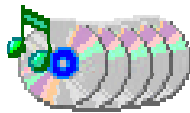


# Technical Reference

**Disc Library / Music Library**



October 2006


Copyright © 2006 William C. McCain



## Configuring a Digital Audio Switch

Disc Library and Music Library support a digital audio switch, which allows your music players to share a single digital input on your receiver or preamp. Whenever the current music player changes, the corresponding input on the digital audio switch is automatically selected. The supported digital audio switches are the Inday DA4X-RS (configured on a serial COM port) and the Nirvis DXS (automatically detected on a Slink-e or SAVR/SAVR2 attached to the Master).

By default, the music players are assumed to be attached to the inputs of the digital audio switch in “player number order”. That is, Player 1 is attached to input 1, Player 2 is attached to input 2, *etc.* By default, the input is automatically switched only on output 1 of the digital audio switch (the DA4X-RS has only one output, but the DXS can have as many as 8 independently-switched outputs). However, you can override these defaults by supplying a configuration file named **DAS.CFG**, which must reside in the Disc Library or Music Library product folder.

 If a serial COM port is configured a Control port, the digital audio switch is *not* automatically switched by default and the **DAS.CFG** file is not utilized. Instead, the automation controller connected to the serial Control port must send a “digital audio switch configuration” command to activate the switch. This command can be sent more than once, which permits the automation controller to dynamically switch reserved (non-music-player) inputs to any of the available digital outputs.

**DAS.CFG** consists of one line of text, which is a list of numbers separated by blanks, like this:

**$r$**   **$o_1$**   **$o_2$**   **$\dots$**   **$o_n$**

The first number,  **$r$** , specifies how many inputs to reserve (*i.e.*, inputs that are not used for Disc Library or Music Library music players). The music players are still connected to the switch in “player number order”, but if  **$r$**  is greater than 0, then Player 1 is attached to input  **$r + 1$**  (origin-1), skipping the first  **$r$**  inputs. The remaining numbers,  **$o_1$**   **$o_2$**   **$\dots$**   **$o_n$** , correspond to the switch’s outputs, in order.

There must be at least two numbers in this line of text:  **$r$**  must be specified (it can be 0) and  **$o_1$**  must be specified, configuring the first output (the remaining outputs can be left unconfigured, in which case they will not be switched).

The numbers  **$o_1$** ,  **$o_2$** , *etc.*, specify the origin-0 input number to be switched to that output. If the number is *less than*  **$r$** , which is the number of reserved inputs, then it is the origin-0 value of one of those *reserved* inputs. In that case, the specified reserved input is switched to that output during Disc Library or Music Library startup.

## Technical Reference for Disc Library and Music Library

If the number is equal to (or greater than)  $r$ , then it means that the corresponding output will be automatically switched to the current music player, as it changes.

Note that a number that is equal to (or greater than)  $r$  does not really refer to one specific input, it refers to all inputs that are connected to music players. Logically, it means “switch this output to whichever input is the current music player”.

There are two special cases:

- For Music Library only,  $r$ , the number of reserved inputs, can be  $-1$ . This means that the music file player (your computer) will be skipped and not switched to a digital switch output. In that case, Player 2 (the first CD changer) must be connected to input 1 (in origin-1 notation), Player 3 must be connected to input 2, *etc.*
- The number  $16$  can be specified for any output. Since that number is higher than the origin-0 value of any input, it means that the corresponding output will be ignored (not switched). At least *one* output must be configured for music players (*i.e.*, the number specified for at least one output must be equal to or higher than  $r$  but less than the number of inputs on the digital switch).

The default configuration is equivalent to a **DAS.CFG** file that contains the following line:

```
0 0
```

## Serial Control Interface

If you have an automation controller, such as those made by AMX and Crestron, you can configure a serial COM port on your computer as a Control port. This provides an interface between the Disc Library or Music Library Master and the automation controller. Using this interface, panels (typically, touch screens) connected to the automation controller can display channel, artist, and song title information for the selection currently playing in Disc Library or Music Library. The automation controller's panels can also provide buttons or other controls that let you change the selection in Disc Library or Music Library.

Command messages flow from the automation controller to the Disc Library or Music Library Master. Status messages flow from the Disc Library or Music Library Master to the automation controller. There is no synchronous relationship between command messages and status messages. An invalid command is simply ignored, there is no error message in response. A valid command may have no response (for example, if it directs Disc Library or Music Library to “Play” when it is already playing) or it may indirectly produce several status messages (which contain new status and title information). The automation controller can receive status messages when no command has been issued (for example, when the current track ends and a new track starts, or the current selection is changed by some other method).

## Technical Reference for Disc Library and Music Library

The Control port interface operates at 38400 bps, 8 data bits, 1 stop bit, no parity. These parameters are not configurable. Special programming in your automation controller is required to use this interface. Details of the serial control interface protocol follow.

### IP Interface

Disc Library and Music Library have Remote versions that use a private IP protocol to communicate with the Disc Library or Music Library Master. This IP protocol normally uses IP port 62832, but it can be configured to use a different port number. In addition to the private IP protocol, the Remote uses the standard HTTP protocol on port 8080 (which can also be configured to a different port number). The Remote uses the HTTP protocol to retrieve files from the Master: database files, cover image files, and play lists.

As with the serial control interface, command messages flow from the Disc Library or Music Library Remote to the Master. Status messages flow from the Disc Library or Music Library Master to the Remote. There is no synchronous relationship between command messages and status messages. An invalid command is simply ignored, there is no error message in response. A valid command may have no response or it may indirectly produce several status messages. The Remote can receive status messages when no command has been issued.

The private IP protocol is very similar to the serial control interface. In fact, the commands have the same format and the command sets are nearly identical (the serial control interface has one additional command not supported by the IP interface). Unlike the serial control interface, the IP interface can easily be used by programs running on the same computer as the Disc Library or Music Library Master. Details of the IP interface protocol follow.

### Serial and IP Interface Details

#### General Considerations

All command messages and status messages are in text format (no binary data is involved). Each message consists of a single line of text, ending with a "line termination character". For the serial control interface, the line termination character on all messages (command and status) is the Carriage Return character (hexadecimal code 0D or decimal code 13). For the IP interface, the line termination character on all messages (command and status) is the Line Feed character (hexadecimal code 0A or decimal code 10). All text characters are in the extended ASCII character set known as "Windows Code Page 1252"

## Technical Reference for Disc Library and Music Library

(which means that accented foreign letters may be present, as well as uncommon special characters such as "™").

Wherever the term "disc" is used in the following interface specifications, it should be understood that the term could also refer to a music file (in the case of Music Library)

### Parameters

In messages that use an explicit player number, it is specified as an origin-1 number. Disc numbers are also specified in origin-1 notation, the same way they are numbered by the CD changers. Track numbers are generally "absolute", which means they are the numbers reported by the CD itself. (The exception is the "track title list" status messages in the serial control interface. These messages show the track number in origin-1 notation, even in the rare case that the CD reports a number higher than 1 for its first track.) Valid absolute track numbers can range from 1 through 99.

The Play List Number parameter in the play list commands and status messages refers to the numeric identifier assigned to each play list in the **PLAYLIST.LST** file (in the Disc Library or Music Library product folder). This number can range from 0 though 99999. The Step Index parameter in the play list commands and status messages is specified in origin-0 notation.

The "Set special play mode" command has two numeric parameters, Mode Number and Category Flags, which require further explanation. The following two tables give the definitions of these parameters. (The same definitions apply to the Mode Number and Category Flags parameters in the corresponding "Special mode set" status message, in the IP interface.)

Mode Number	Meaning
0	Normal.
1	Repeat Track.
2	Repeat Album.
3	Shuffle Tracks on Current Album.
4	Shuffle Tracks in Disc/Music Library.
5	Shuffle Albums in Disc/Music Library.

## Technical Reference for Disc Library and Music Library

The Category Flags parameter is a number formed from 15 individual low-order bit flags, defined as follows:

Category Flags	Meaning
.... .... .... ...1	Blues.
.... .... .... ..1.	Classical.
.... .... .... .1..	Country.
.... .... .... 1...	Data.
.... .... ...1 ....	Folk.
.... .... ..1. ....	Jazz.
.... .... .1.. ....	Miscellaneous.
.... .... 1... ....	Musical Show.
.... ...1 .... ....	New Age.
.... ..1. .... ....	Opera.
.... .1.. .... ....	Pop.
.... 1... .... ....	Reggae.
...1 .... .... ....	Rock.
..1. .... .... ....	Soundtrack.
.1.. .... .... ....	Unknown.

## Command Messages

Command messages have the format "**p c ...**", followed by the required line termination character, where "**p**" is the player number (in decimal characters), "**c**" is a command code, and "**...**" stands for zero or more blank-delimited parameters (which depend on the command code). For many simple commands (such as "Play"), "**p**" is set to "0" (meaning "current player"), and there are no parameters.

There is a "special" command, not shown in the following table: "?" (a line containing just a single question mark, followed by the line termination character). This command means "Query status". It requests the Master to return one or more status messages that inform the automation controller or Remote as to the current state of the Master.

## Technical Reference for Disc Library and Music Library

The supported command messages are defined as follows:

Player	Command Code	Parameters	Function
<i>p</i>	720898	<i>disc</i>	Load disc paused.
<i>p</i>	720899	<i>disc</i>	Load disc and play.
<i>p</i>	720900	<i>disc track</i>	Load disc/track paused.
<i>p</i>	720901	<i>disc track</i>	Load disc/track and play.
0	720909	<i>track</i>	Move to track on current disc.
0	720910	(none)	Stop.
0	720911	(none)	Pause.
0	720912	(none)	Play.
0	720913	(none)	Previous track.
0	720914	(none)	Start reverse scan.
0	720915	(none)	Start forward scan.
0	720916	(none)	Next track.
0	720917	(none)	Previous disc or step.
0	720918	(none)	Next disc or step.
0	720919	<i>mode-no. cat.-flags</i>	Set special play mode.
0	720923	<i>list-no. step-index</i>	Start play list paused.
0	720924	<i>list-no. step-index</i>	Start play list playing.
0	720925	<i>step-index</i>	Move to step in current play list.
0	720930	(none)	End reverse scan.
0	720931	(none)	End forward scan.

So to send a "Play" command, for example, the string to send is "0 720912", followed as always by the line termination character.

## Technical Reference for Disc Library and Music Library

### Serial Control Status Messages

Serial control status messages have a fixed-column format (which makes them easier to parse for a “language and performance challenged” processor such as an AMX or Crestron automation controller). The following table lists these status messages, with an explicit text character for every required column position. Uppercase letters and explicit decimal digits are received exactly as shown; lowercase letters represent decimal digits, with leading zeros, that can vary from message to message. The sequence “**ppp**” is always the player number, as three decimal digits, in origin-1 notation.

Status Message	Parameter Fields	Meaning
<b>DASniiiiooo</b>	<b>n</b> : 0 for DXS, 1 for DA4X-RS <b>iii</b> : number of inputs, three digits <b>ooo</b> : number of outputs, three digits.	Digital audio switch is present.
<b>ppp0ddddddddd</b>	<b>ddddddddd</b> : next disc number, seven digits (0000000 if unknown).	Disc unloaded.
<b>ppp1</b>	(none)	Disc stopped.
<b>ppp2</b>	(none)	Disc paused.
<b>ppp3</b>	(none)	Disc playing.
<b>ppp4dddddddddfff</b>	<b>ddddddddd</b> : disc number, seven digits <b>fff</b> : absolute first track number, three digits (usually 001, but the CD standard allows for discs that start with a track number other than 1).	Disc loaded.
<b>pppD...</b>	<b>...</b> : current album title, variable length (max 32).	Album title.
<b>pppA...</b>	<b>...</b> : current album artist, variable length (max 32).	Album artist.
<b>pppMttt...</b>	<b>ttt</b> : track number, three digits (origin-1) <b>...</b> : track title, variable length (max 32).	Track title (other than last track).
<b>pppNttt...</b>	<b>ttt</b> : track number, three digits (origin-1) <b>...</b> : track title, variable length (max 32).	Track title (last track). End of track title list.
<b>ppp5ttt</b>	<b>ttt</b> : track number, three digits (absolute).	Track starting.
<b>000E...</b>	<b>...</b> : error message text, variable length.	Error message.

## Technical Reference for Disc Library and Music Library

### Digital Audio Switch Command

For the serial control interface only, there is an additional "special" command, not shown in the "Command Messages" table, that configures and controls the digital audio switch, if one is present. (The switch can be an Inday DA4X-RS configured on a serial COM port or a Nirvis DXS on a Slink-e or SAVR/SAVR2 attached to the Master.)

This "digital audio switch configuration" command consists of the explicit lowercase letter "x", followed by a blank, followed by a text string that has the same format as the **DAS.CFG** file. (See the section entitled "Configuring a Digital Audio Switch".) The automation controller would normally send this command in response to the "Digital audio switch is present" status message, shown in the preceding table. The automation controller can send this command again at any time, for the purpose of switching any of the switch's outputs between the current music player and one of the reserved inputs.

### Alphabetical List Utility Program

Note that, as shown in the preceding table, the serial control interface has several unique status messages, not present in the IP interface, which convey text data concerning the *current* album title and artist and track titles. This is because the serial-connected automation controller, in general, lacks the ability to use HTTP to retrieve this information from the Disc Library or Music Library database.

Furthermore, there is no facility, in either the serial interface or the private IP interface, for retrieving a "catalog" of album titles, performers, or composers from the Master. The Remote program uses the HTTP interface to obtain that information, by reading the database files from the Master computer.

However, there is a special utility program, **make\_lists**, distributed with Disc Library and Music Library. This program outputs alphabetical lists of the database by album title, performer, and composer, in the form of AMX "Acess/NetLinX language" source code. This allows the database lists to be compiled into an AMX automation control program, which is desirable since a typical automation controller lacks the capability or performance needed to retrieve that information dynamically. These AMX source code files can easily be "hand-converted" to a similar Crestron source code language, if needed.

**make\_lists** is designed to be run from the command prompt. The Disc Library or Music Library Master must be configured for Remote support and must be active. **make\_lists** can be run on the same computer as the Master, or on a different computer; it uses the same IP interface as the Remote (and can use the same configuration file). The IP address of the Master (and the IP ports, if you are not using the defaults) can be specified as command-line parameters.

## Technical Reference for Disc Library and Music Library

If you have configured the Remote program on the computer on which `make_lists` is to run, you can omit the command-line parameters (in which case, the IP address and port numbers will be read from the Remote configuration file).

To run `make_lists`, start a command prompt window, make the Disc Library or Music Library product folder your current directory, and enter the following command:

```
java -cp . make_lists [ip-addr [ip-port/http-port]]
```

If the `java.exe` file is not in your default search path, you would need to replace `java` in the above command with the complete path to the java executable. The square brackets [...] in the above are not actually entered — they indicate that the parameters within are optional.

The `make_lists` program produces three AMX language “include” files, which are created in the Disc Library or Music Library product folder. For Disc Library, the files are named `DL_SIZES.AXI`, `DL_DATA.AXI` (for Access), and `DL_ARRAYS.AXI` (for NetLinx). For Music Library, the files are named `ML_SIZES.AXI`, `ML_DATA.AXI` (for Access), and `ML_ARRAYS.AXI` (for NetLinx).

## IP Interface Status Messages

IP interface status messages have a blank-delimited format. Uppercase letters and explicit decimal digits are received exactly as shown; lowercase letters represent decimal digits, *without leading zeros*, that can vary from message to message. The lowercase letter “*p*” is always the player number, in origin-1 notation.

Note that there is a set of “player configuration” status messages that flow automatically when the IP connection is initially established. The first of these messages, the “Players configured” message in the following table, gives the total number of music players currently configured for Disc Library or Music Library. This is followed by a sequence of “Player list item” messages, one for each configured player. So if *n* music players are configured, a sequence of *n* + 1 “player configuration” status messages will be received by the Remote immediately after the Master has accepted an IP connection request on the private IP protocol port.

In addition to the “player configuration” messages, there are several other status messages present in the IP interface that are not found in the serial control interface. These are the status messages for “Track elapsed time” (one per second), “Special mode set”, and the three status messages regarding play list operation.

## Technical Reference for Disc Library and Music Library

The following table lists the IP interface status messages:

Status Message	Parameter Fields	Meaning
<b>0 P n</b>	<b>n</b> : total number of music players.	Players configured.
<b>p P ... .. c</b>	<b>...</b> : player brand name <b>...</b> : player model name <b>c</b> : player capacity (e.g. <b>100</b> , <b>200</b> , etc.).	Player list item.
<b>p 0 d</b>	<b>d</b> : next disc number ( <b>0</b> if unknown).	Disc unloaded.
<b>p 1</b>	(none)	Disc stopped.
<b>p 2</b>	(none)	Disc paused.
<b>p 3</b>	(none)	Disc playing.
<b>p 4 d f l m s</b>	<b>d</b> : disc number, seven digits <b>f</b> : absolute first track number <b>l</b> : absolute last track number <b>m</b> : total disc time (minutes) <b>s</b> : total disc time (seconds).	Disc loaded.
<b>p 5 t sm ss tm ts</b>	<b>t</b> : absolute track number <b>sm</b> : start time on disc (minutes) <b>ss</b> : start time on disc (seconds) <b>tm</b> : track playing time (minutes) <b>ts</b> : track playing time (seconds)	Track starting.
<b>p 6 i m s</b>	<b>i</b> : index ( <b>0</b> for lead-in time, <b>1</b> for content) <b>m</b> : elapsed track time (minutes) <b>s</b> : elapsed track time (seconds).	Track elapsed time.
<b>p 7 m f</b>	<b>m</b> : mode number <b>f</b> : category flags for shuffle modes.	Special mode set.
<b>p 8 n i l</b>	<b>n</b> : play list number <b>i</b> : index of first step to be played <b>l</b> : number of steps in play list.	Play list starting.
<b>p 9 i</b>	<b>i</b> : index of step starting to be played.	Play list step starting.
<b>p 10</b>	(none)	Play list stopped.
<b>0 E ...</b>	<b>...</b> : error message text, variable length.	Error message.

# Database File Formats

## Overview

As explained in the *User Guide*, each CD changer is represented by a “slot file” in the Disc Library or Music Library product folder. The naming convention for these slot files is discussed in detail in the *User Guide*. Each slot file lists the contents of a single CD changer, in “slot number” order. In the case of Music Library, an additional slot file, named **MUSIC.001**, lists the music files that have been imported into Music Library. The music files are assigned “slot numbers” in the order in which they are imported (except that the slot numbers of deleted music files can be reused).

The slot files are actually simple text files, each line of which describes one slot. Each “slot line” contains a slot number and the name of a “database entry” file in the **DATABASE** subfolder of the product folder. In the case of Disc Library, the **DATABASE** subfolder itself contains the database entry files. In the case of Music Library, the **DATABASE** subfolder contains two further subfolders, **DISCS** and **MUSIC**; the **DISCS** subfolder contains the database entry files for CDs, and the **MUSIC** subfolder contains the database entry files for music files.

The database entry files are also simple text files. Each line of a database entry file contains a specific piece of information about a CD or music file.

## Slot File Format for CD Changers

Each line of a CD changer slot file has the following format:

**dddd:HHHHHHHH.EEE-nnn-fff.eee**

where **dddd** is the slot number (5 digits, with leading zeros), **HHHHHHHH.EEE** is the name of the database entry file (12 characters, including the period), **nnn** is the number of database entry files with the same hash code (3 digits, with leading zeros), and **fff.eee** is the name of the cover image file in the **COVERS** subfolder of the product folder (variable length). The colon and dashes that separate these fields are literal, they must be present exactly as shown.

- The first 8 characters of the database entry filename, **HHHHHHHH** in the above format description, are the “CD database ID”; this is a hash code that is calculated from the CD’s table of contents (TOC). It is used to look up the CD in the Internet database (freedb.org) as well as in the local Disc Library or Music Library database.
- The last 3 characters of the database entry filename, **EEE** in the above format description, are used to distinguish multiple database entries for the same CD.

## Technical Reference for Disc Library and Music Library

These “filename extensions” are always 3 characters (decimal digits, except under rare conditions involving abnormal retrieval of entries from the Internet).

- Database entries returned from freedb.org are given filenames with extensions from **501** through **999**. (The Internet database can return multiple entries for a single query. The resulting local files are “read only” and they are never modified after they are retrieved.)
- User entries, which can be created by editing a freedb entry or by entering data manually for a CD that is not found in the Internet database, are given extensions from **001** through **499** (extension **500** is not used).
- The extension **000** is special, it is used for a file that holds the TOC data for a disc that is looked up in the Internet database but is “not found”. This special temporary file is deleted automatically whenever the user creates an entry manually or if the CD later becomes “found” in the Internet database.
- The count of database entries with the same hash code, **nnn** in the above format description, is not presently used for any purpose by Disc Library or Music Library. However, it is maintained accurately, so it may be informative to the human user who examines a slot file.
- The final dash and the cover image filename, **-fff.eee** in the above format description, are optional. If they are absent, then no cover image has been assigned to the slot. In that case, a generic cover image is used instead; the *category* field in the actual database entry determines the generic cover image.

Note that all fields except the last one (the cover image filename) are fixed-length. Thus, each line of a CD changer slot file has a “fixed column” format, in which every field starts at a predictable column within the line.

The following are examples of lines in a CD changer slot file:

```
00001:E70DAF10.501-002-Dido_and_Aeneas.jpg
00181:6F0C8008.001-005-the_book_of_secrets.jpg
```

Disc Library and Music Library never re-query the Internet database when one or more freedb-derived entries already exist for a given “CD database ID” (hash code). Whenever a lookup operation is performed, on *any* CD changer, the hash code is calculated from the CD’s table of contents. Then the local database is checked first for entries with filenames consisting of that hash code and a filename extension from **501** through **999**. If any such entries are found, these previously-downloaded entries are used and the Internet database is not queried. To *force* a re-query for a particular hash code (to actually re-query the Internet database), one must manually delete all the existing database files whose filenames start with that hash code and end with an extension of **501** through **999**.

In any case in which the “CD database ID” **HHHHHHHH** for a slot stays the same but the “entry selection” **EEE** is changed, the cover image filename, if present, is

## Technical Reference for Disc Library and Music Library

preserved (for instance, if the user selects a different database entry for the same CD).

### Database Entry Format for CDs

Each database entry file is a simple text file. Each line of such a file contains specific information about a single CD. Using origin-1 notation to refer to the lines in the file (*i.e.*, the first line is "line 1"), the file organization is described as follows:

Line Number	Content
1	Internet database Query command string.
2	Internet database Query response string.
3	Genre (line may be empty).
4	Album title.
5	Artist name.
6	Year (4 decimal digits or empty line).
7	Performer, alphabetical.
8	Composer, alphabetical (line may be empty).
9	Information/comments (line may be empty).
10	[Reserved].
11	Track title (1 <sup>st</sup> track).
...	...
109	Track title (99 <sup>th</sup> track).

If the query fails, that is, if the CD is not found in the Internet database, a local database entry file is created with the filename extension **000**. This special entry contains only line 1. All other local database entry files (with extensions from **001** through **999**) contain at least the first 11 lines shown above (and possibly more, depending on the number of tracks, which can range from 1 to 99).

When the user manually enters data for a CD that was not found in the Internet database, the special entry with extension **000** is used as a "template", in the sense that it tells the Edit Database Entry dialog how many tracks the CD has. Also, the special **000** entry provides the text string that is used for line 1 of the "user entry" that results when the user saves the manually entered data.

## Technical Reference for Disc Library and Music Library

Line 1, the Internet database Query command string, contains the hash code and information about the number of tracks and their timings. The following is an example, taken from a database entry file named **4B0c3706.501**:

**4b0c3706+6+150+59025+101250+114900+159075+209775+3129**

In the above example, the first 8 characters are the hash code; this will always agree with the first 8 characters of the filename (except that the hexadecimal digits are in lowercase). The remaining information indicates that this CD contains 6 tracks, gives the start time of each track (in "frames"), and the total playing time of the CD (in seconds). (There are 75 frames per second.) All of this information is obtained by reading the CD's table of contents. For further details, including the algorithm for calculating the hash code, see the interface specifications posted at freedb.org.

Normally, the first track on a CD is identified as "track 1", but it is possible for a CD to start with a track number other than 1. (This is extremely rare.) In that case, and only in that case, line 1 is further augmented by appending **:ff** to the end of the Internet database Query command string, where **ff** is one or two decimal digits that give the track number of the first track.

In the case of Sony CD changers, the track times are not readily available during playback, so the track times from line 1 of the database entry are used for certain calculations. (For instance, for converting a track's "elapsed" time into the track's "remaining" time during playback.)

Line 2, the Internet database Query response string, contains the category, a hash code, artist name, and album title. There is a single blank between the category and the hash code and also between the hash code and the artist name. The three characters " / " separate the artist name and the album title. The following is an example, taken from the same database entry file as the line 1 example:

**classical 4b0c3706 Wagner / Preludes And Overtures**

The category string at the start of line 2 will always be one of the names in the following table. In the case of a database entry returned by freedb.org, the category name will be one of the 11 names shown in the first three columns. User-edited entries can use any of the 15 names shown in all four columns.

Category Name			
Standard			Extended
blues	folk	reggae	musical
classical	jazz	rock	opera
country	misc	soundtrack	pop
data	newage		unknown

## Technical Reference for Disc Library and Music Library

Note that the hash code in line 2 will not necessarily agree with the hash code in line 1 (and in the filename). That is because, if an exact match for the hash code is not found in the Internet database, the Query command may return one or more entries that are “fuzzy” matches to the original Query command string. In that case, the hash code in line 2 will be the “true” hash code for the entry in the Internet database.

A query to the Internet database can return multiple entries either for an exact match of the hash code or for a “fuzzy” match. However, in all cases in which multiple entries are returned, the Disc Library or Music Library log messages describe the result as an “inexact match”. That is because of the ambiguity of having multiple entries from which to choose. Database entry files are created for all entries returned, and the first such file is listed in the CD changer’s slot file. The user should examine these entries later and choose the one that is most appropriate.

When multiple entries are returned for an *exact* match of the hash code, the multiple freedb-derived entries will each have a different category on line 2, because that is the way that the Internet database resolves “hashing conflicts”. Sometimes these multiple entries will occur because two truly different CDs produce the same hash code (the “hashing algorithm” is not very sophisticated). More often, however, it happens because multiple submissions have been made to the Internet database for the same CD. Sometimes people disagree about the category that should be assigned, so the same CD gets submitted under multiple categories. Other times, people artificially change the category in order to submit a slightly different version of the entry — with a different “genre” perhaps, or a different “information” field, or even different spelling and punctuation in the artist name, album title, and/or track titles.

When a “fuzzy” match occurs on Internet lookup, the Internet database returns *all* of the entries that are considered to have a “fuzzy” match with the Query command string. In this case, one can get multiple entries with the *same* category but with different “true” hash codes. (These two fields together, category and true hash code, form the complete “key” under which entries are filed in the Internet database.)

Multiple entries of the “fuzzy” kind can refer to entirely different CDs, of course. But they often refer to the same CD, with different hash codes, and that happens for two reasons. One is that different pressings, or “re-issues”, of the same CD sometimes have very slightly different track times, usually due to variations in the “lead-in” time between tracks. This will affect the hash code calculation. The other reason is that some of the programs that do the submissions to the Internet database have slight defects in their hash code calculations (such as truncating track times instead of rounding them).

For database entries retrieved from the Internet, one can generally expect that the lines containing the album title, artist name, and all the track titles will be non-empty, with non-blank data. The “performer, alphabetical” field (line 7) is derived from the artist name, so it will be non-blank as well. (An attempt is made to convert

## Technical Reference for Disc Library and Music Library

the artist name to “last name, first name” format when line 7 is generated.) In a “user entry”, it is possible that the user could have deleted any or all of these non-blank items, in which case the corresponding lines in the user entry file will be empty. (This is unlikely, however.)

The Internet database does not contain a specific field for the composer name. However, for certain categories (classical, misc, and soundtrack), the album title, artist name, and “information” fields are scanned for the family name of a “known” composer; if found, the composer’s full name, in “last name, first name” format, is inserted in the “composer, alphabetical” field (line 8).

### Slot File Format for Music Files

Each line of **MUSIC.001** has the following format:

**ddddddd:HHHHHHHH.EEE-001-fff.eee**

where **ddddddd** is the slot number (7 digits, with leading zeros), **HHHHHHHH.EEE** is the name of the database entry file (12 characters, including the period), and **fff.eee** is the name of the cover image file in the **COVERS** subfolder of the product folder (variable length). The colon, dashes, and **001** that separate these fields are literal, they must be present exactly as shown.

- The lowest slot number that is generated for a music file is **0000001** (**0000000** is not used).
- The first 8 characters of the database entry filename, **HHHHHHHH** in the above format description, are a hash code that is calculated from the internal contents of the music file. It is possible for different music files to have the same hash code.
- The last 3 characters of the database entry filename, **EEE** in the above format description, are used to distinguish multiple database entries that have the same hash code. This can happen because the user has imported the same music file from different locations (different pathnames). It can also happen that the same hash code is calculated for entirely different music files. These “filename extensions” are always 3 decimal digits, and they range from **001** through **999** (extension **000** is not used).
- The final dash and the cover image filename, **-fff.eee** in the above format description, are optional. If they are absent, then no cover image has been assigned to the slot. In that case, a generic cover image is used instead; the *category* field in the actual database entry determines the generic cover image.

Note that all fields except the last one (the cover image filename) are fixed-length. Thus, each line of **MUSIC.001** has a “fixed column” format, in which every field starts at a predictable column within the line.

## Technical Reference for Disc Library and Music Library

The following is an example of a line in `MUSIC.001`:

```
0000005:EAD2B89B.001-001-Used_Record_File.jpg
```

### Database Entry Format for Music Files

Each database entry file is a simple text file. Each line of such a file contains specific information about a single music file. Using origin-1 notation to refer to the lines in the file (*i.e.*, the first line is "line 1"), the file organization is described as follows:

Line Number	Content
1	Complete filename (including path).
2	Category, playing time, artist name, album title.
3	Genre (line may be empty).
4	Album title (line may be empty).
5	Artist name (line may be empty).
6	Year (4 decimal digits or empty line).
7	Performer, alphabetical (line may be empty).
8	Composer, alphabetical (line may be empty).
9	Information/comments (line may be empty).
10	[Reserved].
11	Track title (line may be empty).

Each unique music file (each unique pathname) has exactly one database entry file, created when the music file is imported. If the music file has embedded "metadata tags", they are processed during import to obtain the data in lines 2 through 11. If the user edits the database entry, the original entry is modified; unlike the CD database, no additional entries are ever created for the same item. (However, the original metadata can always be recovered by deleting the entry from the database and re-importing the music file.)

If the music file contains no metadata tags and the user has never edited the entry, the database entry file will contain only the first one or two lines. (Line 2 will be present if the playing time is known; otherwise the database entry file will contain only line 1.) If the music file *does* contain metadata tags, or the user *has* edited the entry, all 11 lines shown above will be present in the database entry file.

## Technical Reference for Disc Library and Music Library

The following is an example of line 1:

```
C:\Music\Henry Purcell\Dido and Aeneas\Overture.ape
```

Line 2 has the same format as line 2 in the CD database entry files, except that the hash code is replaced by the music file's playing time, in seconds. The following is an example:

```
newage 306 Scott Cossu / Islands
```

The category string at the start of line 2 will always be one of the 15 names in the "Category Name" table, in the section entitled "Database Entry Format for CDs". The "category" concept is intrinsic to the Internet database for CDs. However, the metadata tags in music files do not contain category information, as such. In general, the metadata tags *do* provide "genre" strings; consequently, the category field is derived from the genre field during import. If no genre information is present in the music file, or if the genre string does not contain a recognizable substring that implies a category, then the category field will be set to **unknown**.

The playing time will be a dash if it is unavailable (which is the case for WAV files and other uncompressed formats). It is possible for the artist name and album title to be missing (but the " / " will always be present).

After the metadata tags are processed, if the artist name field is non-empty, then the "performer, alphabetical" field (line 7) is automatically derived from the artist name (converted to "last name, first name" format, if possible). The composer name may have been found in a "composer tag" in the metadata; if not, then for certain categories (classical, misc, musical, opera, and soundtrack), the album title, artist name, and "comments" fields are scanned for the family name of a "known" composer. In either case, if the composer's name is available, it is placed in the "composer, alphabetical" field (line 8) and converted to "last name, first name" format (if possible).

### Extending the Automatic Conversions

The source code files for the "first name list" and "composer name list" are included with the Disc Library and Music Library products. The "first name list" is compiled from **first\_name.java** and it is used to generate the "performer, alphabetical" field. The "composer name list" is compiled from **composer.java** and it is used to scan for composer names and generate the "composer, alphabetical" field. Both of these source code files can easily be modified and/or extended and re-compiled by a user (even by a non-programmer, as the source code files contain no "logic", they contain only "text string data").

### Play List Format

The Disc Library or Music Library product folder contains a “slot file” named **PLAYLIST.LST**, which lists all the available play lists. The play lists themselves are located in the **JUKEBOX** subfolder of the product folder. The **PLAYLIST.LST** file exists for two reasons:

- The Remote version obtains the list of available play lists by using HTTP to retrieve the **PLAYLIST.LST** file. (The Disc Library or Music Library Master operates a simple, highly secure HTTP server. This HTTP server supports retrieval of files from the product folder and its subfolders, but it does not support the retrieval of “directory lists”.)
- The “slot numbers” in the **PLAYLIST.LST** file simplify the serial control and IP interfaces. They enable the automation controller and the Remote version to refer to play lists “by number” rather than “by name”.

**PLAYLIST.LST** is a simple text file, each line of which describes one play list. Each line of **PLAYLIST.LST** has the following format:

**dddd:fff**

where **dddd** is the slot number (5 digits, with leading zeros) and **fff** is the variable-length name of the play list in the **JUKEBOX** subfolder of the product folder, *not* including the period and filename extension (which is always **.jpl**). Slot numbers can be anything from **00000** to **99999**. The colon that separates these fields is literal, it must be present exactly as shown.

The following is an example of a line in **PLAYLIST.LST**:

**00019:Send More Chuck Berry!**

Play lists are simple text files, each line of which describes one play list step. The lines are in ascending “step number order” (the first line is the first step, the second line is the second step, *etc.*). Each line of a play list file has the following format:

**ppp:dddd:tt**

where **ppp** is the player number (3 digits, with leading zeros), **dddd** is the slot number (5 digits for CDs, 7 digits for music files, with leading zeros), and **tt** is the track number (2 digits, with leading zeros). The colons that separate these fields are literal, they must be present exactly as shown.

With Disc Library, the player number is in origin-1 notation, but with Music Library, the player number is in origin-0 notation. This was done because Music Library added a new “player 1” (the music file player), which forces all the other player numbers to be higher by 1. By changing the “origin convention”, it became possible to copy play lists unchanged from Disc Library to Music Library (for users who

## Technical Reference for Disc Library and Music Library

upgrade). Thus, player number 000 represents the music file player, and it is found in Music Library play lists but not in Disc Library play lists.

In both Disc Library and Music Library, the slot number is in origin-1 notation, exactly as in the players' slot files, and the track number is "absolute" (from 01 through 99) or "--" (for a "disc step" that plays all of the tracks on a CD).

The following are examples of lines in a play list:

```
002:00181:04
000:0010327:--
003:00116:06
001:00095:--
003:00116:18
```

## Extending Support for Music Files

The music file support in Music Library is "user extensible". This means, that if a Java SPI-compliant decoder for a music file format becomes available, you can add support for it "in the field".

Suppose there is a Java SPI decoder named `jmmm.jar` for music files with a filename extension of `mmm`. Here is what you would need to do to get Music Library to be able to import and play these music files:

1. Download the `jmmm.jar` file and save it in your Music Library product folder.
2. Add the extension name (in this case, `mmm`) to the `extension.java` file. This is a source code file that is included with Music Library. (Note that the extension name must be coded in lower case, but the actual extensions on the music files can be in *any* case.) Then re-compile that file, producing an updated `extension.class` file. (Even a non-programmer can do this, as the source code file contains no "logic", it contains only "text string data".)
3. Update the "shortcut" for the Music Library Master program to include `jmmm.jar` at the end of its "classpath" (the `-cp` parameter). For example:

```
-cp .;.\tritonus_share.jar;...;.\jflac-1.1.jar;jmmm.jar
```

That's all that's required to be able to *import* and *play* files with the new `mmm` music format. However, if the new music file format has embedded metadata tags and you want Music Library to decode them during the import, some more work is necessary. Ideally, you would like for Music Library to read the metadata tags from the files themselves and to use that data to build the Music Library database entries. Otherwise, you will just get "bare bones" database entries and you will have to edit them manually to add artist name, track title, genre, *etc.*

## Technical Reference for Disc Library and Music Library

The “tag decoding” is user extensible, too, but (unlike adding one line of text to `extension.java` to recognize a new extension), extending the tag decoding requires some programming experience. The Music Library import process for music files automatically tries to call an extension-specific tag decoder module. If this module is not present, it is just skipped, harmlessly. For `mmm` files, Music Library will call the `mmm_tag.class` module, if available.

To make it easy, at least for a programmer, to produce a new tag decoder module, Music Library includes the source code files for two existing tag decoders: `ape_tag.java` and `flac_tag.java`. They are fully-coded (and functional) examples of how to code a “tag decoder extension” for Music Library.

Of course, before embarking on a tag decoder programming project, you will need to locate a detailed specification for the internal format of the embedded metadata tags in an `mmm` file. That could be the hardest part of the whole project!

## DMX Support

### DMX Serial Connection

Some models of DMX music service receivers have a serial connector on their rear panel, which provides metadata about the current song. This is a female DB25 connector with a non-standard pinout. On this connector, pin 1 is Tx (Transmit Data) and pin 20 is Gnd (Signal Ground).

To utilize the DMX support in Disc Library or Music Library, you will need to make a special cable. Assuming that the computer’s serial port has a DB9 connector, this cable should connect pin 1 on the DMX end to pin 2 (Rx, Receive Data) on the computer end, and connect pin 20 on the DMX end to pin 5 (Gnd, Signal Ground) on the computer end. One easy way to make such a cable is to use a male DB25-to-RJ11 adapter on the DMX end, a data-style (straight-through) cable with RJ11 plugs on both ends, and a female DB9-to-RJ11 adapter on the computer end. These adapters and cables are readily available at many electronics stores.

### DMX Status Messages

If the Disc Library or Music Library Master is configured with a DMX serial port, a “DMX Panel” is displayed on the Master computer’s screen. In addition, the DMX metadata is forwarded on the Disc Library or Music Library serial Control port (if configured). Furthermore, if Remote support is configured, a private IP protocol is used to support “DMX Remote” panels. This private IP protocol is similar to the IP protocol used to support the Remote version of Disc Library or Music Library; it normally uses IP port 65521, but it can be configured to use a different port number.

## Technical Reference for Disc Library and Music Library

Serial control status messages have the format "**DMX***s***text**", followed by a line termination character (Carriage Return, hexadecimal code 0D or decimal code 13), where the status code *s* is a single uppercase letter and **text** is a string of text characters. No blanks or other padding are present between **DMX** and the status code or between the status code and **text**.

IP protocol status messages have the format "*s***text**", followed by a line termination character (Line Feed, hexadecimal code 0A or decimal code 10), where the status code *s* is a single letter (uppercase or lowercase) and **text** is a string of text characters. No blanks or other padding are present between the status code and **text**.

The supported status messages are defined as follows:

Status Code	Text
I	Current channel code. The text is a one-byte code.
L	Current DMX ID code and record label. The text is a string of 0 to 32 text characters.
T	Current song title. The text is a string of 0 to 32 text characters.
A	Current artist name. The text is a string of 0 to 32 text characters.
C	Channel composer. The text is a string of 0 to 32 text characters.
D	Current album title. The text is a string of 0 to 32 text characters.
S	Time since the selection last changed, in milliseconds.

The only significant channel code is hexadecimal 5C. When this code is received, the current channel number is 103 (The DMX Listening Room), regardless of the current DMX ID code. When the current channel code is *not* hexadecimal 5C, the current channel number is the first three characters of the "DMX ID code and record label" string (three decimal digits, with leading zeros).

The first 10 characters of the "DMX ID code and record label" string are the DMX ID code. In all cases, this code is a unique identifier, internal to DMX operations, that specifies the current selection. The record label field starts with the 12<sup>th</sup> character of the "DMX ID code and record label" string and continues to the end of the string (typically, a maximum of 16 characters).

The only situation in which the status code *s* is a lowercase letter is when a DMX Remote program makes initial contact with the Master, using the IP protocol. At initial contact, a complete set of status messages is sent to the IP Remote, with all of the status codes changed to lowercase, except for the "time" message with status code *s*.

## Technical Reference for Disc Library and Music Library

The “time” message always has an uppercase status code **S**, and it flows only in the IP protocol and only at initial contact (it is not used in the serial control protocol). The purpose of the “time” message is to permit the Remote program to display the correct elapsed time for the selection that is playing at initial contact. Thereafter, the Remote program resets its elapsed time counter to zero whenever it receives any of the other status messages (with an *uppercase* status code).

## XM Support

The XM Control product (sold separately) can be integrated with Disc Library or Music Library. If you have Disc Library or Music Library, and also have XM Control, there are several advantages to this integration:

- The XM Master program will start automatically whenever Disc Library or Music Library is started, and will also terminate automatically whenever Disc Library or Music Library is terminated.
- If Disc Library or Music Library is configured with a serial Control port, it will be shared with XM Control. In other words, the XM Control serial commands and status messages will flow on the Disc Library or Music Library serial Control port. This will “save” a serial COM port on your computer, and also “save” an RS232 serial port on your automation controller.
- Disc Library or Music Library and XM Control will share the low-level serial interface library module. Some versions of Windows (such as Windows 98SE) tend to produce abnormal termination messages when multiple programs use separate (non-shared) copies of the serial interface library module.

To integrate XM control with Disc Library or Music Library, there are two simple steps:

1. Copy the XM Control configuration file **XMASTER.CFG** from the XM Control product folder into the Disc Library or Music Library product folder.
2. Update the “shortcut” for the Disc Library or Music Library Master program to include the XM Control product folder in its “classpath” (the **-cp** parameter). For example, for Disc Library:

```
-cp .;c:\xmtuner
```

Or for Music Library:

```
-cp .;c:\xmtuner;.\tritonus_share.jar;...;.\jflac-1.1.jar
```