

Playlist Database Structure

Users can generate an ASCII file which can in turn be imported directly into DAD.

In the most basic example of an ASCII file import; one could create a file with just the DAD cut number and a Letter code for the cuts Transition.

For example:

00007S
00010L

Advanced users can create an ASCII file that contains specific events such as Branch Events, Chain Events, Delays or even AutoFills. To do this, one must have that line entry formatted with specific codes in specific columns.

The DAD Playlist is a database with specific fields. Each field name, starting column and field width is described in the table to the right.

Each type of Playlist event subsequently contains specific data which must be entered into a specific field. Details for each type of DAD Event is listed on the following pages.

Field	Field Name	Type	Starting Column	Field Width
1	CUT	character	1	5
2	FUNCTION	character	6	1
3	DELAY	character	7	8
4	PLAYS	character	15	2
5	SEC	character	17	1
6	TER	character	18	1
7	SEGUE	character	19	1
8	TIME	character	20	8
9	BEGEND	character	28	1
10	CHAIN	character	29	8
11	ROTATE	character	37	8
12	TYPE	character	45	1
13	COMMENT	character	46	35
14	LINEID	character	81	10
15	STARTTIME	character	91	7
16	ENDTIME	character	98	7
17	FOSTART	character	105	7
18	FOLENGTH	character	112	7
19	FISTART	character	119	7
20	FILENGTH	character	126	7
21	LIBLOC	character	133	2
22	LIBNAME	character	135	8
23	GUID	character	143	36
24	ORDERID	Character	179	5
TOTAL				183

Notes for the Playlist Structure

There are some fields which must be used within the Playlist for all items in the list.

These fields are GUID and ORDERID. GUID is a Globally Unique Identifier for that cut. DAD will use the GUID field to help determine what cut is loaded or next to play. This becomes critical when users Refresh the playlist in a Playback deck. Without the GUID, the Playlist could return to the incorrect location in the log. OrderID is the items numerical position in the log. This value starts at the number 1 for the first entry in the Playlist and increments by one for each entry. Examples of GUID and OrderID are shown to the right.

Example of a GUID	OrderID
effadc2e-b1dd-4edf-ae25-ecb2b1cb4f75	00001
cbe110e9-f312-4e17-940f-a25419269f40	00002
Fb2a6289-ab99-4d7c-9dd1-5856c331b4ae	00003



NOTE: If GUID and OrderID are not in the ASCII file being Imported, they will be created by default by DAD. One can also load a Playlist in DAD and Save it, generating both GUID and OrderID values.

Playlist Event Details

Certain events within DAD playlists make special use of the various fields within playlist files. The following table contains the specific codes that are necessary for the event.



NOTE: *There are several Playlist Events that have specific notes pertaining to that type of Event. Please reference the note number [N1, N2] below the table.*

Playlist Event Type	Cut Col: 1 5 char.	Function Col: 6 1 char.	Delay Col: 7 8 char.	Time Col: 20 8 char.	Begend Col: 28 1 char.	Chain Col: 29 8 char.	Rotate Col: 37 8 char.	Type Col: 45 1 char.	Comment Col: 46 35 char.	FOSStart Col: 105 7 char.	LibLoc Col: 133 2 char.	LibName Col: 135 2 char.
Audio Cut [N1]	<5 digit number of file> 00007	A, L, S or N						P			F: [N2]	CUTS [N2]
Command Cut	<5 digit number of file> 00010	A, L, or N			E [N1]			P			F:	CUTS
Rotate Cut	<5 digit number of file> 31000	A, L, S or N			R			P			F:	CUTS
Playlist Cut	<5 digit number of file> 35000	A, L, S or N			L			P			F:	CUTS
Comment						[N1]		C	Text to be displayed			
DCL Event	CMD	A or L						E	<DCL to execute> ALERT 'hi there'			
Rotate Event	ROTAT	A, L or S			R		<playlist name>				F:	CUTS
Playlist Event	PLIST	A, L or S			L	0	<playlist name>					
Timed Event	TIMED	A or L		<HH:MM:SS> 15:36:00	1			T				
Basic Hard Branch	HARD	A or L	0	<HH:MM:SS> 15:36:20	2	0	0	T				
Forward Hard Branch	HARD	A or L	0	<HH:MM:SS> 15:36:30	2	0	1	T				
Basic Soft Branch	SOFT	A or L	0	<HH:MM:SS> 15:37:00	2	1	0	T				

Playlist Event Type	Cut Col: 1 5 char.	Function Col: 6 1 char.	Delay Col: 7 8 char.	Time Col: 20 8 char. <HH:MM:SS> 15:37:30	Begend Col: 28 1 char.	Chain Col: 29 8 char.	Rotate Col: 37 8 char.	Type Col: 45 1 char.	Comment Col: 46 35 char.	FOSStart Col: 105 7 char.	LibLoc Col: 133 2 char.	LibName Col: 135 2 char.
Forward Soft Branch	SOFT	A or L	0	<HH:MM:SS> 15:37:30	2	1	1	T				
Basic Soft Branch with a 'limit' of 12 seconds	SOFT	A or L	12	<HH:MM:SS> 15:37:40	2	1	0	T				
Basic Hard Branch 'All Hours'	HARD	A or L	0	<CC:MM:SS> CC:37:50	2	0	0	T		1		
Chain Event	CHAIN	A, S or L			H	<playlist to chain to>	1	H				
Smart Chain Events	CHAIN	A, S or L	0		H	DOWxxxxx MMDDxxxx YMMDDxxx YYMMDDxx [N1]	1	H				
Smart Chain Event with 2 day increment	CHAIN	A, S or L	0		H	DOWxxxxx MMDDxxxx YMMDDxxx YYMMDDxx [N1]	2	H				
Delay Event	DELAY	A or L	<seconds to delay> 61.10					D				
Time Announce [TAMS]	TAMS	A, S or L				<TAMS file name to use> MYTAMS	<drive letter for TAMS locations> R:	A				
AutoFill Start based on Length	AUTO		<length to fill to in seconds> 3661.00 (this is 01:01:01)		S	0.0	LENGTH	F				
AutoFill End based on Length	AUTO				E	0.0		F				
AutoFill Start based on Target Time	AUTO		<Time of day to fill to in seconds> 46861.0 (this is 13:01:01)		S	0.0	TARGET	F				
AutoFill End based on Target Time	AUTO				E	0.0		F				

Fill Cut Start based on Length using the 'A' category <N5	FILLC	Function Col: 6 1 char.	Delay Col: 7 8 char.	Time Col: 20 8 char.	Begin Col: 28 1 char.	Chain Col: 29 8 char.	Rotate Col: 37 8 char.	Type Col: 45 1 char.	Comment Col: 46 35 char.	FOStart Col: 105 7 char.	LibLoc Col: 133 2 char.	LibName Col: 135 2 char.
Fill Cut End based on Length	FILLC				E	0.0		G			F:	CUTS
Fill Cut Start based on Target Time using the 'B' category	FILLC		<Time of day to fill to in seconds> 56780.0 (this is 15:46:20)		S	0.0	TARGET	G		B	F:	CUTS
Fill Cut End based on Target Time	FILLC				E	0.0		G			F:	CUTS
Backtime Marker	BACK	A		<time to backtime to in seconds> 47400.00 (13:10:00)				B				
Backtime Branch Start	BTIME	A	0	<time of day backtime branch will START at. HH:MM:SS> 15:28:30 [N1]	S	0		T				
Backtime Branch End	BTIME	A	0	<time of day backtime branch will END at. HH:MM:SS> 15:30:30	E	0		T				

Special Notes for Fields

The Function field contains a letter value for the Transition Code. The Transition Code determines how that cut will interact with the following cut. For example, S would force a Segue with this cut and the following cut. These codes are listed in the table to the right.

Function Codes	Type Codes
(A)utoplay	T(A)MS event
auto(L)oad	(B)acktime marker
(N)ormal	(C)omment
(S)egue	(D)elay for
	(E)xecute command
	Auto(F)ill
	(G) FillCut
	c(H)ain to
	(O)ver cut
	normal (P)lay
	(R)otate to
	(T)imed event
	(P)rogram item

The Type field holds the codes for the Type of event. A listing of these codes and their respective Events are listed in the table to the right.

Special Notes for Events

Event Type	Event Note	Note Description
Audio Cut	[N1]	Audio cuts can have other Playlist fields flagged for specific purposes. Audio cuts can also be defined in the Playlist to have specific transition events such as Segues or Overs. This is covered in the next table concerning Audio cuts.
	[N2]	Every Library cut and some specific Events in addition to Cut numbers must have the DAD Library Location [LibLoc] and Library Name [LibName] filed set. For users of Multiple Libraries in DAD, one would specify the Library Location and Name for that cut. For example: Cut 00010 could be in either F: CUTS or in F: SPONSORS
Command Cut	[N1]	Command Cuts should have an 'E' for Execute in the BegEnd field, however it is not required by DAD for the command cut to function.
Comment	[N1]	Users can give the Comment line a color. One should enter the decimal equivalent of the hex value for the color. For example: 16711680 is the Decimal equivalent for FF0000 Hex. This is the Hex value for the color blue.
Smart Chains	[N1]	When using Smart Chains, the entry within the Chain field must be expressed as shown. Smart Chains use the values of DOW, MMDD and YMMDD to automatically determine the Day of Week, two digit numbers for the Month, two digit numbers for the day and digits for the year. The 'xxxxx' portion of the entry would be used by DAD and would be based upon the Playlist currently loaded. All values are defaults for that specific Chain value and should not be altered. For example: If one has a playlist called MONKTAM for Monday KTAM then the Smart Chain would be DOWxxxxx. DAD would automatically load the TUEKTAM log when the Smart Chain is played.
Backtime Branch Start	[N1]	Backtime Markers are two sets of Branches. The Start Marker should have it's time offset by the amount of total time for all audio cuts (taking into account segue lengths) between the Start and End Marker. For example: If there are exactly two minutes of audio between the Start and End Markers with no segues then the Start Marker would be set two minutes before the End Marker.

Special Notes for Audio Cuts

Function Code / Transition Codes

Users can create Transition codes and Segue Types for audio cuts to control how that cut interacts with the cut that follows. The following table covers the four basic Transition codes one can use in DAD. These Transition codes are entered into the Function Field.

Function Code	Function Description
A Autoplay	Autoplay cuts will automatically load and play the cut that follows in the playlist. The cuts will not overlap and will play end to start.
L autoLoad	AutoLoad cuts will automatically load the following cut but will NOT play it. It will Load the cut only.
S Segue	Audio cuts with a Segue transition can overlap with the following audio cut.
N Normal	Cuts given a Normal transition will NOT load the next cut and will NOT play the next cut. When the cut finishes playing, it will remain in the deck until another event or user interaction loads and plays the next file.

Dealing with Segues



NOTE: One should note that there are two basic DAD configurations options that effect on what DAD uses to create the segue. In DAD under SETUP then Cue and Automation Defaults the Switch Segue Origin can be set to Use Library (up) or Use Defaults (down). When set to Use Library, DAD will use the Segue Head pointer to determine when to transition to the next cut. Most facilities prefer this method as it allows them to control the segue overlap on a cut by cut basis. With this switch set to Use Defaults; DAD will force use the Default Transition time and type in this screen. In this mode, all cuts will have the same overlap.

Depending on this switch, specific values in the Playlist database structure can be different. Specifically the value in the Delay field. When using the switch option in Use Defaults, the Default Transition time will be entered into the Delay field. When set to Use Library, the Delay field will be set to 0.0, allowing the cut to transition at the Segue Head pointer.

The following table details the types of Segues available.



NOTE: The *Graphic* column at the end of the Table is **ONLY** used to show a visual graphic of the transition type. The *DAD Playlist* does **NOT** use this column.

Cut Col: 1 5 char.	Function Col: 6 1 char.	Delay Col: 7 8 char.	Plays Col: 15 2 char.	SEC Col: 17 1 char.	TER Col: 18 1 char.	Segue Col: 19 1 char.	Time Col: 20 8 char.	Graphic of Segue Type
00008	S	3.00 [N1]	01	0	0	1		
00008	S	0.0 [N2]	01	0	0	1		
00008	S	0.0	01	0	0	2		
00008	S	0.0	01	0	0	3		
00008	S	0.0	01	0	0	4		
00008	S	0.0	01	0	0	6 [N3]		
00008	S	0.0	01	0	0	7 [N3]		
00008	S	0.0	01	0	0	8 [N3]		

NOTES:

[N1] Entering a time value in seconds into the Delay field for any Segue Transition with all types EXCEPT Segue Type 5 (which is Tracker) will cause the audio cut to overlap the following cut by that amount.

[N2] A value of 0.0 will allow DAD to use the Segue Head marker on the cut determine the transition time.

[N3] When using Segue Type 6, 7 and 8, there will not be an overlap of audio between the cuts. Instead one cut will end and the next will start. Depending upon the Segue Type, the audio cut could fade in or out or be at full volume.

Using SEC or TER markers

One can also embed specific flags to trigger automation events such as SEC or TER markers. If the audio cut has a SEC; which includes SLATE or TER marker set, one must also set the corresponding value in the Playlist.

Cut	Function	Delay	Plays	SEC	TER	Segue	Time
Col: 1 5 char.	Col: 6 1 char.	Col: 7 8 char.	Col: 15 2 char.	Col: 17 1 char.	Col: 18 1 char.	Col: 19 1 char.	Col: 20 8 char.
00008	S	0.0	01	1	1	1	

For example, if one has an audio cut that contains embedded SLATE markers and that cut is scheduled within a Playlist, one must set the SEC field to 1 for that cut. Likewise, if one were to use TER cues, then one would set the value in the TER field to 1. DAD will only execute SEC or TER cues in the Playlist when the value of SEC or TER has been set to a value of 1. When set to 0, the SEC or TER cues will be ignored.

Creating Overs

Advanced users can create Overs. An Over is one audio cut playing over the very beginning of another cut. The base cut (typically a song) must start first, with the Over playing 'over' the top of the base cut. One can also define a 'duck' value to dip the audio playback level of the Base cut while the Over is playing.

Cut Col: 1 5 char.	Function Col: 6 1 char.	Delay Col: 7 8 char.	Plays Col: 15 2 char.	SEC Col: 17 1 char.	TER Col: 18 1 char.	Segue Col: 19 1 char.	Chain Col: 29 8 char.	Rotate Col: 37 8 char.	Type Col: 45 1 char.	LibLoc Col: 133 2 char.	LibName Col: 135 2 char.
44044	S	0.00	01	0	0	1	75		P	F:	CUTS
33350		2.30						90	O	F:	CUTS

In the table above, the base audio cut is 44044 and the cut that will play as the Over is cut 33350.

In this example the Over cut waits 2.30 seconds before it starts to play. When the Over starts to play, DAD will 'duck' the level of the base cut to a value that is 75% of the original volume. This 'duck' level is defined in the Chain field for the Base cut. To 'duck' the level of the Over cut, one would enter a value into the Rotate Field. In this example the Over will play at 90% of the original volume.