# AudioTron Version 3.0 Application Programming Interface

Version: 0.21
Author: HMB
Date: Wednesday, January 23, 2002

This document and its contents are the intellectual property of Voyetra Turtle Beach Inc., (VTB). It may not be copied, reproduced, modified, uploaded, downloaded, transmitted, or distributed in any way.

The VTB Intellectual Property is protected by copyright, trade secret, and other intellectual property laws and by international treaties, all of which provide rights and obligations that are supplemental to the rights and obligations set forth in this Agreement.

Except as specifically provided herein, there are no other warranties expressed or implied, including, but not limited to, any implied warranties of merchantability or fitness for a particular purpose. In no event shall VTB be liable for indirect, incidental, consequential, or other damages including personal injury, lost profits and damages to property arising out of any breach of this warranty or otherwise. Our policy is one of ongoing product updates, improvement and revision. We may revise and discontinue products at any time.

VTB reserves the right to change these terms and conditions without prior written notice and at the sole discretion of VTB.

Turtle Beach
5 Odell Plaza, Yonkers, NY 10701
Tel: (914) 966-0600
Fax: (914) 966-1102
http://www.TurtleBeach.com

***Revision Log:***

| **Version** | **Date** | **Notes** | **Author** |
|---|---|---|---|
| 0.1 | 01/23/2002 | Initial Creation | HMB |
| 0.2 | 2/8/2002 | Added TOC files and updated | HMB |

**Table OF Contents**

# 1 Introduction

In order to make the AudioTron a more useful device in vertical markets as well as give multiple AudioTrons more capabilities together than separately, an API has been developed to allow remote control of the AudioTron.

# 2 Overview

The AudioTron API is based on HTTP. Every request to the AudioTron is sent as an HTTP GET. The replies are returned as the data from the get.

## 2.1 Requests

The API call is defined as a get of a particular page and the parameters are passed in as query string entries. Thus, the URLs for all requests to the AudioTron is formatted as follows:

http://<ipaddr>/<pagename>.asp?<input1>=<xxxx>&<input2>= <yyyy>

For example the command to start playback might be:

http://192.168.0.1/apicmd.asp?cmd=play

The basic pages that constitute the API are:

- Apicmd.asp - Processes commands to start operations
- Apigetinfo.asp - processes requests for information from the internal database
- Apiqfile.asp - Processes requests for adding items to the play queue
- Apiaddfile.asp - Allows the user to tell the AudioTron about files that it might not know about.

Each of these will be described in more detail below.

## 2.2 Responses

The AudioTron will take this input, perform some function and return any required information as the result of this http GET. NOTE: what is returned is NOT HTML. It is strictly text formatted particular to the AudioTron.

Other than apigetinfo, all responses are of the type:

Error =<#> - <explanatory text>

The response to all successful operations is:
Error 0 - Success\n

# 3 Requests

## 3.1 Apicmd.asp

The apicmd.asp processes commands to start operations.

There are 2 forms to the apicmd.asp URL. The first form of the URL is: http://<ipaddr>/apicmd.asp?cmd=<cmdname>. For example, the command to stop playback is: http://<ipaddr>/apicmd.asp?cmd=stop. The possible commands for this form of apicmd.asp are:

- Play - Equivalent to pressing the play button except for play queue construction.
- Stop - Equivalent to pressing the stop button
- Next - Equivalent to pressing the Next button
- Prev - Equivalent to pressing the Prev button
- Clear - Clears the Play Queue

The second form of the URL allows for particular state items within the AudioTron to be set. The format for the URL in this case is: http://<ipaddr>/apicmd.asp?cmd=<cmdname>&arg=<valuel>. For the following commands the acceptable values for arg are:

- -1 always sets the state to off.
- 1 always sets the state to on.
- 0 will toggle the state.

An example of this format would thus be: http://<ipaddr>/apicmd.asp?cmd=mute&arg=-1 . This URL will turn off mute if it is active, if mute is not active then no action will be taken. NOTE: if no state value is given then the state will be toggle.

- Pause - Equivalent to pressing the pause button
- Random - Equivalent to pressing the Random button
- Repeat - Equivalent to pressing the Repeat button
- Mute - Equivalent to pressing the Mute Button

For the command goto, the value of arg is an integer between 0 and the number of items in the play queue. Thus the command to start playing song 18 in the play queue would be:
: http://<ipaddr>/apicmd.asp?cmd=goto&arg=18.

## 3.2 Apigetinfo.asp

The apigetinfo.asp page returns info about a number of different lists within the AudioTron.

The form of the URL for apigetinfo.asp is:

http://<ipaddr>/apigetinfo.asp?type=<typename>&count=<itemcount>&this=<startingitem>

This will return a number of records from a particular list starting at the particular item.

For example: http://<ipaddr>/apigetinfo.asp?type=artist&count=4&this=The%20BandName. This request will return the next 4 entries in the artist list starting with entry for "The BandName".

The "count" and "this" parameters are optional.

If the count parameter is not supplied, the AudioTron will return the rest of the list.
If the "this" parameter is not supplied then the AudioTron will start at the beginning of the list.

The available types are:

- **Title** - This type gets the list of available titles. This is the same list as available under Title on the front panel.
- **Artist** - This type gets the list of available artists. This is the same list as available under Artist on the front panel.
- **Genre** - This type gets the list of available genres. This is the same list as available under Genre on the front panel.
- **Album** - This type gets the list of available albums. This is the same list as available under Album on the front panel (without the artist in album name option checked)
- **Web** - This type gets the list of available internet radio genres. This is the same list as available under Net on the front panel.
- **List** - This type gets the list of available playlists. This is the same list as available under Lists on the front panel.
- **File** - This type is a list of all of the playable entities including files, Playlists, and Internet radio stations. (Subject to filtering requirements discussed later).
- **Playq** - This entry returns the contents of the Play queue.

### 3.2.1 File Filtering

In addition to getting song information about the entire list, the api will allow for filtering of the returned songs. This filtering is only use for a type of file. There are 4 filter criteria available:

- **ftitle** - The AudioTron will return only those songs that completely matches this title string
- **fartist** - The AudioTron will return only those songs that completely matches this artist string
- **falbum** - The AudioTron will return only those songs that completely matches this album string
- **fgenre** - The AudioTron will return only those songs that completely matches this artist string

For example: http://<ipaddr>/apigetinfo.asp?type=file&count=4&fartist=The%20BandName&. This request will return the first 4 entries in the song list with artist matching "The BandName". The user can use one filter criteria of each type in a query. The results of all of the filters are AND'ed together.

### 3.2.2 Input and Output Formats

#### 3.2.2.1 Title, Artist, Genre, Album, Web, and List

The Title, Artist, Genre, Album, Web, and List all share a format.

The "this" parameter is a URL encoded version of the string to be searched for (e.g. "The%20BandName").

The returned data is:

- A header line naming the list
- A line indicating the starting item
- Each entry in the list seperated by \n.
- A Footer indicating the end of the data

Note: This is raw text not HTML formatted.

For example: http://<ipaddr>/apigetinfo.asp?type=artist&count=4&this=The%20BandName

```
[Artist List]
Start=The BandName
The BandName
The BandName Next
The Choice Band
Seths Band
ZZZ Band
[End List]
```

#### 3.2.2.2 File List

The information in the File list is the heart of the AudioTron. This list is a list of all things that the AudioTron can play directly. This list includes all of the files that the AudioTron has discovered locally and all of the Internet radio stations that the AudioTron has gotten from Turtle Radio.

Each of the items in this list contains an ID that is either the URL of the station of a UNC for the file. These are used for the "this" parameter. The list is sorted alphabetically by this ID.

The format for the returned data in this case:

- A header line
- A line indicating the ID of the starting item
- Each Song record surrounded by the lines [Song] [End Song]
- Within each song record are the Following fields:
  - Title - Unique title from the title list
  - Album (optional) - If there is an associated album tag it is listed here
  - Artist  (optional) - If there is an associated Artist tag it is listed here.
  - Genre (optional) - Genre from the tags in the entry or Turtle Radio
  - Track - Album track order if tagged, -1 if not tagged
  - Time - the length of the file in seconds, from the file tags only.
  - ID - The URL or UNC of this item.
- A footer Line indicating the end of the data.

For Example
http://<ipaddr>/apigetinfo.asp?type=file&count=4&this=\\HMB-PROD\AUDIO\audio\20%20kbits.mp3

Might return the following data:

```
[Song List]
Start=\\HMB-PROD\AUDIO\audio\20 kbits.mp3
[Song]
Title=The 20Kbit test file - (1)
Album=MP3 Bite Rate Tests
Artist=AT Test
Genre=Test
Track=-1
Time=0
ID=\\HMB-PROD\AUDIO\audio\20 kbits.mp3
[End Song]
```

```
[Song]
Title=The 20Kbit test file
Album=MP3 Sample Rate Tests
Artist=AT Test
Track=-1
Time=0
ID=\\HMB-PROD\AUDIO\audio\22.mp3
[End Song]
[Song]
Title=Back to Name
Artist=An Artist
Genre=Piano
Track=5
Time=302
ID=\\HMB-PROD\AUDIO\audio\Back to Name.wav
[End Song]
[Song]
Title=Pinch
Artist=Bare Back Band
Genre=Rock
Track=-1
Time=0
ID=\\HMB-PROD\AUDIO\audio\BARE BACK BAND - PINCH.MP3
[End Song]
[End List]
```

### 3.2.2.3 PlayQ

The formats for the File list the Playq list share a lot in common. The major difference is how the list is accessed. Since it is possible to enqueue a single item to the Play Queue multiple times, the ID is not sufficient. Therefore, the play queue is accessed by index. A URL for a play queue query would be: http://<ipaddr>/apigetinfo.asp?type=playq&count=4&this=2. This would result in the return of the items in the play queue with indices 2-6.

The differences between the File returned data and the Playq are as follows:

- The type of the Starting item is a number.
- The Song index in the Song record indices
- Each song entry as an additional field indicating its position in the random playback list.

Thus a possible reply to the URL : http://<ipaddr>/apigetinfo.asp?type=playq&count=4&this=2 would be:

```
[Song List]
Start=2
[Song 2]
RandIndex=5
Title=Dig that test file- (1)
Album=MP3 Bite Rate Tests
Artist=AT Test
Genre=Test
Track=-1
Time=0
ID=\\HMB-PROD\AUDIO\audio\20 kbits.mp3
[End Song]
[Song 3]
```

```
RandIndex=2
Title=Dig that test file
Album=MP3 Sample Rate Tests
Artist=AT Test
Track=-1
Time=0
ID=\\HMB-PROD\AUDIO\audio\22.mp3
[End Song]
[Song 4]
RandIndex=7
Title=Back to Title
Artist=Phil
Genre=Piano
Track=5
Time=302
ID=\\HMB-PROD\AUDIO\audio\Back to Title.wav
[End Song]
[Song 5]
RandIndex=12
Title=Pinch
Artist=BareBack Band
Genre=Rock
Track=-1
Time=0
ID=\\HMB-PROD\AUDIO\audio\BAREBACK BAND - PINCH.MP3
[End Song]
[End List]
```

## *3.3 Apiqfile.asp*

This functions allows for items to be added to the end of the Play Queue.The format for this request is as follows:

http://<ipaddr./apiqfile.asp?type=<itemtype>&file=<string orID>

The apiqfile allows the following types to be added to the Play queue.

- **Title**
- **Artist**
- **Genre**
- **Album**
- **Web**
- **List**
- **File**
- **Fav**

The "file" parameter is the ID or string associated with type (except for fav). For Example, the following are proper requests:

http://<ipaddr>/apiqfile.asp?type=file&file=\\HMB-PROD\AUDIO\audio\20%20kbits.mp3
 - This would enqueue the file \\HMB-PROD\AUDIO\audio\20 kbits.mp3 for playback

http://<ipaddr>/apiqfile.asp?type=artist&file=The%20BandName
- This would enqueue all of the songs by "The BandName" for playback.

For the type fav, the file=<number of the favorite>.

http://<ipaddr>/apiqfile.asp?type=fav&file=10
- This would enqueue the item saved as favorite number 10.

### 3.4 Apiaddfile.asp

This function allows for a file to added to the AudioTron database without re-searching. The format to this request is:

http://<ipaddr>/apiaddfile.asp?type=file&file=\\HMB-PROD\AUDIO\audio\20%20kbits.mp3

This will cause the AudioTron to perform the following actions:

- Verify that this file is a viable file type for the AudioTron.
- Verify the existence of the file
- Read the file and catalog the various tags.
- Add the file to the database.

Currently, there is only 1 file type: file which takes an argument of a UNC filename.

### 3.5 Apigetstatus.asp

Apigetstatus provides status information about the current state of the AudioTron. It provides the same information as the index.asp page. It has no parameters. Thus the URL would be:

http://<ipaddr>/apigetstatus.asp

Below is an example of the returned data.

```
[Status]
STOP_LED=0
PLAY_LED=1
PAUSE_LED=0
RANDOM_LED=0
GROUP_LED=0
MUTE_LED=0
REPEAT_LED=0
State=	Playing
QueueLen=2
CurrIndex=0
CurrTime=45
TotalTime=226
Title=A Celtic Song
Artist=Celtic band
Album=The Celtic Band Album
Source=UNC
SourceID=\\HMB-PROD\AUDIO\audio\The_Celtic_Band_TheCelticSong.mp3
NextIndex=2
```

```
NextTitle=Best I Had Today
NextArtist=Horizontal Azimuth
NextAlbum=HA album
NextGenre=Rock
NextSource=UNC
NextSourceID=\\HMB-PROD\AUDIO\audio\Horizontal Azimuth - BEST I HAD TODAY.MP3
[End Status]
```

The fields are defined as follows:

- [Status] - Header
- STOP_LED, PLAY_LED,PAUSE_LED,RANDOM_LED,GROUP_LED, MUTE_LED,REPEAT_LED. These fields indicate the state of the front panel LEDS (and thus the associated functions). 1 indicates the LED is lit, 0 indicates it is not lit.
- State - indicates the state of the current song playback, the possible values are:
  - Inactive - Playback is not active.
  - Opening - Attempting to open the destination
  - Closing - Closing the destination file
  - Closed - The file has been closed, waiting for Audio playback to complete
  - Buffering %d - Percent of prebuffering that is complete
  - Buffered - Prebuffering is complete, waiting for audio playback to start
  - Playing - Currently playing an item
  - Paused - Playback is paused
  - Stopped - Playback is stopped
  - Unknown - Should never happen
  - Stopping - Stopping playback waiting for closure of decoder and file.
- QueueLen - The current length of the Play queue
- CurrIndex - The index in the play queue of the currently playing song.
- CurrTime - The number of seconds that have elapsed since play back started
- TotalTime - The total duration of this selection in seconds ,if known.
- Title, Artist, Album, Genre - values of the tags from this item
- Source - Type of thing being played either UNC (local file) or URL.
- SourceID - The UNC or URL of the currently playing song.
- NextIndex - If there is an on deck song, this indicates the play queue index of this song.
- NextTitle,NextArtist,NextAlbum,NextGenre,NextSource,NextSourceID - same information as their counterparts in the currently playing song.
- [End Status] - footer

# 4 Table of Contents Files

In addition to the Control API described above, there is also a new feature for speeding up the file database creation. This is being done through support for Table Of Contents (TOC) files.

A TOC file is that contains all of the information that the AudioTron requires to build it's internal playable items database. In particular, it contains pathnames, tags and other information normally pulled from the files themselves. AudioTron simply reads this file line by line and creates its own internal database.

NOTE: The AudioTron does no checking of the validity of this data. It does not check to see if the files described actually exist, that the tag information inside the files is consistent with the TOC file, or that there are files that are not represented in the TOC. It is the responsibility of the program that creates the TOC to keep it up to date.

This file must reside at the top of a particular share and be named atrontc.vtc. If the AudioTron find one of these files, it processes the contents and uses it to create it's internal song and playlist database. It then will ignore the rest of that share. When the AudioTron does a search for new music, it will compare the last update time of the TOC file to the last time that the AudioTron accessed it. If it is unchanged, the AudioTron will not do anything with this share.

Sidenote: these files can be used to prevent the AudioTron from searching a share at all. In this case, an empty atrontc.vtc will prevent the AudioTron from trying to search for music.

## *4.1 TOC file format*

The format is a series of Song records each containing information about that song. The Song structure is composed as follows:

- a series of tags, one per line.
- Each tag is 4 characters long  and may or may not be followed by an equals sign.
- After the equals sign is the value of that tag. NOTE: Everything between the "=" and the "\n" are significant.
- All songs start with a SONG tag line.
- All Songs end with a "END " (Note the Space) line.

Below is a example Song Entry.

```
SONG
FILE=Track 04.mp3
DIR =\AUDIO
TCON=Genre
TLEN=225
TRCK=4
TALB=title
TPE1=Bare Back Band
TIT2=Track 04
END
```

### 4.1.1 TOC Tags

The valid tags are currently: (Yup, they are borrowed from MP3 TAGS).

| | |
|---|---|
| "SONG" | - Indicates the beginning of a  song record |
| "TIT2" | - Indicates the Title of a song |
| "TCON" | - The Genre of the Song (text only) |
| "TALB" | - The album name associated with this song |
| "TPE1" | - The Artist associated with this song. |
| "TRCK" | - Track Number |
| "TLEN" | - The length of the song in seconds. |
| "END " | - Indicates the end of the song record (NOTE THE SPACE) |
| "FILE" | - Indicates the filename for this file.<br>Does not contain any directory information. |
| "DIR " | - Directory information associated with this file.<br>Contains the path from the top of the share to the file in question. NOTE THE SPACE. |

NOTE ON THE CONSTRUCTION OF THE FULL PATH NAME. - The AudioTron will create a full path name for file as follows:

<share name><dir value>\<file value>. Note the "\" between the dir and file name that is not included in the tag values.

The share name is determine by the share location mechanism. The file and dir strings come from the tags of the TOC files.