will appear in this string. All entries that start with a non-alpha character will be represented in the string by a the pound symbol (“#”). *String(30)*

A version of the album title that can be used for sorting. For example, if the AlbumTitle field returns “The Nightfly”, this field will return “Nightfly”. *String(255)*

## Get\_Artists [PageNumber] [PageSize] {UserData}

### DESCRIPTION

Retrieves a list of all artists in the library.

### PARAMETERS

PageNumber Specifies the page number to be retrieved. *Int32*

PageSize Specifies the number of rows in each page. *Int32*

### RESPONSE DEFINITION

GET\_ARTISTS♦[PageNumber]♦[TotalPages]♦[TotalArtists]♦[UserData]♦[AvailableAlpha]♦[ArtistID]♦[ArtistName]♦[AlbumCount]♦[ArtistNameSort] ↗

PageNumber The page number being returned. *Int32*

TotalPages Indicates how many pages are available at the page size specified. *Int32*

TotalArtists Count of the number of artists in the library. *Int32*

ArtistID Unique artist identifier for the artist being returned. *Int32*

ArtistName Name of the artist. *String(255)*

AlbumCount Number of albums associated with this artist. *Int32*

ArtistNameSort A version of the artist name that can be used for sorting. For example, if the ArtistName field returns “The Beatles”, this field will return “Beatles”. *String(255)*

AvailableAlpha A list of letters indicating what initials are contained within the reply. For example, if there is at least one value in the response that begins with the letter “A”, it will appear in this string. All entries that start with a non-alpha character will be represented in the string by a the pound symbol (“#”). *String(30)*

## Get\_Composers [PageNumber] [PageSize] {UserData}

### DESCRIPTION

Retrieves a list of all composers in the library.

### PARAMETERS

PageNumber Specifies the page number to be retrieved. *Int32*

PageSize Specifies the number of rows in each page. *Int32*

### RESPONSE DEFINITION

GET\_ARTISTS♦[PageNumber]♦[TotalPages]♦[TotalComposers]♦[UserData]♦[ArtistID]♦[ComposerName]♦[AlbumCount] ↗

PageNumber	The page number being returned. <i>Int32</i>
TotalPages	Indicates how many pages are available at the page size specified. <i>Int32</i>
TotalComposers	Count of the number of composers in the library. <i>Int32</i>
ComposerArtistID	Unique artist identifier for the composer being returned. <i>Int32</i>
ComposerName	Name of the composer. <i>String(255)</i>
AlbumCount	Number of albums associated with this artist. <i>Int32</i>

### Get\_Genres [PageNumber] [PageSize] {UserData}

#### DESCRIPTION

Retrieves a list of all genres in the library.

#### PARAMETERS

PageNumber	Specifies the page number to be retrieved. <i>Int32</i>
PageSize	Specifies the number of rows in each page. <i>Int32</i>

#### RESPONSE DEFINITION

GET\_GENRES♦[PageNumber]♦[TotalPages]♦[TotalGenres]♦[UserData]★[GenreID]♦[GenreName]♦[AlbumCount]♦[StyleCount]♦[AvailableAlpha]↻

PageNumber	The page number being returned. <i>Int32</i>
TotalPages	Indicates how many pages are available at the page size specified. <i>Int32</i>
TotalGenres	Count of the number of genres in the library. <i>Int32</i>
GenreID	Unique genre identifier for the genre being returned. <i>Int32</i>
GenreName	Name of the genre. <i>String(100)</i>
AlbumCount	Number of albums associated with this genre. <i>Int32</i>
StyleCount	Number of styles associated with this genre. <i>Int32</i>
AvailableAlpha	A list of letters indicating what initials are contained within the reply. For example, if there is at least one value in the response that begins with the letter “A”, it will appear in this string. All entries that start with a non-alpha character will be represented in the string by a the pound symbol (“#”). <i>String(30)</i>

### Get\_Periods [PageNumber] [PageSize] {UserData}

#### DESCRIPTION

Retrieves a list of all years / decades (periods) in the library.

#### PARAMETERS

PageNumber	Specifies the page number to be retrieved. <i>Int32</i>
PageSize	Specifies the number of rows in each page. <i>Int32</i>

#### RESPONSE DEFINITION

GET\_PERIODS♦[PageNumber]♦[TotalPages]♦[TotalPeriods]♦[UserData]★[PeriodID]♦[DisplayString]♦[AlbumCount]♦[TrackCount]♦[Duration]♦[DurationAsString]↻

PageNumber	The page number being returned. <i>Int32</i>
------------	--

TotalPages	Indicates how many pages are available at the page size specified. <i>Int32</i>
TotalPeriods	Count of the number of periods in the library. <i>Int32</i>
PeriodID	Unique period identifier for the period being returned. Distinct years have a period ID that is simply the year. Decades have a period ID that prepends a 1 to the decade (for example, 1980s has a period ID of 11980). <i>Int16</i>
DisplayString	A displayable version of the PeriodID. This translates decades back to a displayable form such as “1980s”. <i>String(5)</i>
AlbumCount	The number of albums associated with this period. <i>Int32</i>
TrackCount	The total number of tracks associated with this period. <i>Int32</i>
TotalDuration	The total duration of all tracks associated with this period. <i>Int32</i>
TotalDurationAsString	The total duration of all tracks associated with this period. This is returned as a formatted string value (ex: 1:23:45). <i>String(20)</i>

### Get\_Playlists [PageNumber] [PageSize] {UserData}

#### DESCRIPTION

Retrieves a list of playlists in the system.

#### PARAMETERS

PageNumber	Specifies the page number to be retrieved. <i>Int32</i>
PageSize	Specifies the number of rows in each page. <i>Int32</i>

#### RESPONSE DEFINITION

GET\_PLAYLISTS♦[PageNumber]♦[TotalPages]♦[TotalPlaylists]♦[UserData]\*[PlaylistID]♦[PlaylistName]♦[DurationInSeconds]♦[DurationAsString]♦[TrackCount]↔

PageNumber	The page number being returned. <i>Int32</i>
TotalPages	Indicates how many pages are available at the page size specified. <i>Int32</i>
TotalPlaylists	Count of the number of playlists in the library. <i>Int32</i>
PlaylistID	Unique playlist identifier for the playlist being returned. <i>Int32</i>
PlaylistName	The name of the playlist. <i>String(255)</i>
DurationInSeconds	Total duration of all tracks in the playlist. This is returned as an integer value in seconds. <i>Int32</i>
DurationAsString	Total duration of all tracks in the playlist. This is returned as a formatted string value (ex: 1:23:45). <i>String(10)</i>
TrackCount	Total number of tracks in the playlist. <i>Int32</i>

### GetFor\_NowPlaying [Zone] [TrackCount] {UserData}

#### DESCRIPTION

Retrieves a list of tracks from the now playing playlist for the specified *Zone*. The first row returned in the result will be the current track for that zone. Based on the value of *TrackCount*, additional rows will be returned to indicate tracks that are queued up to play in the future. If there are no future tracks or less than *TrackCount* number of future tracks, then less rows may be returned.

#### PARAMETERS

Zone	Now Playing tracks will be returned for this zone. <i>uInt8</i>
TrackCount	Specifies the number of rows to return. This is how many tracks in the future will be returned (includes the current track). <i>Int32</i>

#### RESPONSE DEFINITION

GET\_NOWPLAYING♦[Zone]♦[TrackCount]♦[UserData]\*[TrackID]♦[TrackTitle]♦[SequenceNumber]♦[ArtistID]♦[ArtistName]♦[AlbumID]♦[AlbumTitle]♦[TrackDurationInSeconds]♦[TrackDurationAsString]

Zone	Now Playing tracks will be returned for this zone. <i>uInt8</i>
TrackCount	Specifies the number of rows returned. This will normally be the same as the <i>TrackCount</i> value passed in unless there are less than the specified number of tracks in the future of the now playing queue for the specified <i>Zone</i> . <i>Int32</i>
TrackID	The unique track ID for the track being returned. <i>Int32</i>
TrackTitle	The track title. <i>String(512)</i>
SequenceNumber	This number is unique within a playlist and indicates this tracks order within the playlist. Since a playlist can contain a track more than once, this provides an alternative to track ID to uniquely identify a particular instance of a track within a playlist. <i>Int32</i>
ArtistID	The unique artist ID of the primary artist for the track being returned. <i>Int32</i>
ArtistName	The name of the primary artist for the track being returned. <i>String(255)</i>
AlbumID	The unique album ID of the album that this track is associated with. <i>Int32</i>
AlbumTitle	The title of the album that the track is associated with. <i>String(100)</i>
TrackDurationInSeconds	Duration of the track. This is returned as an integer value in seconds. <i>Int32</i>
DurationAsString	Duration of the track. This is returned as a formatted string value (ex: 1:23:45). <i>String(10)</i>

### GetFor\_Search [SearchTerm] [PageNumber] [PageSize] {UserData}

#### DESCRIPTION

Performs a keyword search on all items in the library and returns a list of objects that have a string match. Currently, this search function searches for matches in all album titles, artist names, track titles and genres. The results are returned as a composite result, with the *ObjectType* response field indicating the type of object that contained the search keyword.

#### PARAMETERS

SearchTerm	A string value that is to be used for the search. <i>String(255)</i>
PageNumber	Specifies the page number to be retrieved. <i>Int32</i>
PageSize	Specifies the number of rows in each page. <i>Int32</i>

#### RESPONSE DEFINITION

GETFOR\_SEARCH♦[SearchTerm]♦[PageNumber]♦[TotalPages]♦[TotalResults]♦[UserData]\*[ObjectType]♦[ObjectID]♦[MatchString]↻

SearchTerm	The search term that was searched for. <i>String(255)</i>
PageNumber	The page number being returned. <i>Int32</i>
TotalPages	Indicates how many pages are available at the page size specified. <i>Int32</i>
TotalResults	Count of the number of matches available. <i>Int32</i>
ObjectType	The type of object that this search result pertains to. Valid values are: TRACK, ALBUM, ARTIST, GENRE <i>String(20)</i>
ObjectID	The unique identifier of the object matched. <i>Int32</i>
MatchString	The string value that was matched and contained the keyword. <i>String(512)</i>
ExtendedData1	This contains additional data about the search result. The data contained depends on the ObjectType being returned as follows:  TRACK: Album Title ALBUM: Album Artist Name ARTIST: <not used> GENRE: <not used>
ExtendedData2	This contains additional data about the search result. The data contained depends on the ObjectType being returned as follows:  TRACK: Album Artist Name ALBUM: <not used> ARTIST: <not used> GENRE: <not used>

## GetFor\_StylesForGenre [GenreID] [PageNumber] [PageSize] {UserData}

### DESCRIPTION

Retrieves a list of styles for the specified genre.

### PARAMETERS

GenreID	The unique genre ID to retrieve a list of styles for. <i>Int32</i>
PageNumber	Specifies the page number to be retrieved. <i>Int32</i>
PageSize	Specifies the number of rows in each page. <i>Int32</i>

### RESPONSE DEFINITION

GETFOR\_STYLESFORGENRE ♦ [GenreID] ♦ [PageNumber] ♦ [TotalPages] ♦ [TotalStyles] ♦ [UserData] ♦ [GenreID] ♦ [GenreName] ♦ [AlbumCount] ↗

GenreID	The genre ID of the parent of the styles being returned. <i>Int32</i>
PageNumber	The page number being returned. <i>Int32</i>
TotalPages	Indicates how many pages are available at the page size specified. <i>Int32</i>
TotalStyles	Count of the number of styles returned. <i>Int32</i>
GenreID	The unique genre ID for the style being returned. Genres and styles are essentially the same data type. <i>Int32</i>
GenreName	The name of the style. <i>String(100)</i>
AlbumCount	The number of albums that are associated with the style. <i>Int32</i>

## GetFor\_AlbumsForArtist [ArtistID] [PageNumber] [PageSize] {UserData}

### DESCRIPTION

Retrieves a list of albums for the specified artist.

### PARAMETERS

ArtistID	The unique artist ID to retrieve a list of albums for. <i>Int32</i>
PageNumber	Specifies the page number to be retrieved. <i>Int32</i>
PageSize	Specifies the number of rows in each page. <i>Int32</i>

### RESPONSE DEFINITION

GETFOR\_ALBUMSFORARTIST♦[ArtistID]♦[PageNumber]♦[TotalPages]♦[TotalAlbums]♦[UserData]♦\*[AlbumID]♦[AlbumTitle]♦[AlbumArtistID]♦[AlbumArtistName]♦[Year]♦[TrackCount]↵

ArtistID	The artist ID of the albums being returned. <i>Int32</i>
PageNumber	The page number being returned. <i>Int32</i>
TotalPages	Indicates how many pages are available at the page size specified. <i>Int32</i>
TotalAlbums	Count of the number of albums returned. <i>Int32</i>
AlbumID	The unique album ID for the album being returned. <i>Int32</i>
AlbumTitle	The album title. <i>String(100)</i>
AlbumArtistID	Unique artist identifier for the artist that is the primary artist on the album. <i>Int32</i>
AlbumArtistName	Name of the primary artist for the album. <i>String(255)</i>
Year	The release year of the album. <i>Int16</i>
TrackCount	Total number of tracks associated with the album. <i>Int32</i>

## GetFor\_AlbumsForComposer [ComposerID] [PageNumber] [PageSize] {UserData}

### DESCRIPTION

Retrieves a list of albums that include at least one track by the specified composer.

### PARAMETERS

ComposerID	The unique composer ID to retrieve a list of albums for. <i>Int32</i>
PageNumber	Specifies the page number to be retrieved. <i>Int32</i>
PageSize	Specifies the number of rows in each page. <i>Int32</i>

### RESPONSE DEFINITION

GETFOR\_ALBUMSFORCOMPOSER♦[ComposerArtistID]♦[PageNumber]♦[TotalPages]♦[TotalAlbums]♦[UserData]♦\*[AlbumID]♦[AlbumTitle]♦[AlbumArtistID]♦[AlbumArtistName]♦[Year]♦[TrackCount]↵

ComposerArtistID	The artist ID of the albums being returned. <i>Int32</i>
PageNumber	The page number being returned. <i>Int32</i>
TotalPages	Indicates how many pages are available at the page size specified. <i>Int32</i>
TotalAlbums	Count of the number of albums returned. <i>Int32</i>
AlbumID	The unique album ID for the album being returned. <i>Int32</i>

AlbumTitle	The album title. <i>String(100)</i>
AlbumArtistID	Unique artist identifier for the artist that is the primary artist on the album. <i>Int32</i>
AlbumArtistName	Name of the primary artist for the album. <i>String(255)</i>
Year	The release year of the album. <i>Int16</i>
TrackCount	Total number of tracks associated with the album. <i>Int32</i>

### GetFor\_AlbumsForGenre [GenreID] [PageNumber] [PageSize] {UserData}

#### DESCRIPTION

Retrieves a list of albums for the specified genre.

#### PARAMETERS

GenreID	The unique genre ID to retrieve a list of albums for. <i>Int32</i>
PageNumber	Specifies the page number to be retrieved. <i>Int32</i>
PageSize	Specifies the number of rows in each page. <i>Int32</i>

#### RESPONSE DEFINITION

GETFOR\_ALBUMSFORGENRE♦[GenreID]♦[PageNumber]♦[TotalPages]♦[TotalAlbums]♦[UserData]  
 \* [AlbumID]♦[AlbumTitle]♦[AlbumArtistID]♦[AlbumArtistName]♦[Year]♦[TrackCount]↻

GenreID	The genre ID of the albums being returned. <i>Int32</i>
PageNumber	The page number being returned. <i>Int32</i>
TotalPages	Indicates how many pages are available at the page size specified. <i>Int32</i>
TotalAlbums	Count of the number of albums returned. <i>Int32</i>
AlbumID	The unique album ID for the album being returned. <i>Int32</i>
AlbumTitle	The album title. <i>String(100)</i>
AlbumArtistID	Unique artist identifier for the artist that is the primary artist on the album. <i>Int32</i>
AlbumArtistName	Name of the primary artist for the album. <i>String(255)</i>
Year	The release year of the album. <i>Int16</i>
TrackCount	Total number of tracks associated with the album. <i>Int32</i>

### GetFor\_AlbumsForStyle [GenreID] [PageNumber] [PageSize] {UserData}

#### DESCRIPTION

Retrieves a list of albums for the specified style.

#### PARAMETERS

GenreID	The unique genre ID to retrieve a list of albums for. Note that styles also use GenreID. <i>Int32</i>
PageNumber	Specifies the page number to be retrieved. <i>Int32</i>
PageSize	Specifies the number of rows in each page. <i>Int32</i>

RESPONSE DEFINITION

GETFOR\_ALBUMSFORSTYLE♦[GenreID]♦[PageNumber]♦[TotalPages]♦[TotalAlbums]♦[UserData]♦[\*][AlbumID]♦[AlbumTitle]♦[AlbumArtistID]♦[AlbumArtistName]♦[Year]♦[TrackCount]↷

GenreID	The genre ID of the albums being returned. <i>Int32</i>
PageNumber	The page number being returned. <i>Int32</i>
TotalPages	Indicates how many pages are available at the page size specified. <i>Int32</i>
TotalAlbums	Count of the number of albums returned. <i>Int32</i>
AlbumID	The unique album ID for the album being returned. <i>Int32</i>
AlbumTitle	The album title. <i>String(100)</i>
AlbumArtistID	Unique artist identifier for the artist that is the primary artist on the album. <i>Int32</i>
AlbumArtistName	Name of the primary artist for the album. <i>String(255)</i>
Year	The release year of the album. <i>Int16</i>
TrackCount	Total number of tracks associated with the album. <i>Int32</i>

## GetFor\_AlbumsForPeriod [PeriodID] [PageNumber] [PageSize] {UserData}

DESCRIPTION

Retrieves a list of albums for the specified period.

PARAMETERS

PeriodID	The unique period ID to retrieve a list of albums for. Note that the period ID can be either a distinct year (ex: 1980) or can be a decade reference (by adding a leading 1 to the id, ex: 11980 will return results from 1980 through 1989 inclusive). <i>Int16</i>
PageNumber	Specifies the page number to be retrieved. <i>Int32</i>
PageSize	Specifies the number of rows in each page. <i>Int32</i>

RESPONSE DEFINITION

GETFOR\_ALBUMSFORPERIOD♦[PeriodID]♦[PageNumber]♦[TotalPages]♦[TotalAlbums]♦[UserData]♦[\*][AlbumID]♦[AlbumTitle]♦[AlbumArtistID]♦[AlbumArtistName]♦[Year]♦[TrackCount]↷

PeriodID	The period ID of the albums being returned. <i>Int16</i>
PageNumber	The page number being returned. <i>Int32</i>
TotalPages	Indicates how many pages are available at the page size specified. <i>Int32</i>
TotalAlbums	Count of the number of albums returned. <i>Int32</i>
AlbumID	The unique album ID for the album being returned. <i>Int32</i>
AlbumTitle	The album title. <i>String(100)</i>
AlbumArtistID	Unique artist identifier for the artist that is the primary artist on the album. <i>Int32</i>
AlbumArtistName	Name of the primary artist for the album. <i>String(255)</i>
Year	The release year of the album. <i>Int16</i>
TrackCount	Total number of tracks associated with the album. <i>Int32</i>

## GetFor\_TracksForAlbum [AlbumID] [PageNumber] [PageSize] {UserData}

### DESCRIPTION

Retrieves a list of tracks for the specified album.

### PARAMETERS

AlbumID	The unique album ID to retrieve a list of tracks for. <i>Int32</i>
PageNumber	Specifies the page number to be retrieved. <i>Int32</i>
PageSize	Specifies the number of rows in each page. <i>Int32</i>

### RESPONSE DEFINITION

GETFOR\_TRACKSFORALBUM♦[AlbumID]♦[PageNumber]♦[TotalPages]♦[TotalTracks]♦[UserData]♦\*[TrackID]♦[TrackTitle]♦[TrackNumber]♦[TrackDurationInSeconds]♦[TrackDurationAsString]♦[OutputsPlayingThisTrack]↵

AlbumID	The album ID of the tracks being returned. <i>Int32</i>
PageNumber	The page number being returned. <i>Int32</i>
TotalPages	Indicates how many pages are available at the page size specified. <i>Int32</i>
TotalTracks	Count of the number of tracks returned. <i>Int32</i>
TrackID	The unique track ID for the track being returned. <i>Int32</i>
TrackTitle	The track title. <i>String(512)</i>
TrackNumber	The number of this track from the original recording. <i>uInt8</i>
TrackDurationInSeconds	Duration of the track. This is returned as an integer value in seconds. <i>Int32</i>
DurationAsString	Duration of the track. This is returned as a formatted string value (ex: 1:23:45). <i>String(10)</i>
OutputsPlayingThisTrack	This field can be used to determine if the returned track is currently playing in any of the playback zones. Note that this is a multi-valued field and will contain a delimited list of all zone numbers that are currently playing this track. <i>String(20)</i>

## GetFor\_TracksForPlaylist [PlaylistID] [PageNumber] [PageSize] {UserData}

### DESCRIPTION

Retrieves a list of tracks for the specified playlist.

### PARAMETERS

PlaylistID	The unique playlist ID to retrieve a list of tracks for. <i>Int32</i>
PageNumber	Specifies the page number to be retrieved. <i>Int32</i>
PageSize	Specifies the number of rows in each page. <i>Int32</i>

### RESPONSE DEFINITION

GETFOR\_TRACKSFORPLAYLIST♦[PlaylistID]♦[PageNumber]♦[TotalPages]♦[TotalTracks]♦[UserData]♦\*[TrackID]♦[TrackTitle]♦[SequenceNumber]♦[ArtistID]♦[ArtistName]♦[AlbumID]♦[AlbumTitle]♦[TrackDurationInSeconds]♦[TrackDurationAsString]♦ [OutputsPlayingThisTrack]↵

PlaylistID	The playlist ID of the tracks being returned. <i>Int32</i>
PageNumber	The page number being returned. <i>Int32</i>

TotalPages	Indicates how many pages are available at the page size specified. <i>Int32</i>
TotalTracks	Count of the number of tracks returned. <i>Int32</i>
TrackID	The unique track ID for the track being returned. <i>Int32</i>
TrackTitle	The track title. <i>String(100)</i>
SequenceNumber	This number is unique within a playlist and indicates this tracks order within the playlist. Since a playlist can contain a track more than once, this provides an alternative to track ID to uniquely identify a particular instance of a track within a playlist. <i>Int32</i>
ArtistID	The unique artist ID of the primary artist for the track being returned. <i>Int32</i>
ArtistName	The name of the primary artist for the track being returned. <i>Int32</i>
AlbumID	The unique album ID of the album that this track is associated with. <i>Int32</i>
AlbumTitle	The title of the album that the track is associated with. <i>String(100)</i>
TrackDurationInSeconds	Duration of the track. This is returned as an integer value in seconds. <i>Int32</i>
DurationAsString	Duration of the track. This is returned as a formatted string value (ex: 1:23:45). <i>String(10)</i>
OutputIsPlayingThisTrack	This field can be used to determine if the returned track is currently playing in any of the playback zones. Note that this is a multi-valued field and will contain a delimited list of all zone numbers that are currently playing this track. <i>String(20)</i>

## GetFiltered\_Albums [PageNumber] [PageSize] {UserData}

### DESCRIPTION

Retrieves a list of all albums in the library.

### PARAMETERS

PageNumber	Specifies the page number to be retrieved. <i>Int32</i>
PageSize	Specifies the number of rows in each page. <i>Int32</i>

### RESPONSE DEFINITION

GET\_ALBUMS♦[PageNumber]♦[TotalPages]♦[TotalAlbums]♦[UserData] ♦[AvailableAlpha]♦[AlbumID] ♦[AlbumTitle]♦[AlbumTitleSort]♦[AlbumArtistID]♦[AlbumArtistName]♦[Year]♦[TrackCount]↵

PageNumber	The page number being returned. <i>Int32</i>
TotalPages	Indicates how many pages are available at the page size specified. <i>Int32</i>
TotalAlbums	Count of the number of albums in the library. <i>Int32</i>
AvailableAlpha	A list of starting initial letters available in the result set. <i>String(30)</i>
AlbumID	Unique album identifier for the album being returned. <i>Int32</i>
AlbumTitle	Title of the album. <i>String(100)</i>
AlbumTitleSort	Sortable title of the album. <i>String(100)</i>
AlbumArtistID	Unique artist identifier for the artist that is the primary artist on the album.

	<i>Int32</i>
AlbumArtistName	Name of the primary artist for the album. <i>String(255)</i>
Year	Year the album was released. <i>Int16</i>
TrackCount	Number of tracks associated with the album. <i>Int32</i>

## GetDetails\_Album [AlbumID] {UserData}

### DESCRIPTION

Retrieves details for the specified album.

### PARAMETERS

AlbumID                    The unique album ID to retrieve detailed information for. *Int32*

### RESPONSE DEFINITION

GETDETAILS\_ALBUM♦[RequestedAlbumID]♦[UserData]\*[AlbumID]♦[AlbumTitle]♦[AlbumArtistID]♦[AlbumArtistName]♦[ReleasedYear]♦[Genre]♦[Styles]♦[Flags]↔

RequestedAlbumID	The album ID of the details being returned. <i>Int32</i>
AlbumID	The album ID for the track being returned. <i>Int32</i>
AlbumTitle	The album title. <i>String(100)</i>
AlbumArtistID	The unique artist ID of the primary artist for the album being returned. <i>Int32</i>
AlbumArtistName	The name of the primary artist for the album being returned. <i>String(255)</i>
ReleasedYear	The year that the album was released. <i>Int16</i>
Genre	The genre associated with this album. <i>String(100)</i>
Styles	This is a multi-valued field that returns a delimited list of all styles associated with this album. <i>String(512)</i>
Flags	This is a multi-valued field that returns a delimited list of all album flags associated with this album. Examples of album flags include:  LIVE, INSTRUMENTAL, EXPLICIT LYRICS <i>String(512)</i>

## GetDetails\_Artist [ArtistID] {UserData}

### DESCRIPTION

Retrieves details for the specified artist.

### PARAMETERS

ArtistID                    The unique artist ID to retrieve detailed information for. *Int32*

### RESPONSE DEFINITION

GETDETAILS\_ARTIST♦[RequestedArtistID]♦[UserData]\*[ArtistID]♦[ArtistName]♦[AlbumArtistID]♦[AlbumArtistName]♦[ReleasedYear]♦[Genre]♦[Styles]♦[Flags]↔

RequestedAlbumID	The album ID of the details being returned. <i>Int32</i>
AlbumID	The album ID for the track being returned. <i>Int32</i>
AlbumTitle	The album title. <i>String(100)</i>

AlbumArtistID	The unique artist ID of the primary artist for the album being returned. <i>Int32</i>
AlbumArtistName	The name of the primary artist for the album being returned. <i>String(255)</i>
ReleasedYear	The year that the album was released. <i>Int16</i>
Genre	The genre associated with this album. <i>String(100)</i>
Styles	This is a multi-valued field that returns a delimited list of all styles associated with this album. <i>String(512)</i>
Flags	This is a multi-valued field that returns a delimited list of all album flags associated with this album. Examples of album flags include: LIVE, INSTRUMENTAL, EXPLICIT LYRICS <i>String(512)</i>

## System Feedback

Osonix systems are capable of sending a variety of feedback data including playback status, currently playing track, etc. By default, sending of feedback is turned off and in order to enable data feedback, a command must be sent specifying the type of feedback desired. Feedback data is divided by purpose into message types and one or more feedback message types may be turned on simultaneously. Most types of feedback messages are sent only when necessary (when the data changes). For example, the Playstate feedback message will be sent whenever the play state of any zone changes. In addition, when a feedback message type is enabled, a feedback message of each type that is currently enabled will be sent immediately to report the state at that time. In order to *poll* the unit for a current status, you can simply re-initialize the feedback message type again and this initial status message will be sent immediately. This is convenient since the message will be formatted the same whether it is a response to a *poll* or is an unsolicited response by the system as a result of a data change.

### Feedback TrackChange [On/Off]

#### DESCRIPTION

Enabling this type of feedback will cause feedback messages to be sent when the current track in the now playing queue changes. Two types of messages will be sent by this type of feedback. One will return the CURRENTTRACK information and the other will return the NEXTTRACK information. If there is not a next track in the now playing queue when this feedback occurs, the NEXTTRACK message will be omitted. In addition, one of each type of message will be sent for each playback zone on the system. (See response types “CURRENTTRACK” and “NEXTTRACK” below).

#### PARAMETERS

On/Off                      Specifies whether to turn the specified type of feedback on or off. Valid values include: ON, OFF.

#### RESPONSE DEFINITIONS

FEEDBACK♦CURRENTTRACK\*[ZoneNumber]♦[TrackID]♦[TrackTitle]♦[AlbumTitle]♦[AlbumArtistName]♦[SequenceNumber] ↻

FEEDBACK♦NEXTTRACK\*[ZoneNumber]♦[TrackID]♦[TrackTitle]♦[AlbumTitle]♦[AlbumArtistName]♦[SequenceNumber] ↻

ZoneNumber	The number of the zone that this feedback applies to. <i>uInt8</i>
TrackID	The track ID of the track being returned. <i>Int32</i>
TrackTitle	The track title. <i>String(512)</i>
AlbumTitle	The title of the album that the track belongs to. <i>String(100)</i>
AlbumArtistName	The name of the primary artist of the album that the track belongs to. <i>String(255)</i>
SequenceNumber	This number is unique within a playlist and indicates this tracks order within the playlist. Since a playlist can contain a track more than once, this provides an alternative to track ID to uniquely identify a particular instance of a track within a playlist. <i>Int32</i>

### Feedback TrackPosition [On/Off]

#### DESCRIPTION

Enabling this type of feedback will cause a feedback message to be sent any time the track position changes. During playback, this will result in one message being sent every second.

#### PARAMETERS

On/Off Specifies whether to turn the specified type of feedback on or off. Valid values include: ON, OFF.

#### RESPONSE DEFINITIONS

FEEDBACK♦POSITION\*[ZoneNumber]♦[TrackDuration]♦[TrackPosition]♦[TrackTimeRemaining]♦[TrackDurationString]♦[TrackPositionString]♦[TrackTimeRemainingString] ↗

ZoneNumber	The number of the zone that this feedback applies to. <i>uInt8</i>
TrackDuration	Duration of the current track returned as a integer value of seconds. <i>Int32</i>
TrackPosition	The current playback position of the track returned as an integer value of seconds. <i>Int32</i>
TrackTimeRemaining	The number of seconds remaining in the currently playing track. <i>String(10)</i>
TrackPositionString	The current playback position of the track returned as a formatted string value (ex: 1:23:45). <i>String(10)</i>
TrackTimeRemainingString	The amount of time remaining in the current track returned as a formatted string value (ex: 1:23:45). <i>String(10)</i>

### Feedback RepeatChange [On/Off]

#### DESCRIPTION

Enabling this type of feedback will cause a single feedback message to be sent when the playback Repeat state changes (on / off).

#### PARAMETERS

On/Off Specifies whether to turn the specified type of feedback on or off. Valid values include: ON, OFF.

#### RESPONSE DEFINITIONS

FEEDBACK♦REPEAT\*[ZoneNumber]♦[RepeatState] ↗

ZoneNumber	The number of the zone that this feedback applies to. <i>uInt8</i>
RepeatState	The current state of the repeat setting. Valid values include: ON, OFF. <i>String(3)</i>

### Feedback PlaylistChange [On/Off]

#### DESCRIPTION

Enabling this type of feedback will cause a feedback message to be sent any time the Now Playing queue changes for a particular playback zone.

#### PARAMETERS

On/Off Specifies whether to turn the specified type of feedback on or off. Valid values include: ON, OFF. *String(3)*

#### RESPONSE DEFINITIONS

FEEDBACK♦NOWPLAYINGCHANGED\*[ZoneNumber] ↗

ZoneNumber	The number of the zone that this feedback applies to. <i>uInt8</i>
------------	--

## Feedback PlaystateChange [On/Off]

### DESCRIPTION

Enabling this type of feedback will cause a feedback message to be sent any time the Now Playing queue changes for a particular playback zone.

### PARAMETERS

On/Off Specifies whether to turn the specified type of feedback on or off. Valid values include: ON, OFF. *String(3)*

### RESPONSE DEFINITIONS

FEEDBACK♦PLAYSTATE♦[ZoneNumber]♦[PlayState] ↻

ZoneNumber The number of the zone that this feedback applies to. *uInt8*

PlayState Current playback status for the specified zone. Valid values include: Playing, Stopped, Paused. *String(20)*

## Feedback Volume [On/Off]

### DESCRIPTION

Enabling this type of feedback will cause a feedback message to be sent any time the Now Playing queue changes for a particular playback zone.

### PARAMETERS

On/Off Specifies whether to turn the specified type of feedback on or off. Valid values include: ON, OFF. *String(3)*

### RESPONSE DEFINITIONS

FEEDBACK♦VOLUME♦[ZoneNumber]♦[CurrentVolume] ↻

ZoneNumber The number of the zone that this feedback applies to. *uInt8*

CurrentVolume Current volume of the specified zone as a percentage (0-100). *uInt8*

## Feedback All [On/Off]

### DESCRIPTION

Sets all feedback types on or off using a single command.

### PARAMETERS

On/Off Specifies whether to turn the specified type of feedback on or off. Valid values include: ON, OFF. *String(3)*

## Feedback GetValue

### DESCRIPTION

Returns a message that contains the current feedback settings.

### PARAMETERS

None

### RESPONSE DEFINITIONS

FEEDBACK♦STATUS♦

TRACKPOSITION♦[On/Off] ↻

TRACKCHANGE♦[On/Off] ↻

REPEATCHANGE♦[On/Off] ↻

PLAYLISTCHANGE♦[On/Off]↻  
PLAYSTATECHANGE♦[On/Off]↻

## Cover Art Image Support

Access to album cover art images is provided via HTTP using the built-in web server on the system. By using the available query string parameters to build a full cover art image URL, a jpeg bitmap can be retrieved from the server for any album in the system.

### URL

`http://<server address>/WebServices/CoverArtImage.aspx`

### QUERYSTRING PARAMETERS

AlbumID	If specified, the cover art returned will correspond to the Album with the ID specified. This parameter should not be used in conjunction with TrackID. Only one of the two parameters should be used. <i>Int32</i>
TrackID	If specified, the cover art returned will correspond to the Album that contains the track with the ID specified. This parameter should not be used in conjunction with AlbumID. Only one of the two parameters should be used. <i>Int32</i>
Size	The size (in pixels) of the cover art image to be returned (maximum allowable size is 1000 pixels). If Reflect is turned on, the resulting image will have a width of this size and a height of twice this size. <i>Int16</i>
Reflect	If this parameter is specified with a value of ON, then the returned image will include a vertical reflection in the image. <i>String(3)</i>

### EXAMPLES

`http://192.168.0.1/WebServices/CoverArtImage.aspx?AlbumID=12345&Size=100&Reflect=ON`

`http://192.168.0.1/WebServices/CoverArtImage.aspx?TrackID=4321&Size=250`

## Deprecated commands

### Queue [Zone #] [Location]

This command has been replaced by the `QueueItem` command starting in version 2.4.8.

#### DESCRIPTION

Adds the currently selected item to the Now Playing playlist for the specified [Zone]. The location in the current playlist that the item will be inserted is determined by the value of the *Location* parameter. The *Location* parameter is optional and if not specified, the default is *End*. Valid values for *Location* are listed below.

#### PARAMETERS

**Zone #** Specifies which zone should be acted upon. Must be 0 or a valid zone number.

**Action** Specifies what action should be taken on the specified zone's now playing playlist. Action must be one of the following values:

*ClearAll* – Clears all tracks from the playlist

*ClearPlayed* – Clears all tracks that have completed playing from the playlist

Command	Description / Usage
GET PLAYLISTS REFRESH GET PLAYLISTS FIRST GET PLAYLISTS NEXT GET PLAYLISTS PREVIOUS GET PLAYLISTS LAST GET PLAYLISTS PAGESIZE [PageSize]	<p>Retrieves a list of playlists stored on the server. The PAGESIZE command can be used to set the number of playlists returned per page (specified by the <i>PageSize</i> parameter). This would typically be called upon initialization of the client application. Omitting the paging command parameter (ex: GET PLAYLISTS) will retrieve the current page index and can be useful for refreshing the contents of that page.</p> <p><i>Response format:</i></p> <p>PLAYLISTS♦[PageNumber]♦[TotalPages] ♦[TotalRows]★            [PlaylistID]            ♦[PlaylistName]♦[TrackCount]♦[PlaylistLength]♦[LengthAsString]↻</p> <p><i>Field Descriptions:</i></p> <p><i>PageNumber</i> – The number of the page being returned (numbered from 1 to the total number of pages)  <i>TotalPages</i> – The total number of pages available for this response  <i>TotalRows</i> – The total number of rows available for this response  <i>PlaylistID</i> – A system-assigned ID number uniquely assigned for each playlist  <i>PlaylistName</i> – The user-specified description for the playlist  <i>TrackCount</i> – Total number of tracks in the playlist  <i>PlaylistLength</i> – Total duration of the playlist (in seconds)  <i>LengthAsString</i> – Total duration of the playlist (formatted as hh:mm:ss)</p>
GET NOWPLAYING REFRESH [ZoneNumber] GET NOWPLAYING FIRST	<p>This command is used to retrieve the current Now Playing list of tracks. This command includes an additional parameter of Zone Number and this is unique to the NOWPLAYING commands. As</p>

<p>[ZoneNumber] GET NOWPLAYING NEXT [ZoneNumber] GET NOWPLAYING PREVIOUS [ZoneNumber] GET NOWPLAYING LAST [ZoneNumber] GET NOWPLAYING CURRENT [ZoneNumber] GET NOWPLAYING PAGESIZE [PageSize]</p>	<p>with PLAYLISTS, the basic command “GET NOWPLAYING” will return the first page of tracks from the now playing list. The PAGESIZE command can be used to set the number of playlists returned per page (specified by the <i>PageSize</i> parameter). This would typically be called upon initialization of the client application. The <i>ZoneNumber</i> parameter must be a value of either 1 or 2. Omitting the paging command parameter (ex: GET NOWPLAYING 1) will retrieve the current page index and can be useful for refreshing the contents of that page.</p> <p><i>Response format:</i></p> <p>NOWPLAYING♦[PageNumber] ♦[TotalPages] ♦[ZoneNumber]★ [TrackID] ♦[TrackTitle]♦[AlbumTitle]♦[AlbumArtistName]♦[Sequence] ♦[IsCurrent]↵</p> <p><i>Field Descriptions:</i></p> <p><i>PageNumber</i> – The number of the page being returned (numbered from 1 to the total number of pages)  <i>TotalPages</i> – The total number of pages available for this response  <i>ZoneNumber</i> - The number of the zone that this result applies to  <i>TrackID</i> – A system-assigned ID number uniquely assigned to each track in the system  <i>TrackTitle</i> – The name of the song / title of the track  <i>AlbumTitle</i> – The title of the album on which the track appears  <i>AlbumArtistName</i> – The name of the artist from the album where the track appears.  <i>Sequence</i> – An incremented number assigned to each track in the now playing list. This indicates in which position the track appears in the list.  <i>IsCurrent</i> – If this field is “1”, then this track is the currently playing track (or if stopped, it is the current index that would play when play is started). If it is “0” (zero), this is not the current track.</p>
<p>GET ZONENAME [ZoneNumber]</p>	<p>Used to retrieve the user-specified name assigned to each of the zones in the system. The <i>ZoneNumber</i> parameter must be a value of 1 or 2.</p> <p><i>Response Format:</i></p> <p>ZONENAME★[ZoneNumber]♦[ZoneName]↵</p> <p><i>Field Descriptions:</i></p> <p><i>ZoneNumber</i> – The number of the zone  <i>ZoneName</i> – The user-specified name of the zone</p>