Audiotron Version 3.0 Application Programming Interface

Version: 0.44
Author: HMB
Date: 8/5/2002

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
0.21	2/21/2002	Incorporated comments from Beta	HMB
0.3	3/11/2002	Incorporated Changes for Beta 2	HMB
0.3.1	03/27/2002	Incorporated Reviewer Comments	HMB
0.40	04/22/2002	Changes for Beta 3	HMB
0.4.1	05/1/2002	Added terminate Message command	HMB
0.42	05/24/2002	Added timeout to apimsg.asp	HMB
0.43	06/06/2002	Added Filtering to Other List Types	HMB
0.44	08/05/2002	Added Prefix parameter to apigetinfo	HMB

Table OF Contents

1 Changes

1.1 Version 3.0.2

- Added prefix processing to apigetinfo.asp

1.2 Version 2.5.7

- Added filtering to getinfo for types album, artist, and genre.

1.3 Version 2.5.5

- Added timeout parameter to the apimsg.asp command
- Added terminate command for api message.

1.4 Version 2.1.2x

- Changed apimsg.asp message to be persistent until a front panel or remote control event
- Added Power on and power off commands
- Added filtering to apiqfile requests
- Changed CurrTime in apigetstatus to CurrPlayTime to avoid confusion with other CurrTime
- Added CurrVolume to apigetstatus to reflect the current volume setting.
- Cleaned up some documentation issues.

1.5 Version 2.1.18

- Added Curr times to Status page and
- Changed a number of tags to make them more consistent.
- Changed Favorites Listing
- Added PlayQUpdateTime to status page and Global page.
- Added LastChangeTime for each of the lists to global page.
- Added UPDT tag to TOC format.

1.6 Version 2.1.15

- Made API next requests not wait for timeout.
- Added API to print message on front panel
- Added changes to Tweak TOC performance
- Made apigoto before playback work.
- Added code to upper case shares for api requests
- Fixed api problem with clear followed by q follwed by play.
- Added volume and position to the API
- Added after filtering to the API.
- Fixed genre filtering in API
- Fixed Item count to be count of items returned not looked at.
- Added favorites to API get info.
- Added global info to API.
- Fixed Playlists in get info.
- Added create times to lists

- Fixed mulitple error messages for api calls
- Added ability to get a TOC file from the API.
- Changed returned MIME type for api requests to "text/plain"
- Changed API so that clear queue does not stop playback.
- Changed Documentation to remove starting backslash from DIR in TOC and put one at the end

2 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.

3 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. The MIME type is "text/plain"

3.1 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.

NOTE: all variable names (e.g. cmd, falbum, etc) need to be all lower case. The variable values are case insensitive.

3.2 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

4 Requests

4.1 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
- poweron - Equivalent of pressing the standby button when the Audiotron is in standby. Has no effect if the unit is not in standby.
- poweroff - Equivalent of pressing the standby button during normal operation. Has no effect if unit is in standby.
- msgoff - Terminates any message set using the API and returns to the current screen.

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.

Volume

To set the Audiotron Volume, use the URL form http://<ipaddr>/apicmd.asp?cmd=volume&arg=<xxx>. In this case the arg, the arg is the requested volume. The range of acceptable values for volume are 0 to -96, indicating the decibel setting for the volume control.

Position

To set the Audiotron Position use the URL form: http://<ipaddr>/apicmd.asp?cmd=position&arg=<xxx>. The argument indicates the requested position within the song. The available values for this argument are 0-100, indicating the relative position within the file. 0 is the beginning and 100 is the end.

4.2 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".

In addition to the "this" parameter to indicate where to start a returned list, there is also another parameter called "prefix". This parameter will start a list starting with the item that is either equal or greater than the indicated prefix string. The comparison is case-insensitive. This parameter is available for Title, Artist, Genre, Album, Web, and List.

For example: http://<ipaddr>/apigetinfo.asp?type=artist&count=4&prefix=CT. This request will return the next 4 entries in the artist list starting with entry whose first two letters are greater than "CT".

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).
- **Fav** - Lists the values for each of the favorites.
- **Playq** - This entry returns the contents of the Play queue.
- **Global** - This type will return a set of global information about the Audiotron.

4.2.1 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
- **fafter** - The Audiotron will only return those items that were added to the Audiotron database after the indicated time. This filter is intended to be used to filter the getinfo results to obtain only items newly added to the database.

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.

As of version 2.5.7, filterin has now been extended to types Album, Artist, and Genre. For example, the following query is now valid.
http://<ipaddr>/apigetinfo.asp?type=album&count=4&fartist=The%20BandName

This will return the first four album names by the artist "The Bandname".

4.2.2 Input and Output Formats

4.2.2.1 Note on Atron Relative times

Many replies have an atron relative time as part of their data. This time is the number of seconds since the Audiotron was rebooted. These times can serve several different purposes including:

- Caching data - Since most of the data structures have a last changed time associated with them, the user of the API can use these to determine when information within that structure has changed.
- Fafter filtering of lists - These time stamps can be used to filter requests for list info to only get information that has changed since the last time a request was made.
- Determining Audiotron status changes - Many of the replies have CurrTime as part of the reply. This indicates the number of seconds that the Audiotron has been up. As such the user can calculate an expected CurrTime value (within a certain delta). For example, if the last request for information occurred 100 seconds ago and the CurrTime value at that point was 200 seconds, the expected CurrTime value of this reply would be 300 (100+200). If there is a significant discrepancy between the expected and returned value, then the Audiotron has probably been rebooted in the interim.

4.2.2.2 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
- A line indicating the Audiotron relative time of this response.
- A line indicating the last time anything was added to this list
- A line indicating the total number of items in this list
- Each entry in the list.
  - Each entry consists of 2 lines:
  - Item=xxxxx\n - This line is the value of the of the string for that entry

- 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
CurrTime=800
LastChangeTime=300
ListCount=100
Item=The BandName
Item=The BandName Next
Item=The Choice Band
Item=Seths Band
Item=ZZ Band
[End Artist List]
```

4.2.2.3 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
- A line indicating the Audiotron relative time that the file list was requested.
- 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.
  - CreateTime - The Audiotron relative time this entry was created.
  - 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
CurrTime=1200
[Song]
Title=The 20Kbit test file - (1)
Album=MP3 Bite Rate Tests
Artist=AT Test
Genre=Test
Track=-1
Time=0
CreateTime=800
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
CreateTime=800
ID=\\HMB-PROD\AUDIO\audio\22.mp3
[End Song]
[Song]
Title=Back to Name
Artist=An Artist
Genre=Piano
Track=5
Time=302
CreateTime=800
ID=\\HMB-PROD\AUDIO\audio\Back to Name.wav
[End Song]
[Song]
Title=Pinch
Artist=Bare Back Band
Genre=Rock
Track=-1
Time=0
CreateTime=800
ID=\\HMB-PROD\AUDIO\audio\BARE BACK BAND - PINCH.MP3
[End Song]
[End Song List]
```

4.2.2.4 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:

```
[Play Queue]
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
CreateTime=800
ID=\\HMB-PROD\AUDIO\audio\20 kbits.mp3
[End Song 2]
[Song 3]
RandIndex=2
Title=Dig that test file
Album=MP3 Sample Rate Tests
Artist=AT Test
Track=-1
Time=0
CreateTime=800
ID=\\HMB-PROD\AUDIO\audio\22.mp3
[End Song 3]
[Song 4]
RandIndex=7
Title=Back to Title
Artist=Phil
Genre=Piano
Track=5
Time=302
CreateTime=800
ID=\\HMB-PROD\AUDIO\audio\Back to Title.wav
[End Song 4]
[Song 5]
RandIndex=12
Title=Pinch
Artist=BareBack Band
Genre=Rock
Track=-1
```

```
Time=0
CreateTime=800
ID=\\HMB-PROD\AUDIO\audio\BAREBACK BAND - PINCH.MP3
[End Song 5]
[End Play Queue]
```

4.2.2.5 Global format

The Global information format contains the following fields:

- Header line indicating global info reply
- Version - The current version of the Firmware in the the format MajorVer.MinorVer.DevVer.
- CurrTime - The Atron relative time that the request was processed.
- PlayQUpdateTime - The last time (in ATRON relative time) that the PlayQ changed.
- Statistics about each list. The stats include:
  - Count - The number of items in that list. Note: ItemCount - This is the number of items in the File list that will include songs, Streams (from Turtle Radio) and playlists.
  - LastChange - Indicates the last time in Atron relative time that this list changed.
- A list of Shares the Audiotron currently knows about. These were either found or configured.
- A list of Hosts the Audiotron knows about. Each host record has 3 components:
  - Host=<Name of the host>
  - IP=<IP address of the host>
  - State=<UP or DOWN> - This is the Audiotrons current belief about the state of the host.

```
[Global Info]
CurrTime=400
PlayQUpateTime=375
Version=2.1.15
Location=Default Location
ItemCount=0
SongCount=91
StreamCount=187
ItemLastChange=350
TitleCount=0
TitleLastChange=350
GenreCount=0
GenreLastChange=200
ArtistCount=0
ArtistLastChange=250
AlbumCount=0
AlbumLastChange=300
NetCount=0
NetLastChange=100
PlaylistCount=0
PlaylistLastChange=350
 [Share List]
Share=\\HMB-PROD\AUDIO
Share=\\HMB-PROD\HMBPRODD
Share=\\HMB-PROD\HMBPRODC
[End Share List]
[Host List]
[Host]
```

```
Host=ATRONSRC
IP=192.168.0.1
State=UP
[End Host]
[Host]
Host=HMB-PROD
IP=192.168.0.4
State=UP
[End Host]
[Host]
Host=HMB-PB
IP=192.168.0.9
State=UP
[End Host]
[Host]
Host=ATRON00190A
IP=192.168.0.10
State=UP
[End Host]
[Host]
Host=WMA-STREAM
IP=192.168.0.34
State=UP
[End Host]
[Host]
Host=HMB-PB30
IP=192.168.0.51
State=UP
[End Host]
[Host]
[End Host List]
[End Global Info]
```

4.2.2.6 FAV type

The information returned for the favorite is a list of all 20 favorites. The format for the favorite information is:

- Header line indicating that this is a Favorite information response
- All 20 Favorites each with 2 items: (in both cases the <xx> is the favorite number);
  - FavType=xxxx - Indicates the type of the favorite.
    - The valid values for type are:
      - **Play List -** normal Playlists as well as <Group> and <All Songs>
      - **Album -** Album names
      - **Genre -** Genres
      - **Artist -** Artists
      - **Title -** All song titles and net radio stations
      - **Net Radio -** Net radio categories
  - FavValue=yyyy - Indicates the value of the favorite.
- Footer Line indicating the end of the favorite list.

```
[Favorite List]
[Favorite 1]
FavType=Title
FavValue=01 - Flink Poyd - In the flesh-.wav
[End Favorite 1]
[Favorite 2]
FavType=Not Configured
FavValue=Not Configured
[End Favorite 2]
[Favorite 3]
FavType=Not Configured
FavValue=Not Configured
[End Favorite 3]
[Favorite 4]
FavType=Not Configured
FavValue=Not Configured
[End Favorite 4]
[Favorite 5]
FavType=Not Configured
FavValue=Not Configured
[End Favorite 5]
[Favorite 6]
FavType=Not Configured
FavValue=Not Configured
[End Favorite 6]
[Favorite 7]
FavType=Not Configured
FavValue=Not Configured
[End Favorite 7]
[Favorite 8]
FavType=Not Configured
FavValue=Not Configured
[End Favorite 8]
[Favorite 9]
FavType=Not Configured
```

```
FavValue=Not Configured
[End Favorite 9]
[Favorite 10]
FavType=Not Configured
FavValue=Not Configured
[End Favorite 10]
[Favorite 11]
FavType=Not Configured
FavValue=Not Configured
[End Favorite 11]
[Favorite 12]
FavType=Not Configured
FavValue=Not Configured
[End Favorite 12]
[Favorite 13]
FavType=Not Configured
FavValue=Not Configured
[End Favorite 13]
[Favorite 14]
FavType=Not Configured
FavValue=Not Configured
[End Favorite 14]
[Favorite 15]
FavType=Not Configured
FavValue=Not Configured
[End Favorite 15]
[Favorite 16]
FavType=Not Configured
FavValue=Not Configured
[End Favorite 16]
[Favorite 17]
FavType=Not Configured
FavValue=Not Configured
[End Favorite 17]
[Favorite 18]
FavType=Not Configured
FavValue=Not Configured
[End Favorite 18]
[Favorite 19]
FavType=Not Configured
FavValue=Not Configured
[End Favorite 19]
[Favorite 20]
FavType=Album
FavValue=Our Time in Eden
[End Favorite 20]
[End Favorite List]
```

4.3 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.

The favorites are numbered 0-15 for the favorites labeled 1-16. For the lettered favorites, A=16, B=17,C=18,D=19).

4.3.1 Filtering in apiqfile.asp

Filtering is also available for further refining the file queuing for types Artist, Album, and Genre. The syntax is the same as for filtering for apigetinfo.asp. The filter criteria that are available are: fartist, falbum, and fgenre. Any set of filter criteria that are not the same as the type can be specified on any request. Obviously, you can't filter type=album and a falbum filter.

For example,
http://<ipaddr>/apiqfile.asp?type=artist&file=The%20BandName&falbum=Their%201st%20Album
.
This would enqueue all of the songs by "The BandName" from an album called "Their 1st Album" for playback.

NOTE:The file type must be album for the songs to be enqueued in album order.

4.4 Apiaddfile.asp

This function allows for a file to be 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.

4.5 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]
CurrTime=1200
LastPlayQUpdate=400
CurrVolume=-10
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
CurrPlayTime=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
- CurrTime - Atron Relative time that the status response was generated.
- LastPlayQUpdate - The last time the play queue was updated.
- CurrVolume - The current volume in db.

- 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.
- CurrPlayTime - 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

4.6 Apimsg.asp

This function allows a temporary message to be displayed on the front panel of the Audiotron. The format for this request is:

http://<ipaddr>/apimsg.asp?line1=<text>&line2=<text>&timeout=<sec>.

This function will display the two indicated lines of text for the amount of time set by the timeout parameter. After this time the display will revert to the previous display.

The timeout parameter is optional. If it is not present the text will be displayed until user action on the front panel or remote control or a msgoff command is received by the api.

4.7 Apidumptoc.asp

This function will dump a properly formatted Table of Contents file. The format for this request is:

http://<ipaddr>/apidumptoc.asp?share=<sharename>. For example, http://<ipaddr>/apidumptoc.asp?share=\\hmb-prod\audio will return the contents of a TOC for the share \\hmb-prod\audio

5 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 its 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.

5.1 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.

Note: A SONG is actually a more generic entity more closely meaning file . These SONG records can also contain descriptors for Playlist files and Firmware update files (e.g. nb0).

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
```

5.1.1 TOC Tags

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

"UPDT" - Tells the Audiotron to search the share for songs after processing the TOC. This gives the TOC maintenance engine a way of telling the Audiotron to use its search mechanism but to use the TOC to provide a cache of most of the data. This Tag can be anywhere inside the TOC file.

"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.
Does not contain any directory information.

"DIR " - Directory information associated with this file.
Contains the path from the top of the share to the file in question. NOTE THE SPACE.

– If the file is in the same directory as the TOC file, this entry should be blank (but still present)
– If the file is in a subdirectory of the TOC directory, this entry must end with a backslash but not start with one

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>.

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

5.1.2 TOC Performance Notes

Background

The Audiotron maintains much of its information in alphabetic lists. There is a separate list for the songs, artists, albums, genres, playlists and titles. As such much of the processing required for processing TOCs is associated with building these lists. Thus the key to maximizing performance for TOCs is to minimize the amount of work required to build these lists.

The two longest lists in the Audiotron are the Title and the Song lists. Thus minimizing the overhead of building these two lists will greatly enhance the performance of the TOC processing.

Guidelines

In order for the TOC file processing to maximize performance, the following guidelines should be observed:

– Alphabetic order - There is code in the TOC processing to speed up adding entries that are added in alphabetic order. Thus the first guideline is to alphabetize the TOC by the filename. This will minimize the time to generate the song list.
– File name == title - If the TOC is alphabetized by file name and the song title and the filename are the same, the TOC is by definition alphabetized by title. This will minimize time for inserting titles into the title list.

- Put all music in one share (and thus one TOC). This way all songs will be added alphabetically. If the Audiotron needs to process 2 sets of files, the entries in the later one will be inserted into the list rather than added at the end.